Automated Detection of Oral Lesions Using

Deep Learning for Early Diagnosis of Oral Cancer

by

Gizem Tanrıver

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Data Science

February 10, 2021

Automated Detection of Oral Lesions Using Deep Learning for Early Diagnosis of Oral Cancer

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Gizem Tanrıver

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Assist. Prof. Mehmet Cengiz Onbaşlı (Advisor)

Assist. Prof. Onur Ergen (Co-Advisor)

Prof. Kemal İnan

Assoc. Prof. Vakur Olgaç

Assoc. Prof. Merva Soluk Tekkeşin

Date: _____

ABSTRACT

Automated Detection of Oral Lesions Using Deep Learning for Early Diagnosis of Oral Cancer

Gizem Tanrıver

Master of Science in Data Science

February 10, 2021

Oral cancer is a major health concern in many countries leading to approximately 330,000 deaths every year worldwide. Early detection and treatment remain to be the most effective measures in reducing mortality and morbidity from oral cancer with survival rates achieving 75-90% in most cases. Over the years, screening programmes have been put in place to improve detection of precursor lesions that have high probability of progressing into cancer. The implementation of these programs, however, has proved to be challenging in a real-world setting as they rely on primary care practitioners who are often not adequately trained to recognize these lesions that exhibit substantial variability in presentation. The development of vision-based adjunctive technologies that can assist screening and clinical decision making is crucial for improving early detection of oral cancer.

In this study, we explore the potential applications of various computer vision techniques to the oral cancer domain in the scope of photographic images and investigate the prospects of an automated system for oral cancer screening. Exploiting the advancements in deep learning, we propose a two-stage framework to detect oral lesions with a detection network in the first stage and classify the detected region as benign, precancer, or cancer with a classifier network in the second stage. A custom dataset was developed for the study due to the lack of publicly available dataset for oral cancer. Our preliminary results demonstrate the feasibility of deep learning-based approaches for automated detection and classification of oral lesions in real-time. We envisage that the proposed model offers great potential as a low-cost and non-invasive tool that can support screening processes and improve early detection of oral cancer.

ÖZETÇE

Yüksek Lisans Tez Başlığı
Gizem Tanrıver
Veri Bilimleri, Yüksek Lisans
10 Şubat 2021

Ağız kanseri, her yıl dünya çapında yaklaşık 330.000 ölüme yol açan önemli bir sağlık sorunudur. Erken teşhis ve tedavi, çoğu durumda %75-90'a ulaşan hayatta kalma oranları ile ağız kanserinden ölüm ve morbiditeyi azaltmada en etkili önlem olmaya devam etmektedir. Yıllar içinde, kansere dönüşme olasılığı yüksek olan öncül lezyonların tespitini iyileştirmek için tarama programları uygulamaya konmuş, ancak bu programların gerçek dünyada uygulanmasında zorluklarla karşılaşılmıştır. Bunun başlıca sebebi olarak, tarama programlarının dayandığı birinci basamak pratisyenlerinin sunumda önemli değişkenlik gösteren bu lezyonları tanımak için gereken yeterli eğitimi almamış olması gösterilmiştir. Tarama ve klinik karar vermeye yardımcı olabilecek görme temelli yardımcı teknolojilerin geliştirilmesi, ağız kanserinin erken teşhisini iyileştirmek için çok önemlidir.

Bu çalışmada, çeşitli bilgisayarlı görme tekniklerinin ağız kanseri alanındaki potansiyel uygulamalarını fotoğrafik görüntüler kapsamında araştırıyor ve ağız kanseri taraması için otomatik bir sistemin olasılıklarını inceliyoruz. Derin öğrenmedeki ilerlemelerden yararlanarak, birinci aşamada bir tespit ağı ile oral lezyonları tespit etmek ve ikinci aşamada da tespit edilen bölgeyi bir sınıflandırıcı ağ ile iyi huylu, prekanseröz veya kanser olarak sınıflandırmak için iki aşamalı bir çerçeve öneriyoruz. Ağız kanseri için kamuya açık veri setinin bulunmaması nedeniyle çalışma için özel bir veri seti geliştirilmiştir. İlk sonuçlarımız, oral lezyonların gerçek zamanlı olarak otomatik olarak algılanması ve sınıflandırılması için derin öğrenmeye dayalı yaklaşımların uygulanabilirliğini göstermektedir. Önerilen modelin, tarama süreçlerini destekleyebilen ve ağız kanserinin erken teşhisini iyileştirebilen, düşük maliyetli ve invaziv olmayan bir araç olarak büyük bir potansiyel sunacağını öngörüyoruz.

ACKNOWLEDGEMENTS

I would like to thank my advisors Assist. Prof. Cengiz Onbaşlı and Assist. Prof. Onur Ergen for providing me with the opportunity to pursue this thesis project and for their support throughout my studies. I has been a tough but fruitful learning experience, and I am thankful to Assist. Prof. Onur Ergen for opening the doors of a such interdisciplinary group that I had sought. I would also like to express my gratitude to Assoc. Prof. Merva Soluk Tekkeşin for her precious time in developing the dataset for the study. This thesis would not be possible without her input and guidance. Furthermore, I am very thankful to my thesis committee members Prof. Kemal İnan and Assoc. Prof. Vakur Olgaç for their valuable time and comments.

I would like to acknowledge Koç University Graduate School of Sciences and Engineering and the Scientific and the Technical Research Council of Turkey (TÜBITAK) for offering me the financial support to conduct my studies.

I had the opportunity to meet wonderful colleagues during my time at Koc University. First and foremost, I am very grateful to my colleague and good friend Gökçe İymen for supporting me on every aspect of my studies and her endless friendship. I wish to thank all my friends and colleagues Ecem Çelik, Karim Sonbol, Buse Ünlütürk, Can Adalı, Çelen Naz Ötken, Kerim Uygur Kızıl, Waris Gill, and Mohammed Saif Kazamel for their support during my studies. I must also thank Elif Demirtaş for her contribution on building the dataset for the study.

My heartiest gratitude goes to my family for their unbounded support at all times. Last but not least, I am most grateful to Efe Sürekli who has given me the courage to take the leap in my career and has always believed in me throughout this journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

AP	Average Precision
AR	Average Recall
CNN	Convolutional Neural Network
CSP	Cross-Stage Partial
F-CNN	Fully Convolutional Neural Network
FLOPS	Floating-Point Operations Per Second
FN	False Negative
FP	False Positive
FPN	Feature Pyramid Networks
FPS	Frame per Second
GLCM	Gray Level Co-occurrence Matrix
GPU	Graphics Processing Unit
HOG	Histogram of Oriented Gradients
IoU	Intersection over Union
LBP	Local Binary Patterns
ML	Machine Learning
MLP	Multi-layer Perceptron
NMS	Non-Maximum Suppression
OPMD	Oral Potentially Malignant Disorders
OSCC	Oral Squamous Cell Carcinoma
PANet	Path Aggregation Network
ReLU	Rectified Linear Unit
RGB	Red Green Blue
ROI	Region of Interest
RPN	Region Proposal Network
SGD	Stochastic Gradient Descent
SPP	Spatial Pyramid Pooling
SSD	Single Shot MultiBox Detector
SVM	Support Vector Machines
t-SNE	T-Distributed Stochastic Neighbour Embedding
TP	True Positive

TN	True Negative
TTA	Test Time Augmentation
VIA	VGG Image Annotator

Chapter 1:
INTRODUCTION

1.1 Motivation

Oral cancer is one of the most prevalent cancers worldwide; nevertheless, it is highly amenable to early diagnosis and treatment (Saman Warnakulasuriya & Greenspan, 2020). Unfortunately, most cases of oral cancer are diagnosed at a late stage and progress with high morbidity and mortality (Seoane et al., 2016). Oral squamous cell carcinoma (OSCC), which accounts for over 90% of oral cancer cases, is often preceded by oral potentially malignant disorders (OPMD) (Saman Warnakulasuriya & Greenspan, 2020). These early stage lesions typically do not incur any discomfort and may appear as innocuous lesions at early stages, leading to late presentation of patients to healthcare centres (Markopoulos, 2012). Considering the alarming incidence and mortality rates, screening programs have been put in place in recent years to detect lesions with high probability of progressing into cancer particularly in high-risk populations (Saman Warnakulasuriya & Greenspan, 2020). However, implementation of these screening programs based on visual examination has been found to be problematic in a real-world setting as they rely on primary care healthcare professionals who are often not adequately trained or experienced to recognize these lesions (Scully et al., 2008; Saman Warnakulasuriya & Greenspan, 2020). The substantial heterogeneity in the appearance of oral conditions makes their identification very challenging for primary care healthcare professionals and is considered to be the leading cause of delays in referrals to oral cancer specialists (Scully et al., 2008). These findings imply that adjunctive technologies aimed to support screening process can be effective tools for improving early detection of oral cancer (Ilhan et al., 2020).

1.2 Thesis Statement

Advances in the fields of computer vision and deep learning offer powerful methods to develop adjunctive technologies that can perform an automated screening of the oral cavity and provide real-time feedback to healthcare professionals during patient examination as well as to individuals for self-examination. In this thesis, we aimed to

explore the potential applications of various computer vision techniques to the oral cancer domain in the scope of photographic images and investigate the prospects of a deep learning-based automated system for oral cancer screening. For the purposes of building an automated tool for detection of oral cancer, we evaluated several segmentation and detection algorithms to isolate the lesion area from photographic images and classification algorithms to identify the lesion as benign, precancerous, or cancer.

1.3 Contributions

In this thesis, we have shown that computer vision and deep learning offer great potential for automated detection, segmentation, and classification of oral lesions from photographic images which exhibit substantial variability in terms of the image quality and the presentation of disease. By using some of the most recent and advanced architectures such as YOLOv5 and EfficientNet, our proposed two-stage model provides a means of automated detection and classification of oral lesions in real-time and paves the way for a low-cost, easy, and non-invasive tool that can support screening processes and improve early diagnosis of oral cancer.

Another contribution of this thesis is the development of an oral lesion dataset. As there is no publicly available dataset for oral cancer, we collated a dataset containing photographic images of various types of oral lesions and their corresponding annotations completed by an oral cancer specialist for our studies. The part of the images comes from our collaborating hospital and the rest is collected from public resources for a total of 652 images. Approximately half of the dataset also contains annotations for teeth, which may be useful for studying dental conditions. Although the dataset is relatively small in the context of deep learning, it can be utilized in future studies to investigate a range of research questions and will be a valuable asset for researchers in this field.*

1.4 Outline

The rest of the thesis is structured as follows: Chapter 2 presents the medical background of oral cancer and the overview of deep learning and computer vision

* The research started at Dr. Ergen's lab at Koç University and will continue at his new lab at Istanbul Technical University.

techniques. The deep learning architectures used in image classification, segmentation, and object detection are described in Section 2.3.2. Furthermore, related works on detection and classification of oral lesions using deep learning are described in Section 2.4. Chapter 3 presents our dataset and provides a detailed setup of our detection, segmentation, and classification experiments. Chapter 4 presents our results followed by evaluation and discussion for each experiment. The deployment of our proposed two-stage detection and classification model is presented in Section 4.4. The thesis is summarized and concluded in Chapter 5 with a discussion of limitations and directions for future research.

Chapter 2:
BACKGROUND

2.1 Medical Background of Oral Cancer

2.1.1 Epidemiology of Oral Cancer and Risk Factors

Oral cancer is the eight most common cancer worldwide, accounting for an estimated 657,000 new cases and 330,000 deaths each year (Saman Warnakulasuriya & Greenspan, 2020; WHO, 2018). The incidence rate varies significantly by socio-economic state and geography with two-thirds of global cases occurring in low-income countries and half of them occurring in South Asia countries such as India and Pakistan particularly due to exposure to high-risk factors in these countries (Saman Warnakulasuriya & Greenspan, 2020). High-income countries such as the UK and USA have also seen a steady increase in the number of cases since the 1970s (Saman Warnakulasuriya & Greenspan, 2020).

The most prominent risk factors of oral cancer are tobacco use (including chewing tobacco and using snuff) and alcohol consumption, accounting for 90% of all cases (Petersen, 2005). Another major risk factor is the use of betel quid (i.e., a mixture containing areca nut) with or without tobacco, which is common in South Asia countries (Saman Warnakulasuriya & Greenspan, 2020). According to WHO, the incidence and mortality rates are higher in men than in women, which can be partially explained by higher tobacco and alcohol use by men in certain countries (Petersen, 2005). Oral cancer is considerably more common in older age groups with a mean age-of-onset of 60 years; however, rising incidence rates has been reported in younger adults (under 45 years of age) in recent years especially among white women (Saman Warnakulasuriya & Greenspan, 2020). The tongue cancer in particular has been found to be more prevalent in younger groups compared to other subsites of oral cavity (Saman Warnakulasuriya & Greenspan, 2020). The mortality rates for oral cancer vary substantially depending on the disease stage and factors such as age, gender, and comorbidities. Despite the advancements in diagnostic technologies and treatment, the 5-year survival rate remains at around 50% for oral cancer – which further decreases to below 50% in South Asia countries (Saman Warnakulasuriya & Greenspan, 2020).

2.1.2 Clinical Presentation of Oral Cancer

Anatomical regions of the oral cavity include the tongue, floor of mouth, buccal mucosa (cheeks), palate, gingiva (gum), retromolar trigone, and lip as shown in Figure 1. Although oral cancer may arise in any of those regions, tongue and mouth floor are reported to be more common sites for oral cancer (Markopoulos, 2012; Saman Warnakulasuriya & Greenspan, 2020).

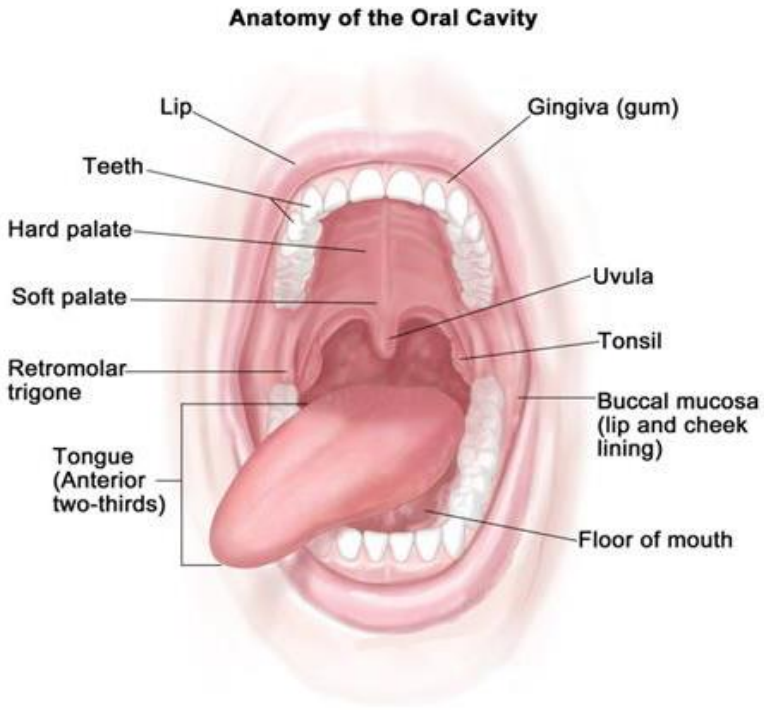


Figure 1: Anatomy of the oral cavity (Coleman et al., 2019).

Around 90% of oral cancers are oral squamous cell carcinoma (OSCC) which is a malignant neoplasm of epithelial origin (Saman Warnakulasuriya & Greenspan, 2020). Some OSCCs may appear on seemingly normal oral mucosa, but most of them are preceded by oral potentially malignant disorders (OPMD) which represent a range of conditions such as leukoplakia, erythroplakia, erythroleukoplakia, and submucous fibrosis (Markopoulos, 2012). The risk of malignant transformation depends on the type of OPMD, the lesion features such as colour, size, and location, and the patient profile especially the gender.

Leukoplakia is the most common OPMD which appears as a persistent white patch or plaque that cannot be rubbed off (Saman Warnakulasuriya & Greenspan, 2020). As

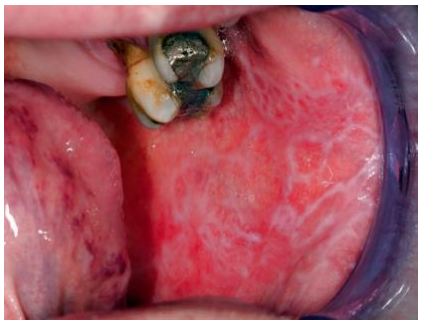
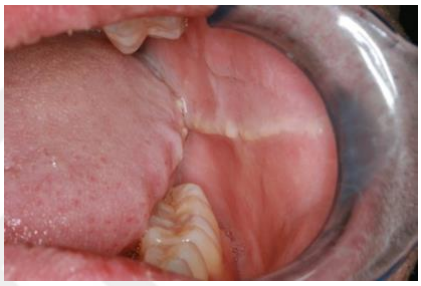
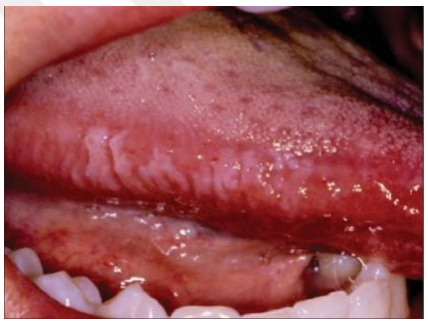
shown in Table 1, leukoplakia has several clinical types such as homogenous, nodular, verrucous, and erythroleukoplakia based on colour, thickness, and texture characteristics of the lesion (Saman Warnakulasuriya & Greenspan, 2020). It should be differentiated from other types of white lesions (Table 2) such as lichen planus, frictional keratosis, alveolar ridge keratosis, hairy leukoplakia, candidiasis, and linea alba buccalis as these are not considered as OPMD (Saman Warnakulasuriya & Greenspan, 2020). The rate of malignant transformation for leukoplakia is reported to be in the range of 0.13–34% with higher rates observed at the sites of mouth floor and tongue (S. Warnakulasuriya & Ariyawardana, 2016). Non-homogenous appearance and high-grade dysplasia are also correlated with higher malignancy rate (S. Warnakulasuriya & Ariyawardana, 2016). Other types of OPMD that have high risk of developing into malignancy are erythroplakia which appear as a persistent red lesion and submucous fibrosis which is caused by excess deposition of collagen and usually exhibits fibrosis bands with leathery mucosa as shown in Table 1. Clinical conditions such as erythema migrans, desquamative gingivitis, pemphigoid, and erosive lichen planus may resemble erythroplakia and should be excluded when making a clinical diagnosis. According to a systemic review performed in 2010, erythroplakia appears to have the highest rate of malignant transformation among all OPMDs at around 44.9% (Villa et al., 2011). The reported rates for oral submucous fibrosis varies between 1.9% to 9.1%, with one study reporting up to 52.5% in the concurrent presence of leukoplakia (Saman Warnakulasuriya & Greenspan, 2020).

Table 1: Lesions of OPMD which have high prevalence of malignant transformation (Saman Warnakulasuriya & Greenspan, 2020).

Leukoplakia	Homogenous leukoplakia ▪ Uniform, white, flat surface ▪ May display cracks on the surface	

	Nodular leukoplakia ▪ Small rounded outgrowths	
	Verrucous leukoplakia ▪ Raised, exophytic surface ▪ Proliferative variant has high risk of malignant transformation	
	Erythroleukoplakia ▪ Mixed white and red (predominantly white) ▪ Irregular margins	
Erythroplakia	▪ Persistent, solitary red patch ▪ Well-defined margins ▪ Occur mostly on the mouth floor, soft palate, and ventral tongue	
Submucous fibrosis	▪ Exhibits blanching and pigmentation ▪ Vertical fibrous brands ▪ Leathery mucosa ▪ Betel chewing common risk factor	

Table 2: Other white lesions that should be excluded when making a clinical diagnosis for leukoplakia (Saman Warnakulasuriya & Greenspan, 2020).

Lichen planus	▪ Commonly as lace-like white striations, but may also present as white plaques, erythema, or erosion. ▪ Multi-focal and symmetrical distribution	
Keratosis	▪ Traumatic or frictional ▪ Usually along occlusion plane	
Hairy leukoplakia	▪ Appears bilaterally on lateral tongue as white stripes or plaques	
Candidiasis	▪ Caused by Candida infection ▪ Can be scrapped off unlike leukoplakia	
Leukoedema	▪ Appears bilaterally on buccal mucosa ▪ Disappears upon stretching	

OPMDs are usually painless at the initial stages therefore may not be noticed. These lesions often develop pain and burning sensation as they progress into OSCC (Markopoulos, 2012). Differential diagnosis of early-stage OSCC should be made to exclude traumatic ulcerations and eosinophilic granuloma (i.e., TUGSE) which can be confused with OSCC (Saman Warnakulasuriya & Greenspan, 2020). While early-stage OSCCs can be very small in size, late-stage OSCC tumours grows over 4cms and often manifest as exophytic growths or profound ulceration, as shown in Figure 2. Late-stage OSCCs are often accompanied with heavy pain, dental mobility, and bleeding (Saman Warnakulasuriya & Greenspan, 2020).

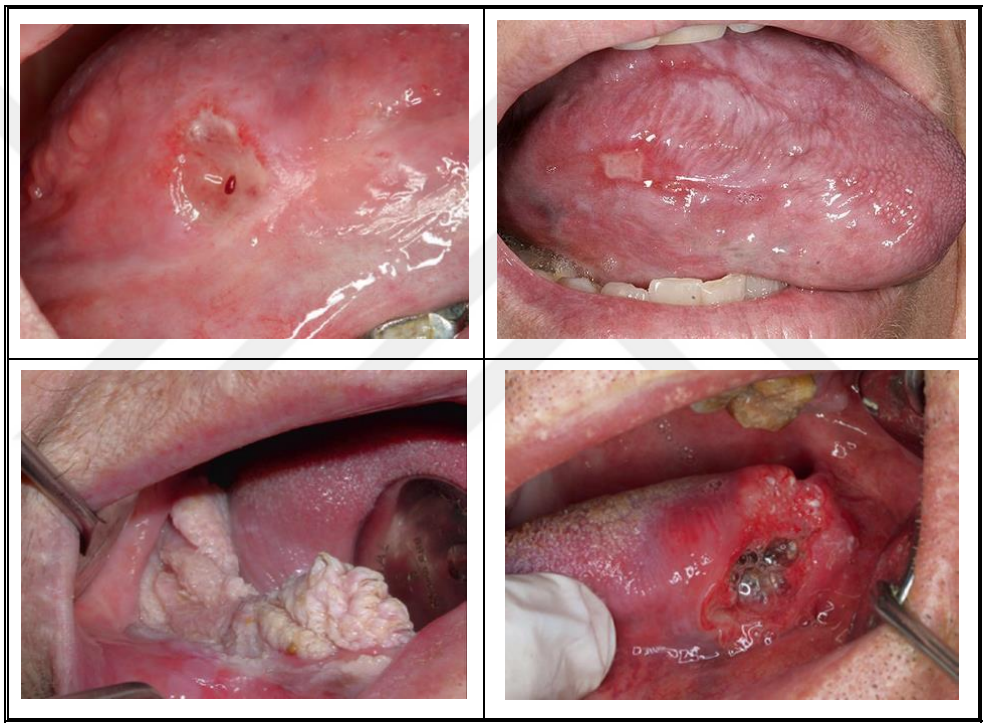


Figure 2: Early-stage (top) and late-stage (bottom) OSCCs with different characteristics.

2.1.3 Diagnostic and Therapeutic Approaches

Visual examination of oral cavity is the initial stage for identification of OPML and OSCC (Scully et al., 2008). This stage often takes place in a primary care setting such as dental practice, primary care physician's office (i.e., general practitioner surgery), or community health centre in the case of a low-resource region. If there is any suspicion for a potentially malignant condition, careful examination of oral mucosa and review of patient history are conducted to decide whether the lesion meets the criteria for a clinical

diagnosis of OPMD. The clinical diagnosis is supplemented with tissue biopsy and histopathological examination which is the gold standard for determining the exact diagnosis (Saman Warnakulasuriya & Greenspan, 2020). Typically, patients are referred to a specialist, but if this is not feasible, biopsy sample can be taken by the staff at the primary care site (Saman Warnakulasuriya & Greenspan, 2020). The process for biopsy is usually slow – taking at least a few days – and the results are subjective to the pathologist's interpretation, not mentioning the uncomfortable experience for the patient (Scully et al., 2008). In a primary care setting, medical staff often do not have the adequate training and experience for performing appropriate risk-assessment and tissue biopsy, which stands as one of the leading reasons for overlooked or misdiagnosed high-risk lesions (Scully et al., 2008). This does not come at a surprise as there is a substantial heterogeneity in the appearance of oral conditions and even highly trained medical professionals may fail in identifying all types of OPMD and early-stage OSCC (Silverman, 1988). A systematic review to evaluate the accuracy of conventional visual examination showed that clinicians had 93% sensitivity but a striking 31% specificity in identifying OSCC lesions, which suggests that many patients are likely to be referred for biopsy to no avail (Epstein et al., 2012).

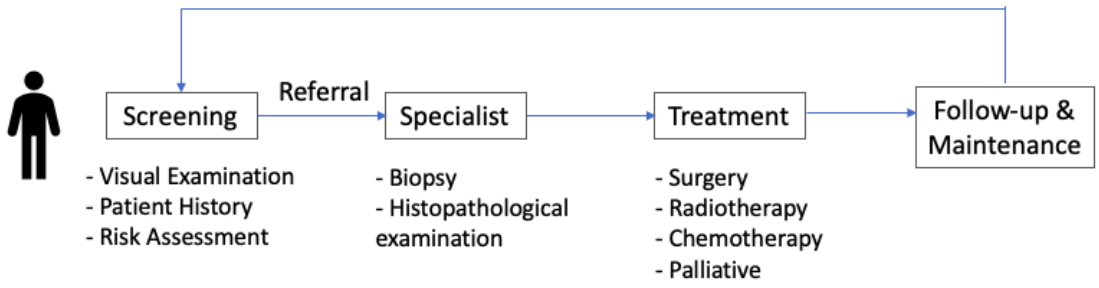


Figure 3: Patient pathway from initial screening to treatment and follow-up.

Surgery and radiotherapy are primary treatment choices in the early stages of OSCC, which often yields high treatment success and leads to curing of the disease permanently (Markopoulos, 2012). The overall survival rate for early-stage OSCC can be up to 75-90% (da Silva et al., 2012). Unfortunately, over 60% of the cases are diagnosed at a later stage and progress with high morbidity and mortality (Seoane et al., 2016). In the later stages of OSCC, a combination of surgery, radiotherapy, and/or chemotherapy is the

preferred choice of treatment; however, despite major advances in treatment modalities, the 5-year survival rate does not go beyond 12% in most cases (Markopoulos, 2012). Furthermore, treatments may not be easily accessible or affordable in some parts of the world, worsening the situation. After receiving treatment, patients may experience local or regional recurrence of the disease. Recurrence rates vary from 18 to 76% for patients who underwent standard treatment, which contributes to poor survival rates (da Silva et al., 2012). Therefore, continuous surveillance of patients is essential for detecting recurrences and long-term management of the disease.

2.1.4 Early Detection and Screening for Oral Cancer

Early detection and treatment offer a great potential to decrease mortality and morbidity outcomes from oral cancer (Markopoulos, 2012). However, diagnostic delay still remains a challenge in many countries. Potential issues are listed below:

- Lack of patient awareness
- Asymptomatic progression in early stages (pain-free)
- Complexity of oral cavity and locality of lesion
- Small size and/or innocuous appearance of lesions
- Variable presentation of disease (including ethnicity related differences in pigmentation)
- Inadequate training or experience of primary care staff on oral cancer
- Lack of access to specialty care in low-resource communities
- Burden of biopsy procedure

Oral cancer screening has been an important part of many healthcare programs worldwide as a measure to address these issues and improve early detection of oral cancer (WHO, 2018). Considering the alarming incidence and mortality rates, screening programs are of utmost importance especially in high-risk populations. Since most OSCC lesions are preceded by an OPMD such as leukoplakia or erythroplakia, detection of lesions with high probability of progressing into cancer is an essential part of the screening programs (Saman Warnakulasuriya & Greenspan, 2020). The screening procedure can take place as a self-examination of oral cavity by the individual or at a primary care site during routine care and includes asymptomatic individuals. Despite various screening guidelines that have been put in place, implementation of screening

programs based on visual examination has been challenging in a real-world setting (Saman Warnakulasuriya & Greenspan, 2020). The majority of baseline encounters with OSCC and OPMD patients occur in dental practices; however, most dentists struggle with recognizing premalignant lesions. According to one study, dentists were able to discriminate OSCC and OPMD from benign lesions with only 31% specificity (Ilhan et al., 2020). Another study showed that about 75% of graduating dental students did not consider they were able to identify precancerous lesions (Burzynski et al., 2002). These findings suggest that adjunctive technologies aimed to support screening process can be effective tools for improving early detection of oral cancer (Ilhan et al., 2020; Tatehara & Satomura, 2020). An effective screening tool can not only reduce mortality and morbidity from oral cancer but also it brings significant cost benefits in the long term. Besides, distinguishing benign lesions from precancerous and cancerous lesions with high specificity has the potential of reducing needless biopsy operations for patients and further adds up to the cost benefits.

2.2 Overview of Machine Learning

Machine learning (ML) is a growing field of computer science. In essence, it aims to get computers to automatically *learn* based on example data or past experience. More specifically, it is built on a mathematical model defined by certain parameters which are learned during training phase through an optimization process (Alpaydin, 2014). The trained model can then be used to make predictions or describe a system.

ML algorithms are used in a vast range of applications from fraud detection, online advertising, and machine translation to medical diagnosis. The study of ML involves various approaches and conventionally these approaches fall into three main categories:

- **Supervised learning:** In supervised learning, both the input X and the output label y are provided to the algorithm for each data point by a supervisor. If the output label comes from a discrete set of values, it is called a “classification” problem. On the other hand, if it is a continuous variable, it is called a “regression” problem. In this supervised approach, all data points should be labelled beforehand with corresponding ground-truth labels for each instance (Alpaydin, 2014).

- **Unsupervised learning**: In unsupervised learning, the output labels are not provided to the model. The purpose of this approach is to unearth the commonalities or patterns in the input data without having the output labels. The most common method is clustering, where each data point is allocated to a cluster (Alpaydin, 2014).
- **Reinforcement learning**: Reinforcement learning is another approach which is used in applications where a sequence of actions – also called policy – is desired as an output to reach a certain goal. Unlike supervised learning, correct outputs are not provided to the algorithm; instead, an agent discovers the optimal policy by trial and error by interacting with its environment (Bishop, 2006).

2.2.1 Machine Learning Algorithms

In a classification task, the aim is to assign each data point to a class label *c*. If the task involves only two classes, it is called a "binary classification" task. If it involves more than two classes, then it is called a "multi-class classification" task. The following are some of the most commonly used models for classification.

- **K-Nearest Neighbour** classifier is a distance-based algorithm which emanates from the notion that the more similar the data points are, the smaller distance there will be between them hence they are more likely to belong to the same class (Alpaydin, 2014). The model assigns a data point to the class which takes the majority voting among its k nearest neighbours, where k is a positive integer (commonly k = 3, 5, 7 etc.). Both the choice of k and the distance metric are hyperparameters for this model.
- **Random Forest** is an ensemble model constituted of many decision trees each of which is generated by selecting features from a given dataset based on a set of learned rules (Alpaydin, 2014). The partitioning of a decision tree occurs recursively starting from the root node until reaching a terminal node which represents the target output for that branch. An impurity measure such as *entropy* or *gini index* is used as a metric to quantify the goodness of the split at every step. In the case of random forest, different trees are formed by either taking random subsets of dataset or random subsets of the features, then training each one of them

individually. The resulting predictions from each tree are combined to give a final prediction. This ensemble approach can boost the accuracy significantly.

- **Support Vector Machines (SVM)** is a more complex type of linear discriminant method, which is very powerful and widely used in many applications. Considering that there may be multiple solutions instead of a unique one that can classify our data, SVM aims to find the most *general* classifier by maximizing the margin that is the distance from the boundary to the closest datapoint (Alpaydin, 2014; Bishop, 2006). For this purpose, SVM utilizes the data points that are closer to the boundary – called "support vectors", which helps to estimate the error and find the optimal boundary. This can be represented as a dual problem and solved efficiently with convex optimization. While SVM can work as a linear classifier, it can also classify data when it is not linearly separable by mapping it to a higher dimensional space (i.e. "kernel trick") (Hendrycks, 2015).

2.2.2 Neural Networks

Neural Networks are a class of ML algorithms which were designed with an inspiration from human brain. Similar to the neurons in the brain connected through a net of synapses, neural networks are composed of artificial neurons. The most basic form of artificial neural networks is "perceptron". As shown in Figure 4, it takes a weighted sum of input features x_j, j=1,..,d, and a bias term, and applies an activation function f to give output y, where each w_j represents the weight of the connection from x_j to y (Alpaydin, 2014). The Eq. 2-1 shows the mathematical representation of a perceptron. W_0 in this equation is the bias term which is added to the weighted sum to make the model generalize better. Given an input x and output y, the parameters of the model, the weights w, are learned by the model during training. The limitation of this simple perceptron is that it can only solve linear problems.

$$y = \sum_{j=1}^{d} w_j x_j + w_0 \qquad \textit{Eq. 2-1}$$

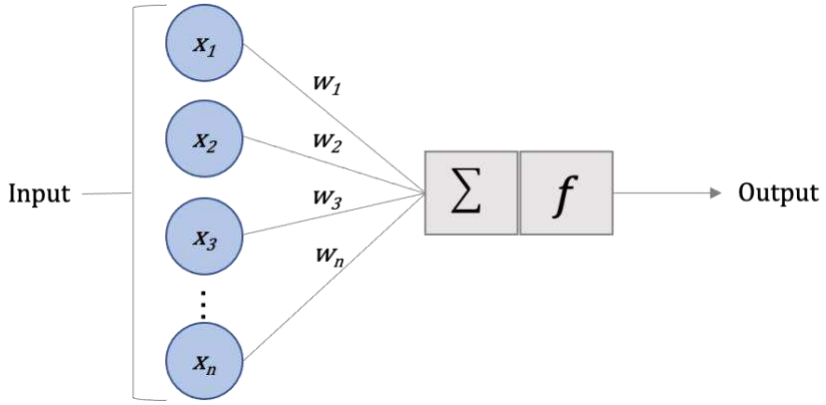


Figure 4: Mathematical representation of perceptron. x_i: input features; w_i: weight terms; f: Activation function. Bias term is not shown for simplicity.

An extended version of perceptron is a "multi-layer perceptron" (MLP) which contains one or more *hidden layers* between the input layer and the output layer, as illustrated in Figure 5 (Alpaydin, 2014). By utilizing hidden layers, this feedforward neural network can learn any nonlinear function unlike the simple perceptron. In this model, nonlinear activation functions such as sigmoid, ReLU (rectified linear unit), or tanh are used in the hidden layers to ensure that the output is a linear combination of nonlinear functions.

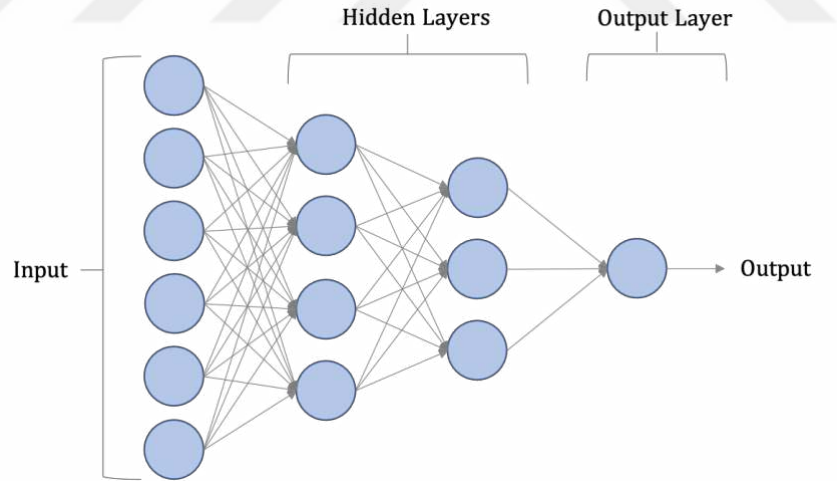


Figure 5: Representation of a multi-layer perceptron with input layer, two hidden layers, and an output layer.

Training of an MLP involves an iterative cycle of forward propagation of the input and backpropagation of the error back to the inputs, which updates the weights of all connections in the network at each iteration (Aggarwal, 2018). During forward propagation, an input vector is fed to the network and passed through the hidden layers to generate an output. In the case of classification, if the problem is binary, sigmoid is

used to compute the output label, whereas if it is a multiclass problem, softmax is used as the activation function as described in Eq. 2-2, where z is the input vector, c is the output class, and K is the total number of classes. Softmax function computes the exponential of each z_c vector and normalizes it by the sum of all z_k vectors, such that each class probability falls between 0 and 1 and they sum to 1. At this stage, the loss is computed by comparing the predicted output $\hat{y}_i$ with the ground truth label y_i for N samples based on a defined loss function such as cross-entropy loss shown in Eq. 2-3. Then, through backpropagation, the model parameters θ_t are updated based on the learning rate α and the computed gradients with respect to loss (Eq. 2-4), which indicate the direction and the magnitude of the update. The iterations continue until gradient descent converges to the closest minima.

$$sigmoid(z) = \frac{1}{1+e^{-z}}\ ; \quad softmax(z_c) = \frac{e^{z_c}}{\sum_k e^{z_k}}\ \ for\ c = 1,..,K \qquad \textit{Eq. 2-2}$$

$$\mathcal{L}oss = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) \qquad \textit{Eq. 2-3}$$

$$\theta_{t+1} = \theta_t - \alpha \frac{\partial}{\partial \theta_t}\mathcal{L}(\theta_t) \qquad \textit{Eq. 2-4}$$

While any arbitrary function can be theoretically approximated using a single hidden layer, shallow networks face the problem of overfitting to the training data and not generalizing well to the unseen test data (Aggarwal, 2018). In that respect, deep networks with multiple layers have shown great benefits in solving more complex problems as the increased number of hidden layers act as a regularizer and helps the network to learn patterns in the training data (Aggarwal, 2018). Although this increased depth makes the training process harder, the exponential growth of computational power in the recent years has led to a surge of interest in deep learning especially in the area of computer vision.

2.3 Computer Vision

2.3.1 Image Processing Techniques

Humans are equipped with exquisite vision capabilities and can recognize any object instantly as they see them. Yet, computational modelling of a such vision system is not as straightforward. Within-object variations (i.e. variations in viewpoint, illumination, occlusion, background clutter etc.) as well as within-class variations (e.g. presence of different types of cats for a cat recognition task) present challenges in a simple task of object recognition since even a minor variation of an object may hinder its recognition (Ghodrati et al., 2014; Russell & Norvig, 2010).

One traditional approach to this problem is the use of image processing to extract informative visual features and feed them into an ML model such as SVM for classification. Image processing encompasses a range of techniques and has been widely used for object recognition and detection tasks prior to the advancement of convolutional neural networks, including the field of medical imaging. In this approach, the features are extracted only once, and they remain unchanged throughout the training process unlike the features in convolutional neural networks (Shapiro & Stockman, 2000).

Digital images consist of raw pixels whose values correspond to light intensity and/or colour depending on the spectral band. While simply using raw image pixels to train a model may not be very informative, image features extracted from the raw pixels can provide much more relevant information about the content and the characteristics of an image and prevents issues resulting from within-object and within-class variations (Shapiro & Stockman, 2000). Histograms, lines, edges, and corners are some of the typical image features that are used in image processing. More complex features also exist which can provide information about shape and texture of the objects in an image.

The following are some of the powerful feature descriptors that are widely used in image processing:

- **Colour Histogram**: Histogram is a plot that shows the distribution of pixel intensities (i.e., frequencies) with pixel values on x-axis and their corresponding intensities on y-axis. The pixel values (usually range from 0 to 255) are first split into a number of bins and then the number of pixels is counted for each bin to produce a histogram. Colour histograms are similar

with the difference that they are specifically used for calculating distribution of colours in multi-spectral images. It could be considered as a stack of histograms, each of which representing pixel distribution for a colour channel such as red, green, and blue in the case of an RBG image. As they provide a summary of colour distribution in an image, they are considered to be robust to rotation and translation of objects (Shapiro & Stockman, 2000).

- **Histogram of Oriented Gradients (HOG):** The idea behind HOG is that object shape and appearance can be largely represented by the distribution of gradient orientations in an image (Dalal & Triggs, 2005; Szeliski, 2011). In this technique, the image is split into so called cells, then a histogram of gradients is computed for each cell, where pixels with large gradients receive a large weight. The gradients within each cell histogram are normalized by a larger group of cells, called blocks, across the image in order to reduce the effect of illumination variations.
- **Hu Moments:** Weighted average of pixel intensities, called image moments, are useful features for describing properties such as shape, area, and centroid of a blob in an image (Szeliski, 2011). Hu Moments are a subgroup of moment features that are shown to be invariant to scale, translation, and rotation of an image, assuming an infinite image resolution (Hu, 1962). The function computes a set of seven Hu invariants, which were introduced by Hu et al. in 1962. Hu Moments have since been applied in various areas thanks to their robustness against geometric transformations even in noisy images.
- **Gray Level Co-occurrence Matrix (GLCM):** A GLCM is a widely used texture feature descriptor developed by Haralick et al in 1973 (Haralick et al., 1973). It is based on measuring the co-occurrence of the grayscale values over an image, where each entry corresponds to the number of times a pixel value *i* is adjacent to a pixel value *j*, often normalized by the total number of comparisons. As the co-occurrence matrices are not so useful on their own, a set of statistical properties, so-called Haralick features, are computed to represent texture of an image (Haralick, 1979). These include but not limited to properties such as contrast, dissimilarity, homogeneity, angular second moment, energy, and correlation. If the features are intended to be rotationally

invariant, a common practice is to compute GLCMs at various angles and taking the average of texture properties over all angles (Hall-Beyer, 2017). GLCM features are frequently used in medical imaging.

- **Local Binary Patterns (LBP):** LBPs is another type of texture feature which was first proposed in 1994 by Ojala et al (Ojala et al., 1994). To determine the texture pattern in an angular space, an image is first divided into cells and each pixel is compared to its 8 neighbours following a circle, resulting in an 8-digit binary number for each pixel. A value of 1 is given if the neighbour's value is greater than the centre pixel and 0 otherwise. A histogram is then computed for each cell by counting the occurrences of the 8-digit codes and normalizing over the cell. One useful and popular extension of LBPs is called "uniform pattern" LBPs which implements a rotation invariant version of LBPs (Barkan et al., 2013; Ojala et al., 2002). This also reduces the length of the feature vector and simplifies the computation.

It is a common practice to extract a combination of various feature descriptors to ensure that the relevant image data are captured. Once the features are extracted, they are flattened and concatenated into a single vector, called *feature vector*, and fed into an ML model for training (Shapiro & Stockman, 2000).

2.3.2 Convolutional Neural Networks

As discussed in the previous section, feature extraction is one approach for building models that works well with image data. Yet, this approach requires domain expertise and human effort to extract relevant features for a certain use case. Another approach is the use of convolutional neural networks (CNNs) which has been widely applied to image recognition tasks and become the standard practice in computer vision in the last decade with increased computational power and availability of large datasets. Unlike structured data, image data exhibit locality where adjacent pixels are more correlated or similar than pixels that are far away, creating a spatial dependency (Aggarwal, 2018). The *convolution* operation (Figure 6) that is at the core of CNNs exploits this spatial dependency, making CNNs well-suited to image data (Aggarwal, 2018). Another important aspect of image data is its size. For instance, a typical RGB image of size 512x512 contains 786,432

pixels; yet, training a non-convolutional neural network with this size of data involves a large number of parameters to learn and also requires a huge amount of data to avoid overfitting, which is extremely inefficient. However, in CNNs, connections are established at the local level only (i.e. each unit looks at a local region of input) and weights are shared between different units in a layer so that the features that are useful in one region can be also used in other regions of the input space (Alpaydin, 2014). These two important properties *sparse connections* and *parameter sharing* make CNNs very powerful and efficient in dealing with images.

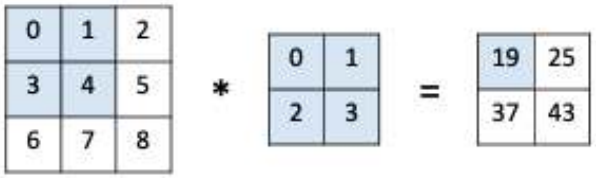


Figure 6: An example of convolution operation between a two-dimensional input of size 3x3 and a filter (kernel) of size 2x2. The filter is applied over the input and the dot product between the filter and each input window is computed to produce the output feature map.

A typical CNN architecture is generally composed of one or more convolution layers followed by pooling layers, and fully connected layers at the end, as shown in Figure 7 (Aggarwal, 2018). In convolutional layers, convolution operation is applied at all possible regions of an input by sliding a filter (i.e., weights) on the input and taking a dot product between each region of input and the filter as shown in Figure 6. As a result, a grid of values is produced to form so-called *feature map* or *activation map* whose units share the same parameters (weights and bias of the filter). Here, the filter size, stride, and application of padding to the input are some of the design choices. The number of filters used in each convolution layer determines the depth of the resulting feature map, hence the feature detecting capabilities of that layer. The convolution operation is followed by an activation function, often ReLU by default, in order to add non-linearity to the model. With these stacked convolution layers, features are learned with increasing abstraction such that earlier layers learn low-level features such as edges and later layers learn high-level features that incorporate more complex information about image – a concept which is called "hierarchical feature engineering" (Aggarwal, 2018). In contrast to the fixed feature extraction process in image processing, the features are *learned* in CNN during training.

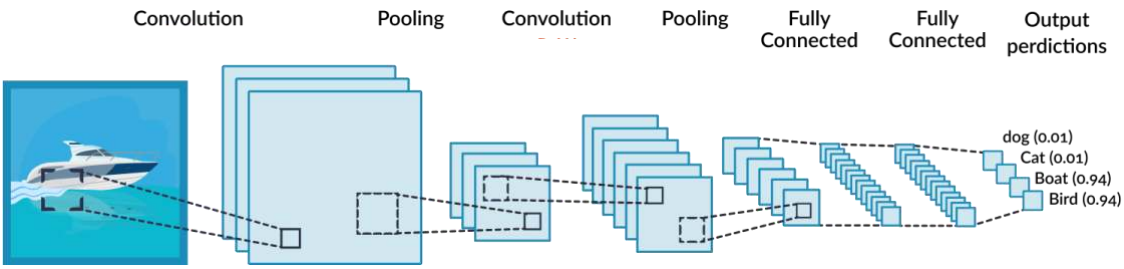


Figure 7: A typical CNN architecture composed of convolution, pooling, and fully connected layers producing prediction scores per class at the last layer. The class label which receives the highest probability score is the predicted output (MissingLink.ai, 2019).

Another essential part of CNNs is pooling, which is a type of down-sampling. The pooling layer takes the feature maps produced after the convolution operation as inputs, partitions the input into – usually non-overlapping – regions, and applies a pooling operation such as max-pooling or average-pooling. As the name implies, max-pooling operation outputs the maximum value in a region, and average-pooling outputs the average value. The pooling operation significantly reduces the spatial size of each feature map without changing the number of feature maps, which in turn helps to reduce number of model parameters and computational burden.

After consecutive layers of convolution and pooling, the resulting feature maps are flattened and passed through one or more layers of fully connected layers, as seen in feedforward neural networks in section 2.2.2. An activation function such as softmax or sigmoid is applied at the end of the architecture to provide output probabilities.

CNN Tricks & Strategies

Computer vision problems often suffer from inadequate data which may cause CNN models to overfit to the training data and not generalize well to unseen test data. The scarcity of annotated data is especially of great concern in the area of medical imaging. Different strategies have been proposed to overcome overfitting which can be identified by observing the learning curve plots such as loss or accuracy. When overfitting occurs, the performance on the validation set tends to get worse after a certain number of iterations while the performance continues to improve on the training set. Some common approaches for tackling overfitting are applying a form of regularization such as L2 regularization (also known as weight decay), using dropout layers that randomly set some neurons to zero during forward propagation, early-stopping that is the termination of training process early when the model starts overfitting, and using batch normalization

that keeps activation outputs in Gaussian range to improve gradient flow while also acting as a regularizer (Aggarwal, 2018).

Another common approach is data augmentation which introduces perturbations on the dataset (Aggarwal, 2018). The modifications may include flipping (horizontal or vertical), cropping, shifting, and rotating an image, changing brightness, contrast, or saturation of an image, adding noise, and blurring of an image. Data augmentation, if designed appropriately, can prevent overfitting to the training set and improve test accuracy.

Transfer learning is one of the most powerful techniques that is widely used when working with small datasets. In transfer learning, a model that is pretrained on a large dataset is used as a base to train a model for a given task (Aggarwal, 2018). ImageNet dataset which contains over a million images of everyday objects (~1k categories) is commonly used for transfer learning in classification tasks (Deng et al., 2009). Similarly, Microsoft COCO (Common Objects in Context) dataset contains 328k images with a total of 2.5 million labelled instances of common objects and used as a benchmark in object detection and segmentation (T. Y. Lin et al., 2014). If the dataset on which the pretrained model was trained is similar to the dataset we are working on, we can use the pretrained model as a feature extractor and customize it for our given task simply by modifying the last classifier layer(s). If the target domain is different to the source domain from which we are transferring knowledge (i.e., medical imaging vs. common object classification), we can unfreeze some or all layers of the pretrained model and train both unfrozen layers and the newly added layers – a process called "fine-tuning". The initialization of some or part of our model with pretrained weights allows us to re-use already learned low-level features in our task and significantly alleviate overfitting issues encountered with a small dataset.

Advanced CNN Architectures

The real success of CNNs first appeared with the introduction of LeNet-5 in 1998 (LeCun et al., 1998). Since then, various CNN architectures have been developed which achieved or exceeded human performance in many vision tasks. The most prominent change from the earlier models could be considered as the use of more layers, which has provided the highest performance gain (Aggarwal, 2018). Others incorporated numerous

computational tricks to improve performance. We will discuss a few of the pivotal CNN architectures below:

- **Inception:** The inception architecture first emerged in 2014 with GoogLeNet (Szegedy et al., 2015). The key building block of the architecture is the inception module, whose name was inspired from the notion of *network within a network*. The inception concept is based on using convolutions of different sizes to capture different levels of information in an image – bigger filter sizes capturing information from a larger spatial area while smaller filter sizes capturing more detailed information. In a typical inception module, three different filter sizes such as 1x1, 3x3, and 5x5 are used in parallel as shown in Figure 8, similar to a small network by itself. 1x1 convolution is often used prior to the convolution operations with larger filter sizes in order to decrease the depth of the feature map, as a way to control computational burden. Because these 1x1 convolutions improves efficiency by reducing the output dimensions, they are also called "bottleneck operations". As the filters are learned through training, the model can learn which of the different sizes of filters and paths are more useful. Another aspect of Inception architecture is that it uses average pooling instead of a stack of fully connected layers in the output layer. In particular applications, the predicted class label is sufficient as the model output and this information can be provided compactly by the average pooled representation, which saves computational efficiency. Despite this, the overall architecture is still considered computationally complex. Further improvements have been made in the succeeding versions of GoogLeNet such as incorporation of residual connections (Szegedy et al., 2016, 2017). The authors showed that combining the inception architecture with residual connections (explained in ResNet architecture below) in Inception-ResNets achieves outstanding performance on ImageNet dataset (Szegedy et al., 2017).

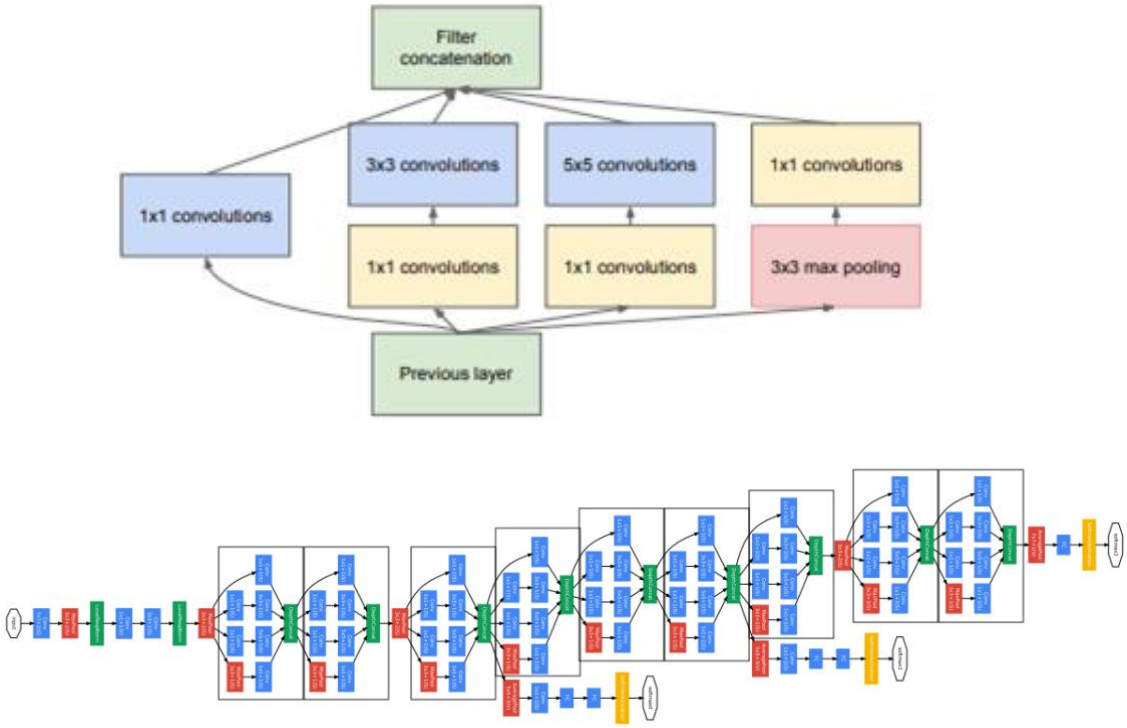


Figure 8: (a) An example of Inception module in GoogLeNet. (b) GoogLeNet architecture (Szegedy et al., 2015).

- **ResNet:** Residual Neural Network (ResNet) is a type of neural network that uses residual (skip) connections between layers (He et al., 2016). One idea behind this is to prevent the vanishing gradient problem that occurs in deep networks due to large number of operations. As shown in Figure 9, the skip connection simply takes the activation from a previous layer and add it to an upstream layer. This creates numerous paths of different length from input to output, allowing the model to propagate through different paths depending on the complexity of the input, without restricting gradient flow. The use of skip connections not only improve accuracy but also speeds up training. The skip connections form the so-called residual blocks which are put together to build the whole model. In the original paper, authors implemented ResNet models with 34, 50, 101, and 152 layers and tested on ImageNet dataset (He et al., 2016). ResNet with 152 layers showed the highest performance among the four architectures.

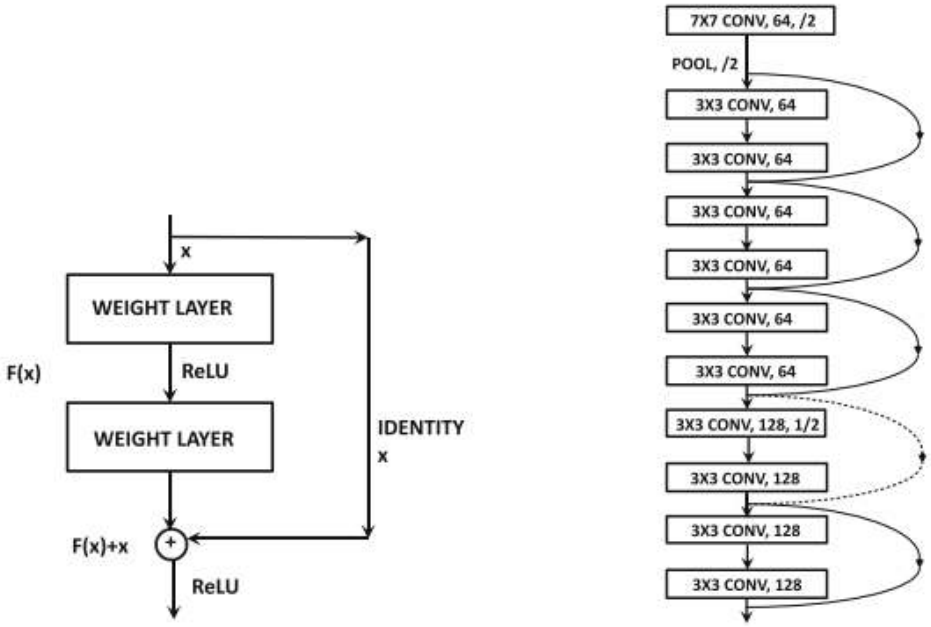


Figure 9: (a) Residual block with a skip connection. (b) Partial architecture of ResNet (Aggarwal, 2018) .

- **DenseNet**: Dense Convolutional Network (DenseNet) has a similar architecture to ResNet but with the difference that it contains connections between each layer and every other layer (Huang et al., 2017). As shown in Figure 10, the feature maps produced at each layer is used as an input by all upstream layers, increasing the variation of the input in the upstream layers and promoting feature reuse. These dense connections help DenseNet to deal with vanishing gradient problems and provide a great performance gain, similar to ResNet. Various DenseNet architectures were tested in the original paper, among which the largest model, DenseNet-161, achieved the highest accuracy (Huang et al., 2017).

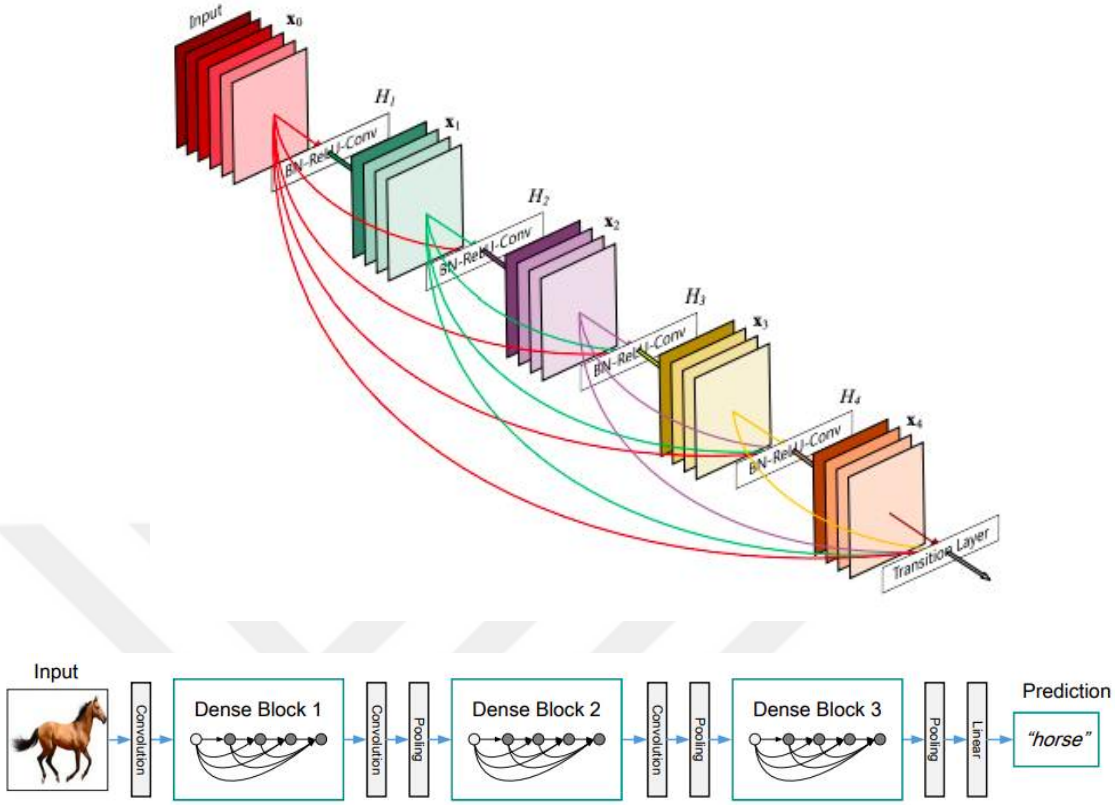


Figure 10: (a) Dense block with 5-layers. (b) DenseNet with three dense blocks (Huang et al., 2017).

- **EfficientNet**: EfficientNets are a family of CNN models recently developed by Google Brain team in 2019 (Tan & Le, 2019). After extensive studies on model scaling, the team concluded that better performance can be achieved by careful selection of depth, width, and resolution of a network based on the available resources. As demonstrated in Figure 11, the authors proposed a compound scaling method which performs scaling on all width, depth, and resolution dimensions of a baseline network using appropriate scaling coefficients determined through grid search. EfficientNets are available in various scales built on a baseline network that uses mobile inverted bottleneck convolutions as shown in Figure 12. The authors show that this optimization of accuracy and efficiency, the latter measured in terms of floating-point operations per second (FLOPS), yields an improved performance with fewer parameters compared to other CNN architectures.

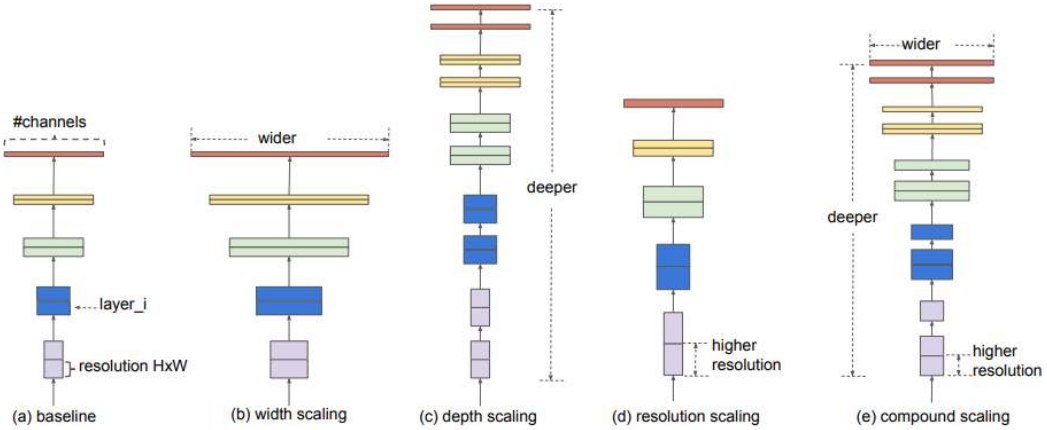


Figure 11: Model scaling strategy of EfficientNets. Baseline network structure (a); conventional scaling of only one dimension at a time (b),(c),(d); proposed compound scaling method which scales all dimensions (Tan & Le, 2019).

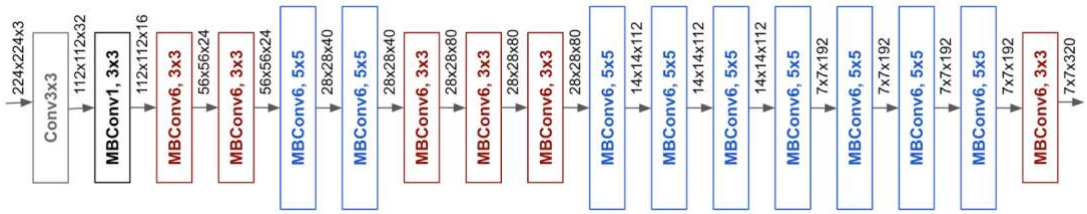


Figure 12: Architecture of EfficientNet-b0 which uses mobile inverted bottleneck convolutions as its building block.

The CNN architectures that were discussed so far are typically used for classification tasks. Different types of models were also developed using CNNs for other object recognition tasks such as object detection or localization and segmentation (Figure 13), which are discussed in the following sections.

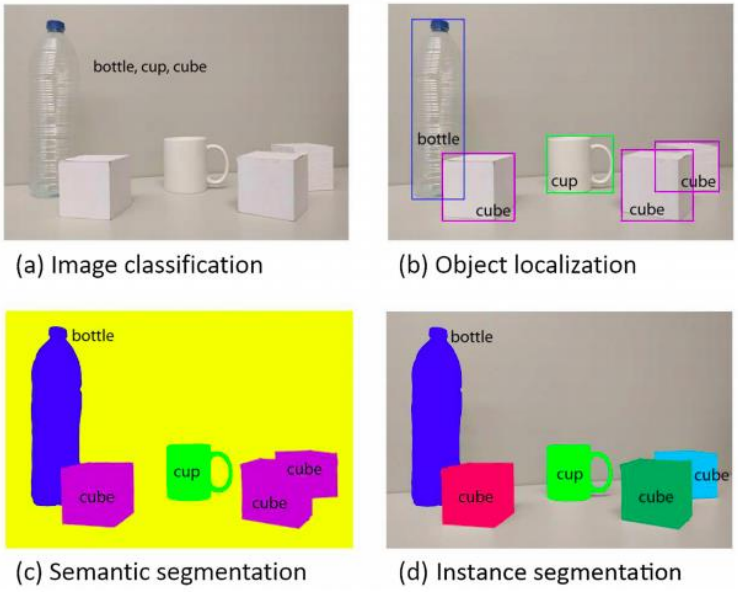


Figure 13: Types of object recognition tasks (Garcia-Garcia et al., 2018).

Object Detection

Object detection is an area of computer vision that is concerned with detecting objects in images and videos. Until recently, methods based on feature descriptors such as Histogram of Oriented Gradients (HOG) and Scale Invariant Feature Transform (SIFT) had been the primary approach for object detection (Dalal & Triggs, 2005; Lowe, 2004). However, these methods require domain expertise and a great deal of handcrafting of features (Z. Q. Zhao et al., 2019). Subsequently, deep learning-based methods emerged using CNNs that can *learn* feature representations from images and can achieve remarkable results.

Deep learning-based object detection algorithms usually fall into two categories (Z. Q. Zhao et al., 2019). One category is the region-based frameworks, and the other is regression/classification-based frameworks. In both cases, the aim is to find where objects are located in an image and what class / category they belong to (Z. Q. Zhao et al., 2019). The term "object localization" can be used if the aim is to detect objects of only *single* class. The output of object detection algorithms is a bounding box for each detected object along with a confidence score for the corresponding detection.

Region-Based Detectors

Region-based detectors involve generating region proposals followed by classification of the generated proposals. The pioneering architecture of this group, R-CNN, employs a search algorithm to generate ~2000 region proposals per image which are then cropped and passed through a pretrained CNN for feature extraction, as shown in Figure 14 (Girshick et al., 2014). A fixed length of feature vector is produced for each proposal, which is then fed into an SVM for classifying the region. The bounding boxes are finalized by box regression to adjust size and a non-maximum suppression (NMS) operation to eliminate redundant boxes. Therefore, R-CNN involves three stages of training – training of CNN for feature extraction, training of SVM for classifying regions, and training of bounding box regressors.

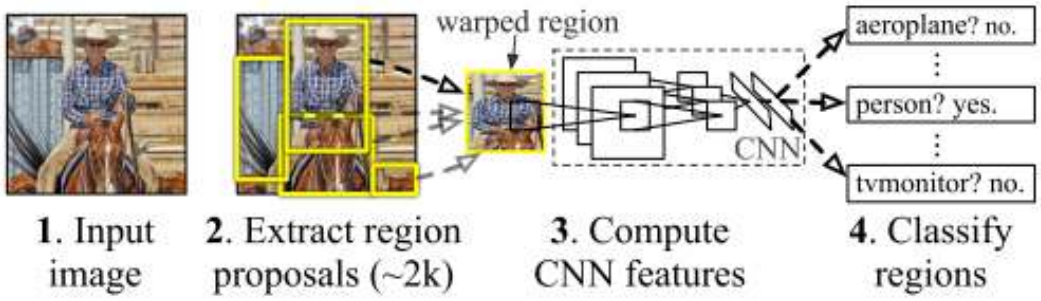


Figure 14: Overview of R-CNN architecture (Girshick et al., 2014).

Fast R-CNN is an improved version of R-CNN, which combines multiple stages of training into one (Girshick, 2015). The main challenge in R-CNN is the computational inefficiency brought by extracting features for each region proposal independently. Fast R-CNN overcomes this challenge by performing CNN on the whole image to extract features in once along with implementing a region of interest (ROI) pooling layer (see Figure 15). As the generated proposals can have different sizes and aspect ratios, the ROI pooling ensures that a fixed length of feature vector is produced for each proposal. For each ROI, softmax classifier outputs class probabilities while bounding box regressor outputs refined bounding box coordinates. Furthermore, it employs a multi-task loss that combines classification loss and regression loss in one loss function.

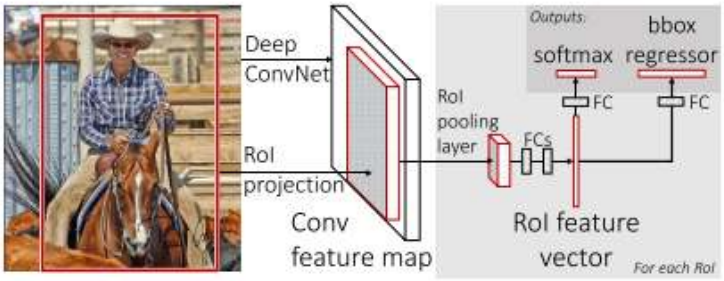


Figure 15: Overview of Fast R-CNN architecture (Girshick, 2015).

A further improvement of Fast R-CNN came with the replacement of selective search algorithm with a region proposal network (RPN) to generate proposals (Ren et al., 2017). In this model called Faster R-CNN, region proposals are generated by running RPN over the last feature map output of the backbone in a sliding window fashion as shown in Figure 16. In order to be able to produce proposals of various scales and aspect ratios, RPNs utilize “anchor boxes” that act as reference boxes with predefined scale and aspect ratios. By default, 9 anchor boxes (3 scales and 3 aspect ratios) are used for each sliding

window position. In the detection stage, Fast R-CNN model is used whose convolutional layers are shared with RPN to provide a unified network in Faster R-CNN (see

Figure 17). In regard to its implementation, the RPN is trained with the rest of the network and the loss function contains the loss from object detection stage as well as the loss from RPN, encouraging the model to learn to generate fewer but more precise region proposals. The use of RPN provides a significant boost of efficiency and speed in Faster R-CNN model.

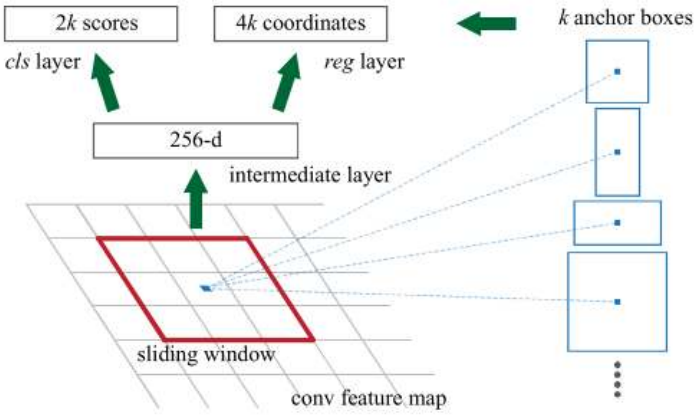


Figure 16: Region Proposal Network (RPN) in Faster R-CNN. RPN slides over the last convolutional feature map and uses k anchor boxes for each sliding window to generate region proposals (k=9 by default). The resulting feature vector is fed into a classification layer (*cls* layer) and a box regression layer (*reg* layer) (Ren et al., 2017).

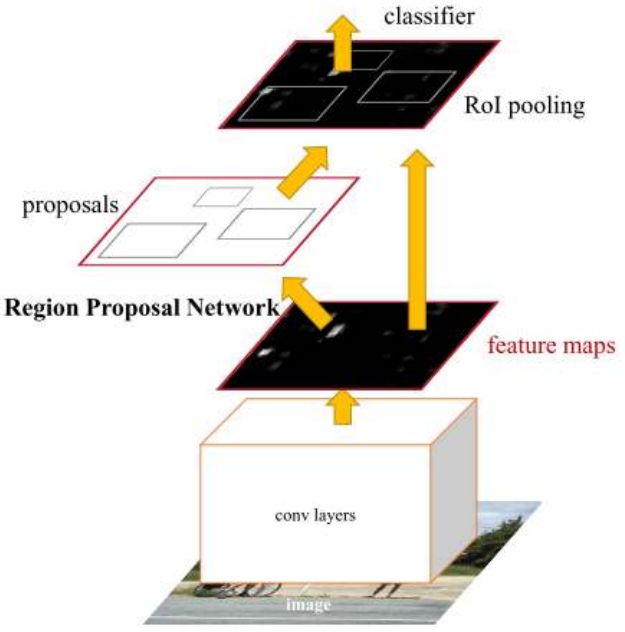


Figure 17: Overview of Faster R-CNN architecture (Ren et al., 2017).

While the region-based detectors provide relatively accurate predictions, one major limitation (in earlier models especially) is that they are computationally very expensive due to the large number of proposals that need to be generated for each image (Z. Q. Zhao et al., 2019). Even though the use of RPN increases efficiency in Faster R-CNN model, it still lags behind in terms of speed and complexity due to being a two-stage detector.

Regression / Classification-based Frameworks

Regression/classification-based frameworks implements a unified approach for object detection by combining localization and classification tasks (Z. Q. Zhao et al., 2019). The most recognized of these models is YOLO (You-Only-Look-Once) which was proposed by Redmon et al. in 2016 (Redmon et al., 2016). YOLO uses a fully convolutional neural network (F-CNN) with 24 convolution layers and 2 fully connected layers at the end. It predicts both bounding box coordinates and class probabilities for the predictions in a single stage. As illustrated in Figure 18, the image is first divided into a grid of size $S \times S$ and for each grid cell B bounding boxes are predicted along with corresponding confidence scores and C conditional class probabilities. Figure 19 shows that the predictions for each grid cell is represented as a tensor of size $(B * 5 + C)$ where each box consists of (x, y) coordinates for the center of the box, (w, h) for the width and height of the box, and a confidence score. The confidence score (objectness score) indicates how likely there is an object in the box and how accurate the prediction is based on intersection over union (IoU) between the ground truth box and the predicted box, which is formulated as $Pr(Object) * IOU_{pred}^{truth}$. During training, only one object (the one which has the highest score) is detected per grid cell regardless of the number of bounding boxes B to enable specialization of the bounding boxes in certain classes, sizes, and aspect ratios. At test time, class probabilities are multiplied with the confidence score to produce class-specific confidence scores for each bounding box, given as $Pr(Class_i|Object) * Pr(Object) * IOU_{pred}^{truth}$. Only the predictions that have a confidence score higher than a set threshold are retained. If there are multiple overlapping boxes for the same object, redundant detections are eliminated by NMS to produce the final boxes. NMS works by computing the IoU between the predicted box with the highest score and any other predicted box and eliminating those boxes whose IoU is higher than a set threshold.

In general terms, YOLO loss function is computed as the sum of the following components:

- Objectness loss (based on box confidence scores)
- Localization loss (based on IOU_{pred}^{truth})
- Classification loss (based on class probabilities)

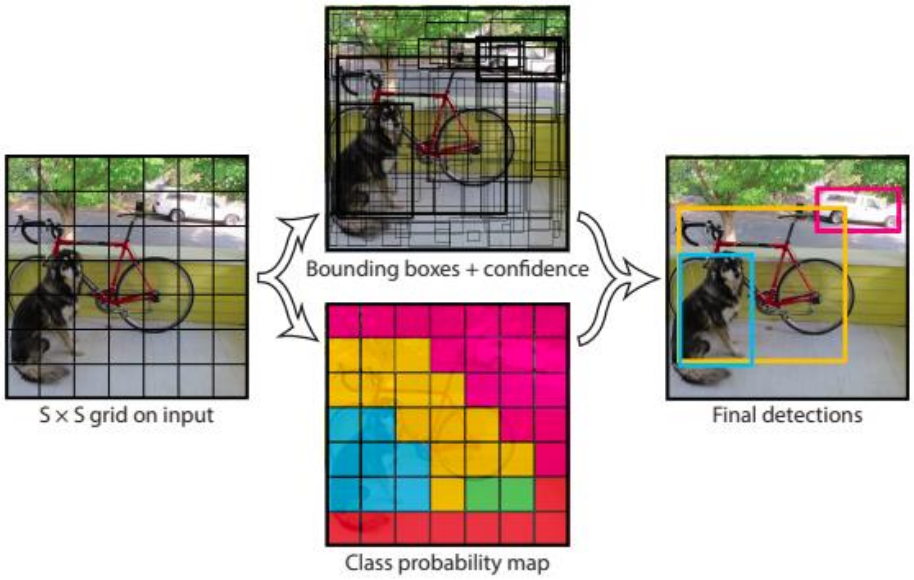


Figure 18: Overview of YOLO model.(Redmon et al., 2016)

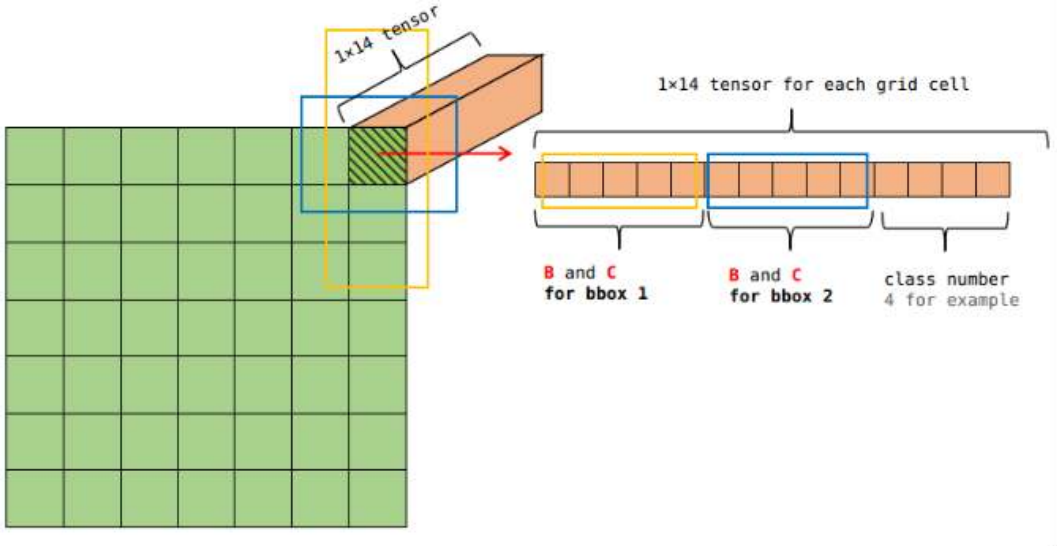


Figure 19: Representation of YOLO predictions for each grid cell. In this example, there are 2 bounding boxes, each of which contains 4 coordinate points (x,y,w,h) and a confidence score (Handuo, 2018).

An improved version, YOLOv2, was later introduced with some new strategies to improve accuracy (Redmon & Farhadi, 2017). Firstly, batch normalization is added to

convolution layers, which eliminates the need for using dropouts and increases the accuracy. Secondly, in the original YOLO, the classifier is pretrained on input sizes of 224×224. In YOLOv2, the resolution of the classifier is increased by first training the classifier on 224×224 images and then fine-tuning it with 448×448 images for a few more epochs. Another modification is the adoption of "anchor boxes" as used in Faster R-CNN. Instead of predicting arbitrary boxes that try to specialize in detecting certain classes, the authors define 5 anchor boxes with certain size and aspect ratios and make predictions on the offsets to each anchor box (see Figure 20). The predicted offset values for each bounding box are constrained by applying sigmoid function so that the predicted location coordinates fall between 0 and 1 and are relative to the grid cell location. Constraining the offset values makes the training process much more stable, although it doesn't bring any improvement to accuracy. Furthermore, in order to decide on the number and the dimensions of the anchor boxes, k-means clustering algorithm is run on the training set. A set of anchor boxes is selected which provides a good level of IoU with the ground-truth boxes while keeping model complexity at a minimum. Finally, YOLOv2 allows multi-scale training by resizing the network every few iterations to a new image size – a multiple of 32 within the range of 320 to 608. This makes the model robust to images of different resolutions.

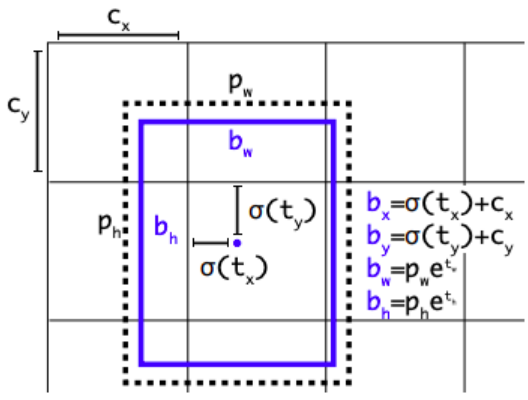
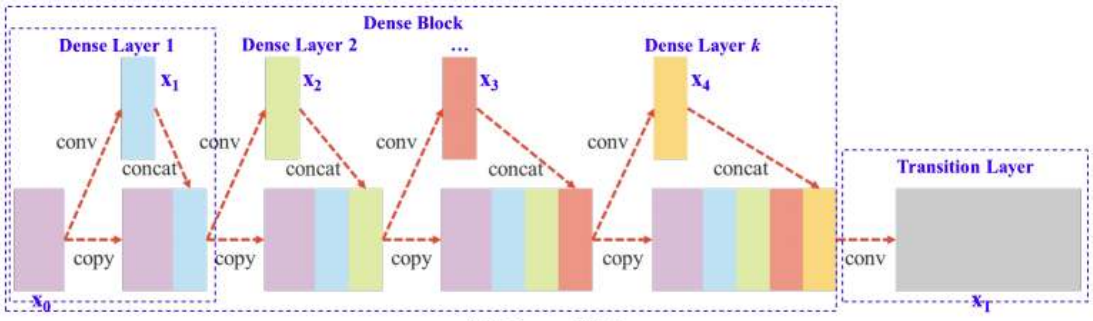


Figure 20:Prediction of bounding box coordinates using anchors. 4 coordinates (t_x, t_y, t_w, t_h) are predicted for each bounding box. The width (b_w) and height (b_h) of the prediction (blue box) are calculated as offsets from anchor's (dotted box) width and height (p_w, p_h). The center coordinates of the prediction (b_x, b_y) are calculated by applying sigmoid function on t_x and t_y and are relative to the grid cell location (c_x, c_y). (Redmon & Farhadi, 2017)

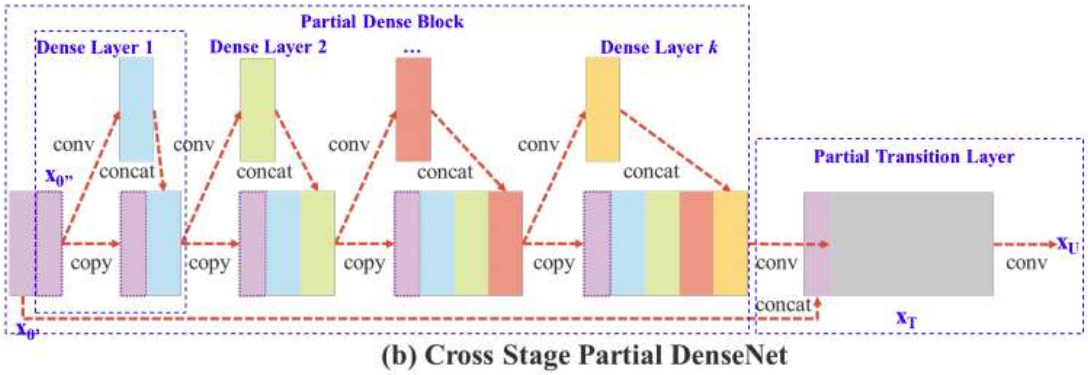
YOLOv3, which was proposed in 2018, makes further improvements to YOLO architecture (Redmon & Farhadi, 2018). In YOLOv3, softmax function that is used for

producing class scores is replaced with independent logistic classifiers. With this multi-label classification approach, the predicted class labels don't have to be mutually exclusive. In line with this change, mean square error is replaced with binary cross-entropy loss for calculating the classification loss for each label. Another change in YOLOv3 is the way bounding box predictions are evaluated. Logistic regression is used for calculating the box confidence scores, which should give 1 for an anchor that overlaps the ground truth object the most among all anchors and ignore other predictions that overlap the ground truth more than 50%, similar to Faster R-CNN (Ren et al., 2017). Unassigned anchor boxes don't bring any localization or classification loss but only impact confidence loss. One major change in YOLOv3 is the use of 3 different scales for feature extraction, similar to Feature Pyramid Networks (FPN) (T.-Y. Lin et al., 2016). Combining feature maps from different scales provides better semantic information with more detail and enhances performance on smaller objects. Lastly, a new backbone (Darknet53) is used for feature extraction, which involves consecutive 3x3 and 1x1 convolution layers as well as skip connections as in residual networks.

The most recent of YOLO descendants is YOLOv4 in which numerous, both known and novel, ideas are incorporated for better speed and accuracy (Bochkovskiy et al., 2020). For feature extraction, the authors evaluate various backbones and based on experimental results they implement CSPDarknet53 as the backbone, which uses the so called "cross-stage partial (CSP) connections" motivated by DenseNets (Wang et al., 2020).



(a) DenseNet



(b) Cross Stage Partial DenseNet

Figure 21: (a) DenseNet block. (b) DenseNet block with cross-stage partial connections (Wang et al., 2020).

The authors also implement a modified version of the Path Aggregation Network (PANet) (Liu et al., 2018) for aggregating the feature maps produced in different layers of the backbone, instead of the FPN which was used in YOLOv3. Furthermore, in order to enhance receptive field and select the most significant features from the backbone, Spatial Pyramid Pooling (SPP) block (He et al., 2014) is added over the backbone. Additionally, the authors use a modified version of Spatial Attention Module (SAM) block (Woo et al., 2018) to improve performance of the model. For detection, YOLOv4 uses the same dense prediction head as YOLOv3.

In YOLOv4, the authors implement "Bag of Freebies" that further improve the performance of the model without compromising on inference speed (Bochkovskiy et al., 2020). Some of these strategies are related to data augmentation such as CutMix (cutting a part of one image and paste it to another) and were already known to computer vision community (Yun et al., 2019). The main contribution in YOLOv4 is mosaic augmentation which mixes 4 training images at different scales into one and helps the model to detect smaller objects. Furthermore, the authors implement class label smoothing to represent class labels for bounding boxes in a way to prevent overfitting (Szegedy et al., 2016). With this strategy, the upper limit of class prediction is lowered from 1.0 to a value such as 0.9. For detector part, YOLOv4 incorporates Complete IoU (CIoU) loss which the authors found achieves better convergence on box regression. In addition to the typical IoU computation, CIoU loss takes into account the distance between the predicted box and the ground-truth and aspect ratio (Zheng et al., 2020). Furthermore, a genetic algorithm is utilized in YOLOv4 for selecting optimal hyperparameters.

The authors further explore a variety of strategies called “Bag of Specials” that can provide performance gains while increasing inference time to some degree (Bochkovskiy et al., 2020). Among these, CSP connections, PAN, SPP, and SAM blocks are the main contributors to the performance gain as discussed above. In YOLOv4, the authors investigate the use of Mish activation function to address the vanishing gradient problem instead of the common choice of ReLU (Misra, 2019). Another contribution is the use of multi-input weighted residual connections (MiWRC) in the backbone part, which is inspired by the weighted bi-directional FPN used in EfficientDet architecture (Tan et al., 2020). By using MiWRC, features with different resolutions are weighted differently so that they contribute to the output feature more equally. Finally, with regards to the processing of box predictions, YOLOv4 deploys Distance IoU-NMS (Zheng et al., 2020) to suppress redundant boxes and select the best box.

YOLO offers very fast predictions compared to most two-stage detectors thanks to its simple architecture while providing reasonably accurate results. This makes it a popular choice for real-time applications. The latest YOLOv4 especially outperformed most state-of-the-art detectors and took much attention in the computer vision community.

Semantic Segmentation

Semantic segmentation is one of the image recognition tasks which deals with assigning each pixel of an image to a particular class (Garcia-Garcia et al., 2018). It can be considered as an extension to object detection as it provides a pixel-level mask for an image. As shown in Figure 13, semantic segmentation offers a finer inference compared to the object detection task which provides the spatial location of the objects in an image using bounding boxes. One important note is that semantic segmentation does not differentiate between different instances of the same object class i.e., it assigns the same class label to the pixels of different instances of the same object.

The most prominent models of semantic segmentation are built on Fully Convolutional Neural Network (F-CNN) or their variants. F-CNN transforms a typical CNN built for classification by substituting the fully connected layers with convolutional layers in order to produce pixel-level dense predictions, as illustrated in Figure 22 (Long et al., 2014). The part of a segmentation model utilizing F-CNN produces a low-resolution image representation in the form of a feature map and is called the *encoder*. The

subsequent part of the segmentation model, called the *decoder*, aims to produce pixel-level predictions either by mapping this low-resolution feature map to a high-resolution segmentation mask or by learning to decode the feature map.

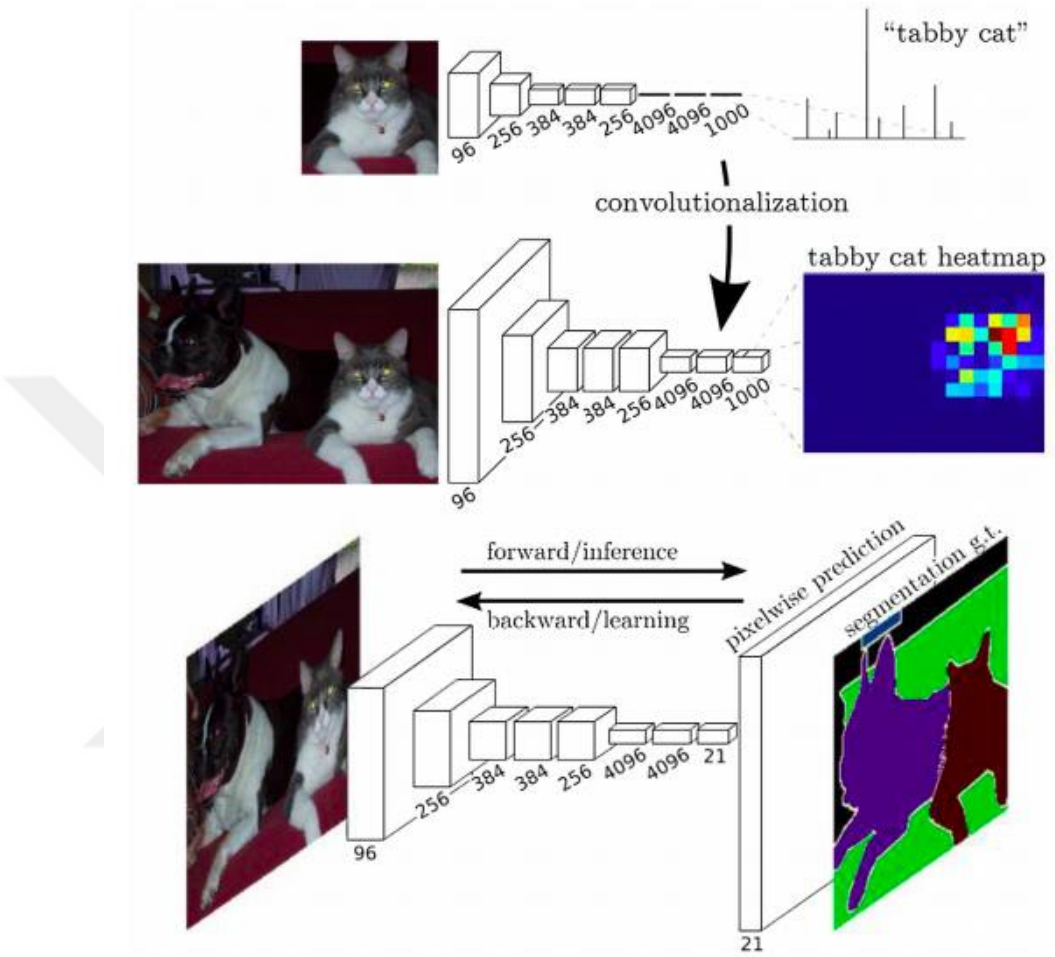


Figure 22: Formation of Fully Convolutional Neural Network (Long et al., 2014)

U-Net

U-Net is a well-known segmentation model developed by Ronneberger et al. for biomedical image segmentation (Ronneberger et al., 2015). It is built on F-CNN architecture and consists of a contracting *encoder* part and an expanding *decoder* part as shown in Figure 23. The encoder part is a typical convolutional network used in F-CNNs and employs 2x2 convolution operations (no padding) followed by ReLU and 2x2 max-pooling. In the decoder part, the feature maps are upsampled by "up-convolution" operation (upsampling followed by 2x2 convolution), which is followed by concatenation with a corresponding feature map from the encoder part and 3x3 convolutions followed

by ReLU. The up-convolution operation, also called deconvolution or transpose convolution, doubles the size of the feature map while halving its depth at each step. The Figure 24 shows the up-convolution operation for two-dimensional input. By using upsampling operations in the decoder part, the resolution of the output is increased, allowing for more precise segmentations compared to the conventional F-CNN architecture. Furthermore, the combination of upsampled feature maps with the high-resolution feature maps from the encoder part through skip connections helps the model to learn relevant features from the encoder part and produce even more precise outputs. Finally, output segmentation map, whose depth equals to the number of classes, is produced by applying 1x1 convolution to the feature map in the last layer.

The authors of U-Net achieved remarkable results on a very small dataset of microscopy images (Ronneberger et al., 2015) and since then U-Net has been applied in a wide range of applications. Despite being one of the early models of semantic segmentation that use deep learning, U-Net is still widely used thanks to its simple and efficient architecture and relatively good segmentation accuracy.

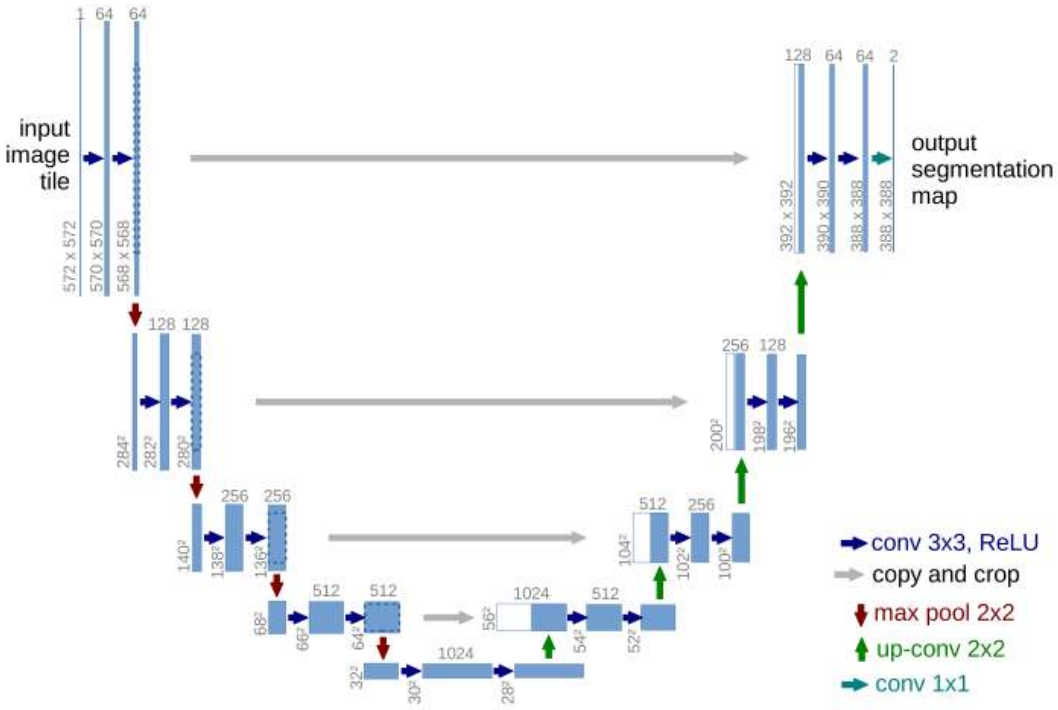


Figure 23: U-Net architecture. The blue boxes correspond to the feature maps and the white boxes are the copied feature maps. Different type of operations used in the architecture are shown with arrows as described in the bottom right section (Ronneberger et al., 2015).

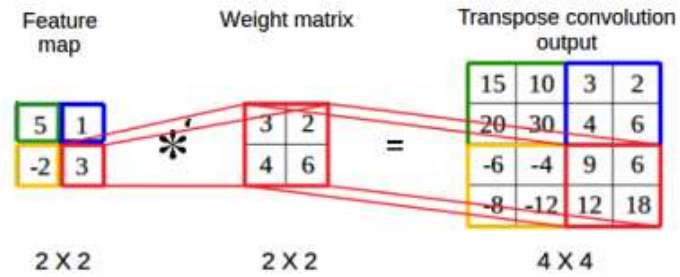


Figure 24: An example of up-convolution (transpose convolution) operation on two-dimensional feature map with 2x2 filter and stride of 2. Each value of the feature map is multiplied with weight matrix element-wise to produce values for a corresponding window in the output. (Gurumurthy et al., 2019)

Instance Segmentation

As discussed in the previous section, semantic segmentation does not differentiate between different instances of the same object class. Yet, instance-level recognition can be desirable in situations where we want to locate and identify each instance separately. Instance segmentation is an object recognition task which combines object detection and semantic segmentation tasks (Hafiz & Bhat, 2020). Hence, it provides a different label for each instance as it detects all objects in an image and segments each of those instances simultaneously.

Mask R-CNN is one of the most well-known instance segmentation frameworks, which was developed by Facebook AI Research (He et al., 2017). It extends Faster R-CNN by adding a branch for pixel-level segmentation which predicts binary masks for each ROI using F-CNN, as can be seen in Figure 25. In order to retain the spatial information of ROI feature maps for segmentation task, the ROI pooling layer in Faster R-CNN model is replaced with a new ROI align layer. ROI align layer not only works to align extracted features with each ROI, but also helps to improve mask accuracy significantly. As for the loss function, Mask R-CNN just adds segmentation loss to the loss used in Faster R-CNN; thus, the loss is defined as $\mathcal{L}_{cls} + \mathcal{L}_{box} + \mathcal{L}_{mask}$ for each ROI. Since a binary mask is produced for each class on each ROI, average binary cross-entropy loss is used for $\mathcal{L}_{mask}$. Another change in Mask R-CNN is that it implements FPN for the backbone in addition to ResNets. The use of FPN improves the accuracy of the model considerably as it extracts ROI features of different scales from different levels of the network. Overall, Mask R-CNN achieves good accuracy for both box and mask

predictions and runs relatively fast at 5 frame-per-second (FPS) – although not completely optimized for speed.

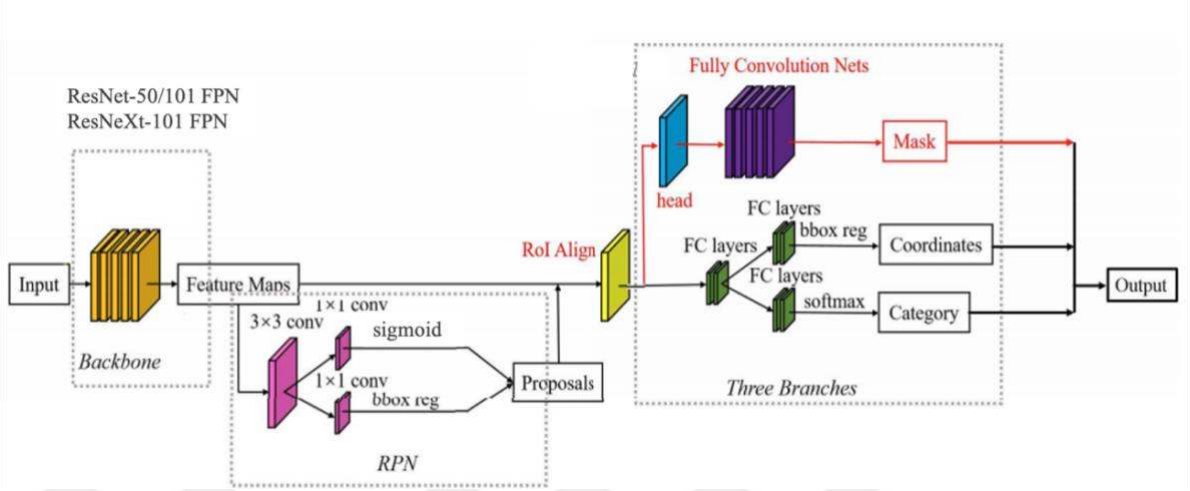


Figure 25: The Mask R-CNN framework (Figure adapted from (Jiao & Zhao, 2019)).

2.4 Related Works and Current Technology

In the field of medical image analysis, machine learning techniques have long gained attention as a way to aid computer-assisted diagnosis. With the progress in CNN architectures over the last decade, the methodologies have been applied to various areas of medical imaging especially for the image classification task. CNNs have been shown to be successful in medical tasks such as detection of diabetic retinopathy from retinal fundus photographs (Raman et al., 2019), classification of gastrointestinal diseases from endoscopy images (Borgli et al., 2019), classification of head and neck cancer using hyperspectral imaging (Halicek et al., 2017), and identifying skin cancer from clinical images – with higher than dermatologist-level performance (Esteva et al., 2017; Rezvantalab et al., 2018), to name a few.

The literature on image-based automated diagnosis of oral cancer is largely focused on histopathological images (Aubreville et al., 2017; Gupta et al., 2019; Jaremenko et al., 2015; Wieslander et al., 2017). Furthermore, there are studies conducted using various imaging technologies such as computed tomography (Xu et al., 2019), optical coherence tomography (Heidari et al., 2018; Wilder-Smith et al., 2009), hyperspectral imaging (Jeyaraj & Samuel Nadar, 2019), and autofluorescence imaging (Rahman et al., 2010; Roblyer et al., 2009; Song et al., 2018; Uthoff et al., 2019; Uthoff, Song, et al., 2018;

Uthoff, Wilder-Smith, et al., 2018). Among them, the use of autofluorescence imaging has been widely explored for the detection of oral neoplastic lesions as it has been shown by several studies that oral tissue with increased dysplasia emit decreased fluorescence under fluorescence excitation at 400 nm due to molecular and morphological changes accompanying neoplastic processes, which can be useful for distinguishing neoplastic lesions in the oral cavity (Lane et al., 2006; Svistun et al., 2004). Early research with the use of autofluorescence imaging in oral cancer focused on image processing and simple classification algorithms to identify neoplastic lesions (Rahman et al., 2010; Roblyer et al., 2009). In a pilot study with 76 patients and 33 normal volunteers, Rahman et al. identified cancerous or suspiciously neoplastic tissues with 90% sensitivity and 87% specificity with an LDA algorithm based on the ratio of red-to-green fluorescence emitted from oral mucosa, which is used as a means of measuring fluorescence loss (Rahman et al., 2010). In 2011, Matsumoto proposed a hand-held device based on autofluorescence, called VELscope system, to facilitate the detection of precancerous and cancerous lesions (Matsumoto, 2011). Yet, the detection depends on visual examination of the resulting fluorescence, which requires user training for interpretation. Moreover, there is a lack of evidence for the reliability of autofluorescence imaging according to existing literature and guidelines (Lingen et al., 2017; Macey et al., 2015). In more recent work, Uthoff et al. developed a dual-modality smartphone-phone based imaging system and employed deep learning to classify normal and precancerous or cancerous tissue (Song et al., 2018; Uthoff et al., 2019; Uthoff, Song, et al., 2018; Uthoff, Wilder-Smith, et al., 2018). The authors collected both autofluorescence and white-light close-up images (dual-modal) of lesions from patients using the proposed device and combined them into single three-channel images, which were then fed into a VGG-based CNN model. With 170 image pairs, the authors achieved a 4-fold cross-validation accuracy of 86.9% (not specified test set) for this binary image classification task. Nevertheless, there is no clear evidence on whether the autofluorescence imaging is able to distinguish precancerous or cancerous lesions from benign lesions (Ilhan et al., 2020).

While the techniques discussed so far rely on a special imaging equipment, there has been a few studies performed with white-light photographic images. In an early study from 2013, the authors extracted a selection of grey level co-occurrence matrix (GLCM) and grey level run-length (GLRL) texture features from digital camera images and employed an artificial neural network for classification of oral carcinomas into six groups

arising in different areas of oral cavity (Thomas et al., 2013). In 2017, Anantharaman et al. employed deep learning to detect mouth sores in an attempt to provide a mobile assessment tool called "Oro Vision" for orofacial diseases (Anantharaman et al., 2017). The authors trained an Inception-v3 model on 75 close-up images to classify them into 2 classes, namely canker sore (aphthous ulcer) and cold sore, and achieved an accuracy of 66% on a test set of 6 images. While the results were not considerable due to the use of a small dataset, the study demonstrated preliminary feasibility of using CNNs for orofacial diseases. Recently, Shamim et al. evaluated the efficacy of different CNN models with close-up photographic images of tongue lesions (Shamim et al., 2019). The authors claimed that they achieved 97.5% validation accuracy on binary classification task (benign vs precancerous) using VGG19 and 96.7% validation accuracy on multiclass classification task (4 benign and 1 precancerous type) using ResNet50 with a dataset of size 200 and 300 images, respectively for each task. However, only tongue lesions were included in the study and the models' performance on an external test set was not reported.

In addition to image classification, different CNN architectures have been successfully employed for object detection and segmentation in areas related to oral cancer, such as segmentation of gingival diseases from fluorescence images using convolutional autoencoder (Rana et al., 2017), detection of oesophageal abnormality from endoscopic images using Faster R-CNN (Ghatwary et al., 2019), and tooth segmentation and classification on panoramic X-ray images using a combination of Mask R-CNN with U-Net (S. Zhao et al., 2020). As an extension to their initial work focused on classification of mouth sores (Anantharaman et al., 2017), Anantharaman et al. implemented Mask R-CNN with ResNet backbone for detection and segmentation of mouth sores (Anantharaman et al., 2019). After training the network heads only on a set of 30 images, the authors reported to have achieved an average pixel accuracy (dice coefficient score) of 0.744 on the test set composed of 10 images. Although the scope of the study was limited to mouth sores, it paves the way for the detection of other pathological oral lesions using Mask R-CNN.

Very recently, two articles have been published assessing the efficacy of CNN-based models for automated detection and classification of oral lesions from photographic images. Welikala et al. implemented Faster R-CNN for detection and ResNet101 for classification on a dataset of 2155 oral cavity images with labels based on referral

decision (Welikala et al., 2020). The images were labelled as ‘no lesion’, ‘no referral needed’, ‘refer for other reasons’, ‘refer - low risk OPMD’, or ‘refer - cancer/high risk OPMD’. The images with no lesions were not used in detection, resulting in 1433 images for detection tasks. Each model was evaluated on 3 separate tasks respectively. In classification experiments with ResNet101 of entire images, binary classification of ‘no lesion’ vs ‘lesion’ (includes all classes except ‘no lesion’) achieved an F_1-score of 87.07%; binary classification of ‘non-referral’ (includes ‘no lesion’ and ‘no referral needed’ classes) vs ‘referral’ (includes ‘refer for other reasons’, ‘refer - low risk OPMD’, and ‘refer - cancer/high risk OPMD’ classes) achieved an F_1-score of 78.30%; and 5-class classification achieved a macro-average F_1-score of 50.57% on test set of 204 images. In experiments with Faster R-CNN on 1433 images (after removal of images with no lesions), detection of ‘lesion’ (includes all 4 lesion classes) achieved an F_1-score of 41.35%; detection of ‘non-referral’ (includes ‘no referral needed’ class) and ‘referral’ (includes ‘refer for other reasons’, ‘refer - low risk OPMD’, or ‘refer - cancer/high risk OPMD’ classes) lesions achieved a macro-average F_1-score of 32.41%; and 4-class classification achieved a macro-average F_1-score of 24.50% on test set of 148 images. The classification results were encouraging especially for binary classification tasks, yet the models did not perform well on multi-class classification of different types of referral decision. The authors did not achieve very good results on detection tasks which leaves plenty of room for improvement.

Fu et al. proposed to use cascaded (two-step) CNNs to detect OCSCC lesions from photographic images of biopsy-proven OCSCC lesions and normal controls (Fu et al., 2020). From 44,409 clinical images collected retrospectively, 5775 were randomly selected for training, 401 for internal validation and 402 for external validation, after excluding images based on quality measures. A separate clinical validation set of 666 images was also formed to compare results with performance of human readers. In this clinical validation set, non-OCSCC malignancies, oral epithelial dysplasia, and benign lesions were included (for a total of 77 additional images) and classified as ‘oral cancer’ in addition to OCSCC lesions. The authors first trained Single Shot MultiBox Detector (SSD) network to localize suspicious OCSCC lesions from entire images. The trained detector network was then used as a pre-processing step to crop the lesion area. The candidate patches (patches with the highest scores) were used to train the second network DenseNet-121 to classify target regions as oral cancer or normal mucosa (negative

control). With this two-stage network, the authors achieved 84.1% accuracy on the external validation test. The performance of the algorithm was further evaluated on the clinical validation set and compared with that of human readers. The model achieved 92.3% accuracy on the clinical validation set, comparable with the performance of a panel of 7 medical specialists who achieved an average accuracy of 92.4%. The other panels composed of medical and non-medical students achieved an accuracy of 87.0% and 77.2%, respectively. While the proposed model was shown to perform on a par with human experts in identifying OCSCC lesions, whether the model can differentiate between OCSCC and non-OCSCC oral diseases as well as between non-OCSCC oral diseases and normal oral mucosa remain unanswered. The detection results of the SSD network and whether it was able to recall all lesions are also not provided in the study.

Both Welikala et al. and Fu et al. are working towards developing mobile applications to deploy their models in real-world. The work of Welikala et al. is part of the MeMoSA® (Mobile Mouth Screening Anywhere) project, ran by Kingston University and University of Malaya, which aims to facilitate remote assessment and interpretation of oral lesions through a mobile app (Ching, 2018). The feasibility studies are performed in Malaysia. Fu et al. developed an app called CACalculator which localizes suspicious OCSCC lesion and gives a probability score for lesion being OCSCC or not (Fu et al., 2020, sec. Appendix). According to the authors, the app is being tested at the outpatient clinic of Hospital of Stomatology, Wuhan University.

Chapter 3:
METHODS

3.1 Dataset

The study was conducted in collaboration with the Institute of Oncology at Istanbul University and approved by the Ethics Committee of Istanbul University to be conducted from December 2019. Photographic images of oral cavity were collected from consented patients at the Istanbul University Hospital, which formed the initial source of our dataset. The rest of the dataset includes images collected from publicly available sources using web search engines due to the lack of publicly available dataset for oral cancer as yet. During image collection from public sources, we confirmed no duplicate images were included. Our final dataset consists of 652 images, each image containing at least one oral lesion. The anatomical location of the lesions varies greatly, including buccal mucosa, alveolar mucosa, tongue, mouth floor, palate, gingiva, lip etc. The dataset comprises of a diverse set of lesions coming from a wide range of oral diseases. The lesions are classified as 'benign', 'precancer', or 'cancer' based on the disease involved and its probability of progressing into oral cancer, as shown in Table 3. Example images from our dataset are displayed in Figure 26. Besides the heterogeneity of oral diseases included in the dataset, the images exhibit considerable variability in quality (i.e., lighting, zoom, angle, sharpness) and resolution. The median width and height of the images are 546 and 397 pixels with the width ranging from 156 to 5184 pixels and the height from 137 to 3456 pixels. The size of the lesions and the ration of lesion size to entire image size also varies significantly, as illustrated in Figure 27.

Table 3: Lesion classes and corresponding oral diseases.

Lesion Class	**Disease**	
Benign	Geographic tongue	Nicotine stomatitis
	Fissured tongue	Leukoedema
	Yellow tongue	Fibroma
	Median rhomboid glossitis	Fordyce granules
	Hairy leukoplakia	Pemphigoid

	Candidal leukoplakia	Gingivitis
	Pseudomembranous candidiasis	Periodontitis
	Keratosis (reactive / traumatic)	Pericoronitis
	Linea alba	Abscess
	Lichen planus	Erythema multiforme
	Ulcers (aphthous ulcer, traumatic ulcer, viral ulcers, TUGSE, etc.)	Pemphigus vulgaris
Precancer	Leukoplakia	Tobacco pouch keratosis
	Erythroplakia	Betel chewing keratosis
	Erythroleukoplakia	Submucous fibrosis
Cancer	Squamous cell carcinoma	

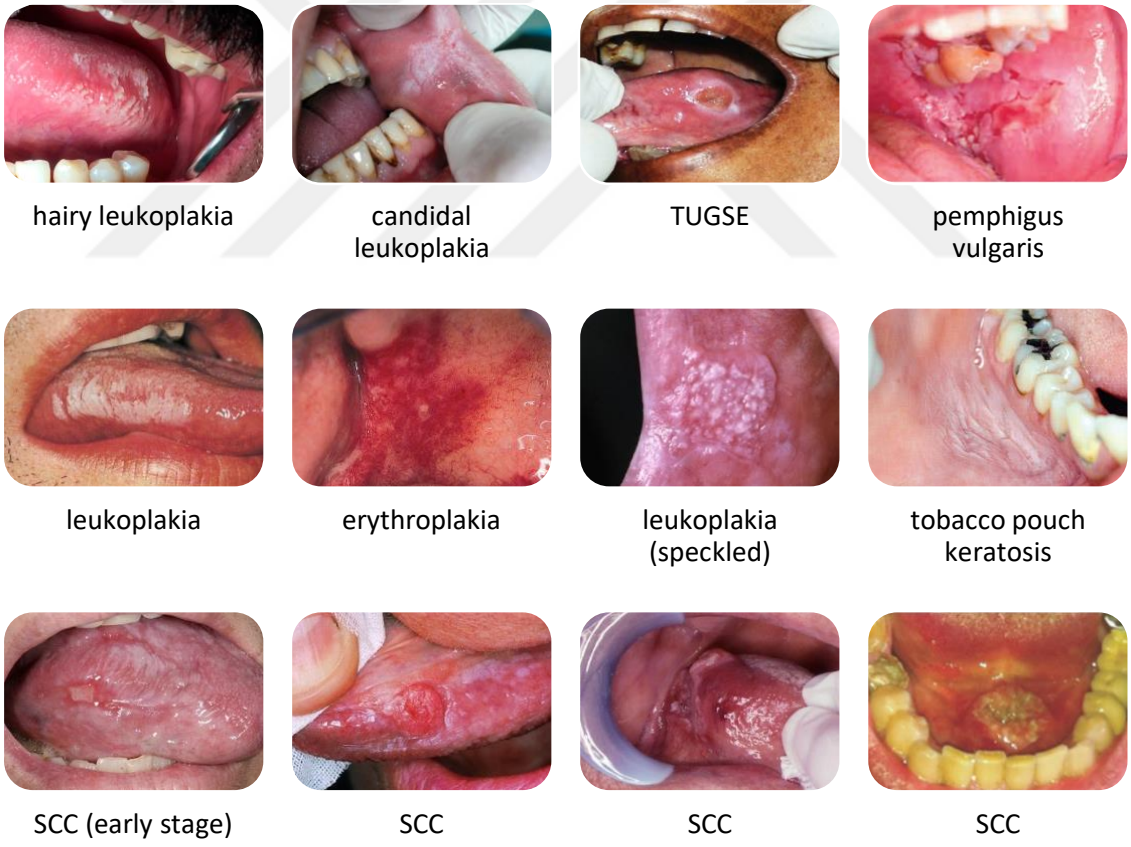


Figure 26: Example images from our dataset. Benign, precancerous, and cancerous lesions are shown in the first, second, and third rows respectively.

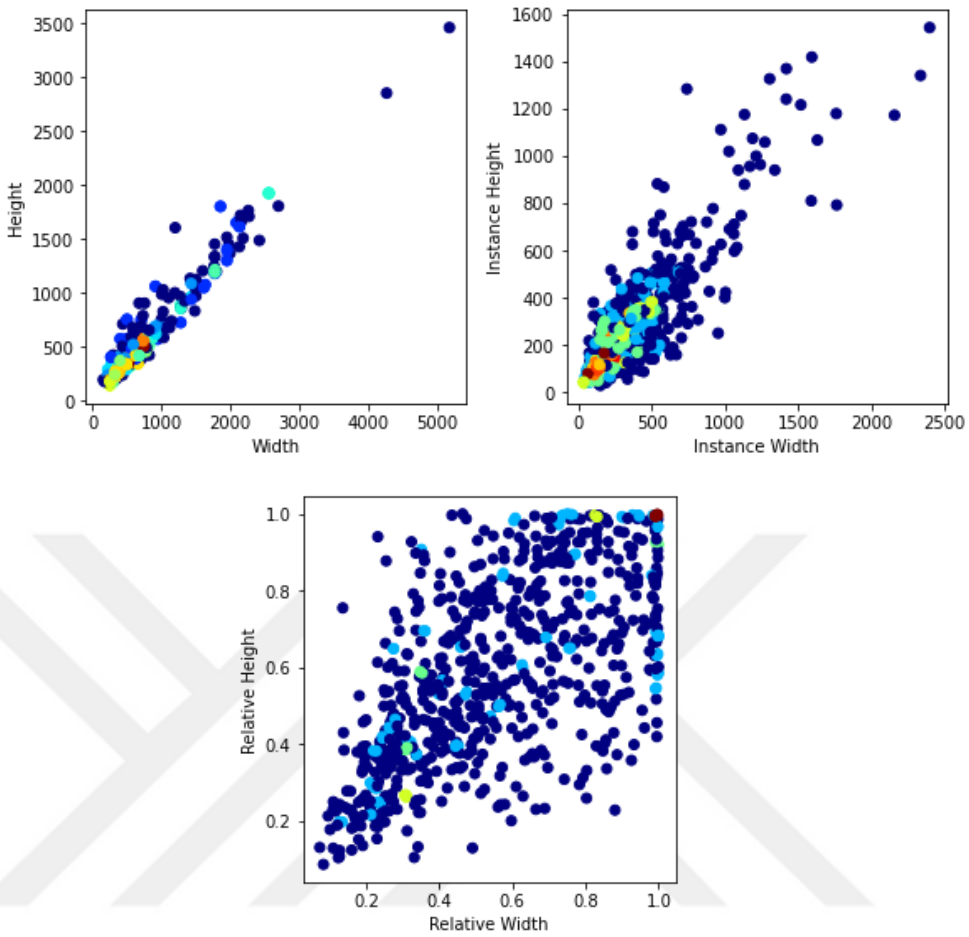


Figure 27: The dataset exhibits considerable variability in terms of image resolution. Absolute height and width of images (top left) and lesion instances (top right); relative width and height of lesion instances with respect to the width and height of corresponding image in the dataset (bottom) are shown.

The images were annotated under the supervision of an expert oral pathologist Assoc. Prof. Merva Soluk Tekkeşin using VGG Image Annotator (VIA) tool (A. Dutta et al., 2016; Abhishek Dutta & Zisserman, 2019). Bounding polygons were provided around regions of lesions along with their corresponding class as region attribute as shown in Figure 28. Some lesions did not have well-defined boundaries and/or had poor image quality, which presented a challenge during annotation and may have affected the accuracy of the models. The resulting annotations were saved as a JSON file.

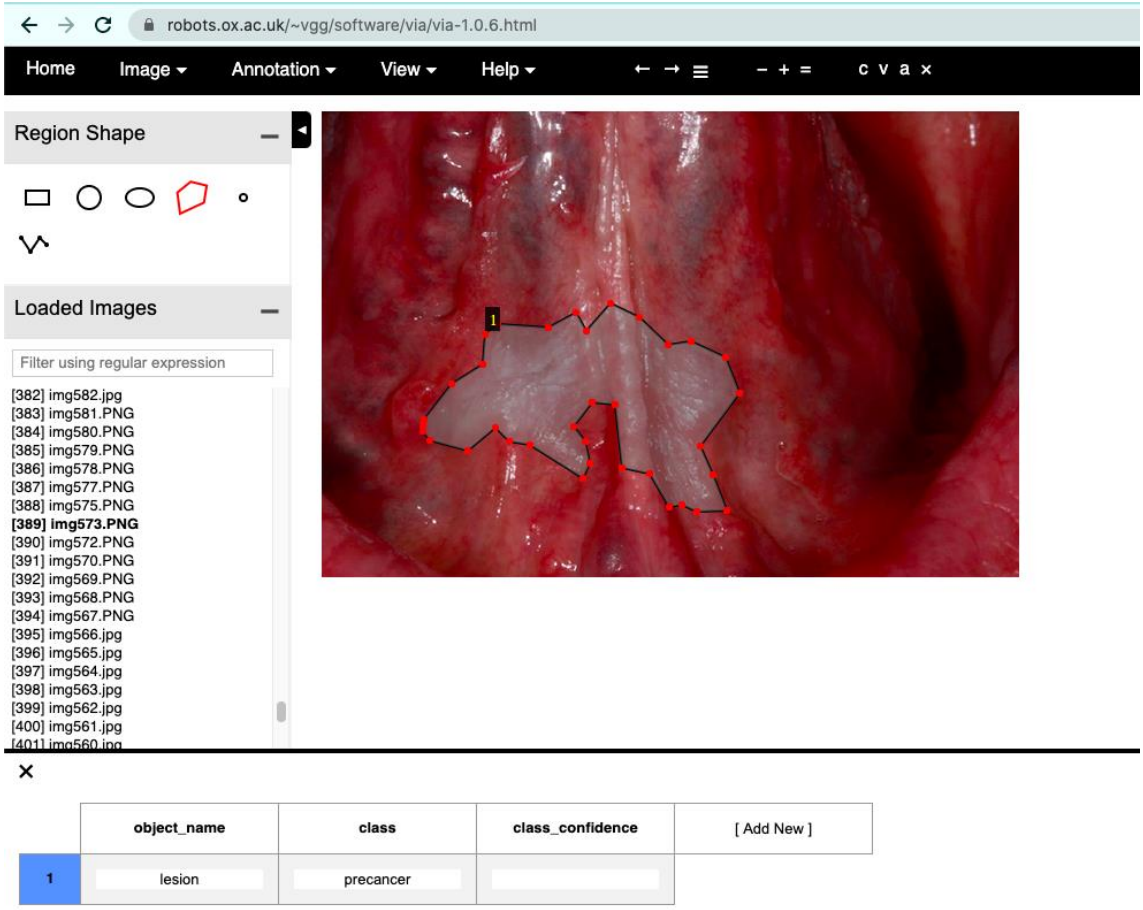


Figure 28: Screenshot from web-based VIA tool showing how annotations were completed. The images were annotated under the supervision of an expert oral pathologist. Bounding polygons were provided around regions of lesions along with their corresponding class as region attribute.

A total of 652 images were used for detection and segmentation experiments, which were split into approximately 80% training, %10 validation, and 10% test sets in a stratified fashion based on lesion class distribution (i.e., each set has the same proportion of class labels as in the original dataset). This resulted in 527 training, 61 validation, and 64 test images as shown in Table 4. As the oral lesions are often obscured by various structures in the mouth and complex background, we cropped the lesion areas as a rectangle (set1) or polygon (set2) and used these target regions for the classification experiments. This resulted in 286 benign, 257 precancerous, and 163 cancerous lesions for a total of 706 lesion regions. Since a good number of images contain more than one lesion, some of which can be of different class, we found that it is more practicable to classify close-up lesion areas separately, instead of assigning a global class for an entire image with multiple lesions. This approach also helped to boost the size of our dataset for

classification. A few of the images with multiple lesions had produced lesions with overlapping areas after rectangle cropping, in that case the secondary lesions were either removed from the dataset or taken together with the primary lesion if they belong to same class. Following this pre-processing and further removal of lesions with poor quality, a classification dataset of 684 images was produced and split into 552 training, 63 validation, and 69 test images. The class distribution of each set is described in Table *5*.

Table 4: Number of instances and images for detection and segmentation experiments according to lesion class and dataset type.

	Benign	**Precancer**	**Cancer**	**Total instances**	**Total images**
Training	230	210	131	571	527
Validation	27	22	15	64	61
Test	29	25	17	71	64

Table 5: Number of images for classification experiments according to lesion class and dataset type.

	Benign	**Precancer**	**Cancer**	**Total images**
Training	219	203	130	552
Validation	26	22	15	63
Test	29	23	17	69

3.2 Segmentation Experiments

Segmentation is an important step in automated diagnostic systems as it serves to delineate structural features such as boundaries of a lesion and provide a pixel-wise segmentation of anatomical structures. As described in previous chapter, semantic segmentation does not differentiate between different instances of the same object class. Lesion *instances* belonging to the same class are not considered separate instances by definition – in fact, they are often connected through underlying tissue-level changes which may not be fully visible in a standard photographic image. In this set of experiments, we aimed to explore lesion vs background segmentation as well as segmentation of different classes of lesions.

3.2.1 U-Net

We explored the capability of U-Net in distinguishing oral lesions from background in photographic images as well as its capability to distinguish between different types of oral lesions. We deviate from the typical U-Net structure shown in Figure 23 by replacing encoder part (i.e., backbone) with a pretrained classification model. We evaluated U-Net architecture with the backbones in Table 6 using qubvel's segmentation library built on Pytorch framework (Yakubovskiy, 2020).

Table 6: Backbones implemented with U-Net, corresponding number of parameters (in million), and selected hyperparameters such as batch size and learning rate.

Encoder	Parameters	Batch size	Learning Rate	Epochs
EfficientNet-b3	10M	8	0.0002	40
Densenet-161	26M	4	0.0001	40
Inception-v4	41M	8	0.0003	40
EfficientNet-b7	63M	4	0.0001	40
ResNeXt-101_32x8d	86M	12	0.0001	40

Dataset was prepared for segmentation by generating masks for each image from polygon annotations, where each mask is a stack of Boolean arrays, one for each class. For all models, the images and their corresponding masks were resized to 512 x 512 pixels with zero-padding to retain original aspect ratio. Data augmentation was applied to the training set, including horizontal and vertical flipping, random rotation, shifting, zooming, adding noise, sharpening, embossing, blurring, and changing brightness, contrast, and saturation using albumentations library (Buslaev et al., 2020). Due to our small dataset, we initialized backbones with pretrained ImageNet weights. Images were normalized according to the data statistics of corresponding pretrained model. For training, dice loss was used with Adam optimizer in segmentation experiments. Sigmoid activation function was used for segmentation of lesion vs background, where softmax was used for segmentation of all three lesion classes and background. Hyperparameters were optimized on the validation set for each model and selected as shown in Table 6. Once hyperparameters were optimized, the models were trained on both training and validation sets and its final performance was evaluated on test set. To further boost the test accuracy, test-time augmentation (TTA) was applied by testing the model on

augmented versions of the image (horizontal and vertical flips and rotations of 0, 90, 180, 270 angles) and taking the average of the results. All experiments were run on Tesla V100 GPU with 32GB memory.

3.2.2 Evaluation metrics

The performance of the models was measured with dice coefficient score (F_1-measure) and IoU (Jaccard) score which are frequently used metrics for assessing segmentation accuracy. Both metrics compare the intersection between the predicted mask and the ground-truth mask for each class with minor differences, as defined in below formulas (Taha & Hanbury, 2015). The scores are calculated per class and then averaged to produce the final score. The threshold for output binarization is set at 0.5.

$$Dice\ Score = \frac{2\ TP}{2\ TP + FP + FN} \qquad \textit{Eq. 3-1}$$

$$Jaccard\ Score = \frac{TP}{TP + FP + FN} \qquad \textit{Eq. 3-2}$$

3.3 Detection Experiments

Oral lesions are often masked or obstructed in a complex background of oral cavity and the photographic images exhibit extensive variability in terms of relative size and location of lesions; therefore, the localization of oral lesions is critical for providing a reliable classification result. We implemented two algorithms for this task, YOLOv5, which is one of the state-of-the-art algorithms for object detection, and Mask R-CNN, which is a powerful algorithm that provides both object segmentation and detection at the same time, as were discussed in 2.3.2.

3.3.1 YOLOv5

YOLOv5 is the Pytorch implementation of YOLOv4 (originally written in Darknet), released by Ultralytics for easier implementation and deployment to mobile applications, faster training and inference (Jocher, Stoken, Borovec, ChristopherSTAN, et al., 2020). Despite the version being called *v5*, YOLOv5 uses a CSP backbone with SPP block and PANet for network head same as YOLOv4 architecture. As “Bag of Freebies”, it utilizes

mosaic data augmentation and genetic algorithms for hyperparameter evolution. One of the other features is the auto-learning of bounding box anchors based on training data using k-means. There are four versions of the model based on the width and depth of the network, performance of each on COCO dataset is shown in Table 7.

Table 7: Performance and complexity of YOLOv5 models (release v2.0) on COCO val2017 with image size 640 pixel (AP: average precision, FPS: frame per second, params: number of parameters in million, FLOPS: floating point operations per second in billion). $Speed_{GPU}$ measures per image inference speed using one Tesla V100 GPU and includes image pre-processing, inferencing, post-processing and NMS (Jocher, Stoken, Borovec, ChristopherSTAN, et al., 2020).

Model	AP	AP50	$Speed_{GPU}$	FPS_{GPU}	Params	FLOPS
YOLOv5s	36.1	55.3	**2.2 ms**	**476**	7.5M	13.2B
YOLOv5m	43.5	62.5	3.2 ms	333	21.8M	39.4B
YOLOv5l	47.0	65.6	4.1 ms	256	47.8M	88.1B
YOLOv5x	49.0	67.4	6.4 ms	164	89.0M	166.4B
YOLOv5x + TTA	**50.4**	**68.5**	23.4 ms	43	89.0M	354.3B

We implemented release v2.0 of the repository with torch 1.6 and CUDA 10.1 (Jocher, Stoken, Borovec, ChristopherSTAN, et al., 2020). All training and optimization experiments were run on Tesla V100 GPU with 32GB memory. Test results were obtained on Tesla P100 GPU with 16GB. Leaky ReLU activation was used for convolution modules in our experiments instead of Mish. Hardswish activation was later implemented in release v3.0 but the AP improvement was minor on larger models while costing 10% in inference speed according to the results on COCO dataset (Jocher, Stoken, Borovec, NanoCode012, et al., 2020). When evaluated on our dataset, Hardswish was not found to improve AP in comparison to Leaky ReLU.

First, we converted our annotations from VIA format to YOLO format as required for the framework. All images were resized to 512 pixels along the longest image dimension and padded along shorter dimension to a minimum multiple of 32 for rectangular inference. With regards to augmentation, we used mosaic augmentation (example shown in Figure 29) as well as vertical and horizontal flipping, scaling, shifting, and augmenting colour space (i.e., hue, saturation, and contrast) of image. To prevent overfitting, all models were initialized with pretrained COCO weights. Hyperparameters were initially optimized by using a genetic algorithm for 100 generations as a coarse

optimization step, including the hyperparameters related to augmentation, then finer optimization of key hyperparameters was carried out. Nesterov stochastic gradient descent (SGD) optimizer with a learning rate of 0.001, momentum of 0.98, weight decay of 0.0005, and batch size of 8 were selected based on performance on validation set. Class weights were adjusted based on the labels. Models were run for 80 epochs including an initial warm up for 3 epochs. Confidence threshold and IoU NMS threshold which are used at test time to produce final outputs were selected as 0.55 and 0.3 for one-class lesion detection and 0.3 and 0.4 for three-class lesion detection, respectively. After hyperparameter optimization, the models were trained on both training and validation sets to utilize as much data as possible and final performance was evaluated on test set. Final results are reported with and without TTA which applies horizontal flipping and processing at 3 different resolutions (original resolution and 2 reduced multiples). Ensembling of multiple models such as YOLOv5s and YOLOv5m was also performed to further boost performance.

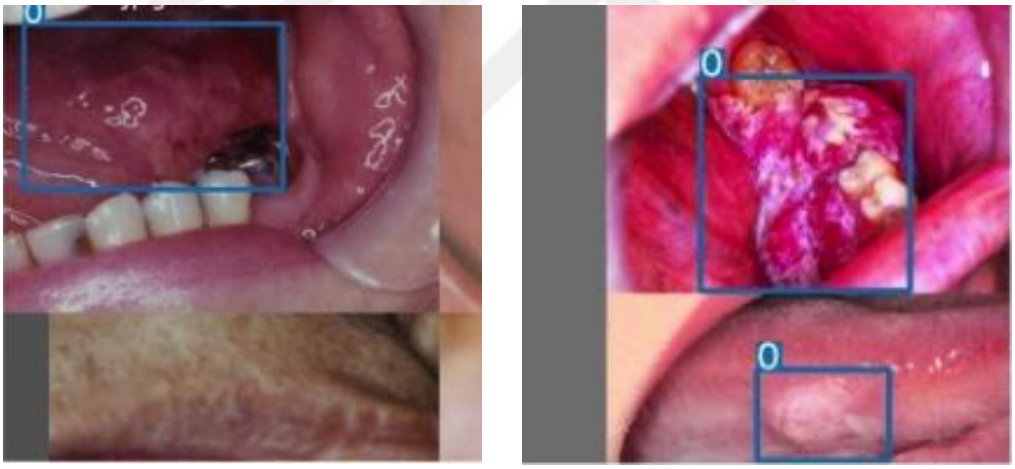


Figure 29: Example of special data augmentation technique called mosaic augmentation used in YOLOv5 to improve detection performance. Four images are combined at a maximum, each in different portions.

3.3.2 Mask R-CNN

The Pytorch implementation of Mask R-CNN provided by Facebook's Detectron2 library was used for the instance segmentation experiments (Wu et al., 2019). As a backbone, ResNet-50 FPN, ResNet-101 FPN, and ResNeXt-101 32x8d FPN were evaluated which were all pretrained on ImageNet. The experiments were run on release

v0.3 of the repository with torch 1.7 and CUDA 10.1 dependencies using Nvidia Tesla V100 GPU with 32GB memory (except for inferencing which were run on Tesla T4 GPU with 16 GB).

Table 8: Performance of Mask R-CNN with different backbones on COCO val2017. All experiments were run at 3x schedule (270,000 iterations or ~37 COCO epochs) with batch size of 16 images. 8 NVIDIA V100 GPUs & NVLink were used. Inference time is reported per image. (Data taken from (Wu et al., 2019).

Backbone	box AP	mask AP	Train memory	Inference time
ResNet-50 FPN	41.0	37.2	3.4GB	43 ms
ResNet-101 FPN	42.9	38.6	4.6GB	56 ms
ResNeXt-101 FPN	44.3	39.5	7.2GB	103 ms

All images were resized to 400 pixels along shorter image dimension with a limit of 512 on the longer dimension. With regards to data augmentation, vertical and horizontal flipping, random cropping, random rotation at multiples of 90 degrees, and augmenting colour space (i.e., brightness, saturation, contrast, and lighting) of image were applied to the training data. Hyperparameters were optimized based on performance on validation set. Adjustments to the anchor scales and aspect ratios were made according to the distribution of our data as shown in Figure 27. As the relative area of lesion instances in our dataset covers a wide range, from a minimum of approximately 38 pixels (i.e., sqrt of area) up to 512 pixels when resized, anchor scales were chosen as [[48, 64, 128, 256, 512]]. Similarly, the aspect ratios (i.e., height to width) of [[0.25, 0.5, 1.0, 2.0, 5.5]] were used in our implementation to account for the variation in our dataset, instead of the original aspect ratios of [[0.5, 1.0, 2.0]]. For the head of Mask R-CNN, a reduced ROI batch size of 256 was used which provided a more balanced foreground and background samples at a ratio of 1:3. SGD optimizer with a learning rate of 0.0002, momentum of 0.9, and weight decay of 0.0005 were selected based on optimization experiments. All models were run for 10000 iterations (~76 epochs) at a batch size of 4 including initial warm up for ~3 epochs. Learning rate was decayed by 0.1 at iteration 6000 for all models, with an additional earlier decay at iteration 3000 for ResNeXt-101 model. To prevent overfitting, all models were initialized with pretrained COCO weights and best model checkpointing was utilized. The freezing of ResNet stages (pretrained) was found to be providing worse results on our dataset, therefore the entire backbones were finetuned. The IoU NMS threshold and confidence threshold which are used at test time to produce

final outputs were selected as 0.5 to eliminate highly overlapping and low score predictions. After hyperparameter optimization, the models were trained on both training and validation sets and final performance was evaluated on test set. Results are reported with and without TTA which applies horizontal flipping and processing at 3 different resolutions (400, 500, 600 pixels) and takes average of the augmented results.

3.3.3 Evaluation Metrics

Average precision (AP) is a widely used metric for measuring accuracy in object detection. In order to describe how AP is calculated, precision and recall should be explained first. Precision measures the percentage of correct predictions among all predictions. Recall measures the percentage of correct predictions among all ground-truths and is called sensitivity. Formulas for precision and recall are given below:

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{Number\ of\ predictions} \qquad \textit{Eq. 3-3}$$

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{Number\ of\ ground\ truths} \qquad \textit{Eq. 3-4}$$

In the concept of object detection, a prediction is considered true positive (TP) if the IoU between the ground truth and the prediction is greater than a certain IoU threshold (usually >0.5) and its predicted class is correct, otherwise it is considered false positive (FP) (Hui, 2018). If an object is not detected at all, it is considered a false negative (FN). When precision and recall are calculated and plotted based on above formulas for a batch of images, the area under the resulting precision-recall curve gives the AP. We can observe that precision decreases as recall increases and vice versa, therefore interpolation of average precision from a range of recall values is practical. If AP is calculated at a fixed IoU such as 0.5 and 0.75, these are abbreviated as AP50 and AP75 respectively. Since the value of threshold may introduce some bias to the evaluation, AP is also calculated over a range of IoU thresholds from 0.5 to 0.95 with a step size of 0.05 and the average of these 10 values is taken to produce the final AP of that class in COCO evaluation (COCO Consortium, 2016). If there are multiple object classes, then the AP is averaged over all classes – this can be also called as mean AP (mAP). Average recall (AR) is calculated in a similar fashion.

$$AP[class, IoU] = \int precision_{interp}(r)\, dr \qquad \textit{Eq. 3-5}$$

$$AP[class] = \frac{1}{Number\ of\ IoU\ thresholds} \sum_{IoU} AP[class, IoU] \qquad \textit{Eq. 3-6}$$

$$AP\ = \frac{1}{Number\ of\ classes} \sum_{class} AP[class] \qquad \textit{Eq. 3-7}$$

3.4 Classification Experiments

Since many photographic images of oral lesions also contain other structures such as teeth, dental tools, and gloves and some images contain multiple lesions of different classes, we isolated lesion regions as explained in section 3.1 and performed classification experiments on these cropped images.

3.4.1 Feature Extraction

In order to obtain baseline results for classification, we extracted hand-crafted features using image processing and evaluated various ML algorithms on the classification dataset. After some experimentation, colour histogram, Hu moments, LBP, and GLCM features were selected. OpenCV library was used for extracting histogram and Hu moments features (Bradski, 2000); Scikit-image library was used for extracting GLCM and LBP features (Van der Walt et al., 2014). The images were resized to 256 x 256 pixels and converted to grey scale (except for colour histogram) prior to feature extraction. Extracted features were concatenated and min-max scaled, then fed into ML models such as Random Forest, SVM, and MLP. Results were compared with a dummy classifier that generates predictions based on class distribution in the training set (also called *stratified* strategy). Scikit-learn library was used for implementation of ML models and producing evaluation metrics (Pedregosa et al., 2011).

3.4.2 CNN Algorithms

Different types of CNN architectures were evaluated for CNN-based lesion classification. Selected architectures and their corresponding performance on ImageNet dataset are provided in Table 9. For ResNet-152 and DenseNet-161, official Pytorch versions were implemented (Paszke et al., 2019), both of which take an input size of 224 x 224 pixels. Furthermore, we used Lukemelas' version for EfficientNet (Melas-Kyriazi, 2020) and Cadene's version for Inception-v4 implementations (Cadene, 2020). The last layer of each model was modified accordingly to produce class scores for 3 classes – benign, precancer, cancer. In addition to the models described above, we built an ensemble of DenseNet161 and ResNet-152, which uses the average of the outputs produced by two models for prediction. We used ImageNet pretrained weights to initialize our models due to small size of our dataset and trained all layers. For DenseNet161 and ResNet-152, we also evaluated training in three stages – heads only, some layers (denseblock3 onwards in DenseNet161 and block 3.14 onwards ResNet-152), all layers – with learning rate reduction at every stage.

Table 9: Selected CNN architectures and their performance on ImageNet-1k dataset showing top-1 accuracy and number of parameters for each architecture (Cadene, 2020; Melas-Kyriazi, 2020; Paszke et al., 2019).

Architecture	Image Size	Version	Acc@1	Params
EfficientNet-b4	380 x 380	Lukemelas	82.6	19M
Inception-v4	299 x 299	Cadene	80.062	42.7M
ResNet-152	224 x 224	Pytorch	78.428	60M
DenseNet-161	224 x 224	Pytorch	77.560	28.7M

Images were resized to respective input sizes based on the pretrained model as noted in Table 9. Zero-padding was applied during resize where needed to retain aspect ratio. Data augmentation such as horizontal and vertical flipping, rotation at multiples of 90 degrees, perspective distortion, shifting, shearing, blurring, adding noise, and augmenting colour space (i.e., brightness, saturation, contrast, and jitter) were applied at random to training set. Images were normalized according to mean and standard deviation of the pretraining settings. Hyperparameters including data augmentation hyperparameters were optimized based on performance on validation set and selected as shown in Table 10. Momentum was set as 0.9 for all experiments. Learning rate was reduced by a factor of

0.5 when validation loss plateaued for at least 3 epochs and the best model was saved during training in order to prevent overfitting. After hyperparameter optimization, the models were trained on both training and validation sets and the final performance was evaluated on test set. T-distributed stochastic neighbour embedding (t-SNE) technique was employed to visualize learned feature mappings in two-dimensional space using scikit-learn library (Pedregosa et al., 2011). All experiments were run with torch 1.7 and torchvision 0.8.1 with CUDA 10.1 using Nvidia Tesla T4 GPU with 16 GB memory.

Table 10: Selected hyperparameters for classification experiments.

	Optimizer	**Batch size**	**Learning rate**	**Epochs**	**Weight decay**
EfficientNet-b4	Rmsprop	16	5e-6	35	1e-2
Inception-v4	SGD	16	1e-3	35	1e-3
ResNet-152	SGD	16	1e-3	35	5e-4
DenseNet-161	SGD	16	1e-3	35	5e-4
Ensemble	SGD	16	1e-3	35	5e-4
ResNet-152 staged	SGD	16	1e-3, 5e-4, 5e-5	3-12-20	5e-4
DenseNet-161 staged	SGD	16	1e-3, 5e-4, 5e-5	3-12-20	1e-3

3.4.3 Evaluation Metrics

Accuracy, precision, recall, and F_1-score were calculated from generated confusion matrices and were used to evaluate classification experiments. Precision and recall were explained in 3.3.3. F_1-score is harmonic mean of precision and recall, therefore, is a more practical measure than precision and recall alone. We report weighted macro-average F_1-score, which computes the metric for each class and takes the weighted average, as it accounts for class imbalance. Confusion matrices and related metrics were calculated using scikit-learn library (Pedregosa et al., 2011).

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \qquad \textit{Eq. 3-8}$$

3.5 App Deployment

We built a web application to deploy our best performing models, using the Python-based app framework Streamlit v0.74.1 (Teixeira, 2020). The app takes an image as an input and generates bounding box detections along with a classification score for each detected lesion using a second-stage classifier network as illustrated in Figure 30. For detection part, we employed YOLOv5 instead of Mask R-CNN due to its better inference speed. We added a slider sidebar to play with the confidence threshold and NMS IoU threshold parameters of YOLOv5 for demo purposes. The application can also be accessed from a mobile phone and allows for direct upload of a photograph using phone's camera.

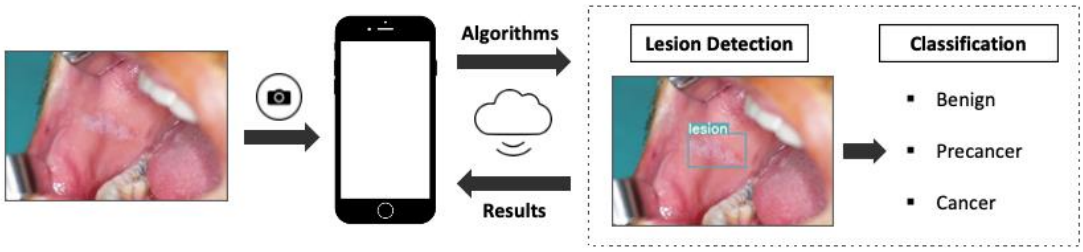


Figure 30: Schematic of the proposed two-stage framework deployed on a smart phone. A detection network detects oral lesions in the first stage and a classifier network classifies the detected region as benign, precancer, or cancer in the second stage.

Chapter 4:
RESULTS & DISCUSSION

4.1 Segmentation Results

U-Net architecture was evaluated with different backbones for lesion segmentation task. The results for background vs lesion segmentation task are shown in Figure 31 and Table 11. U-Net architecture performed generally well on this task with any of the evaluated backbones achieving over 0.9 dice score on the test set as the primary segmentation evaluation metric. The best performing model U-Net with EfficientNet-b7 backbone and TTA achieves a Jaccard score of 0.873 and a dice score of 0.929. The predicted mask outputs and their corresponding ground-truth masks are visualized in Figure 32, which shows that lesions of various types and difficulty level are segmented with good precision.

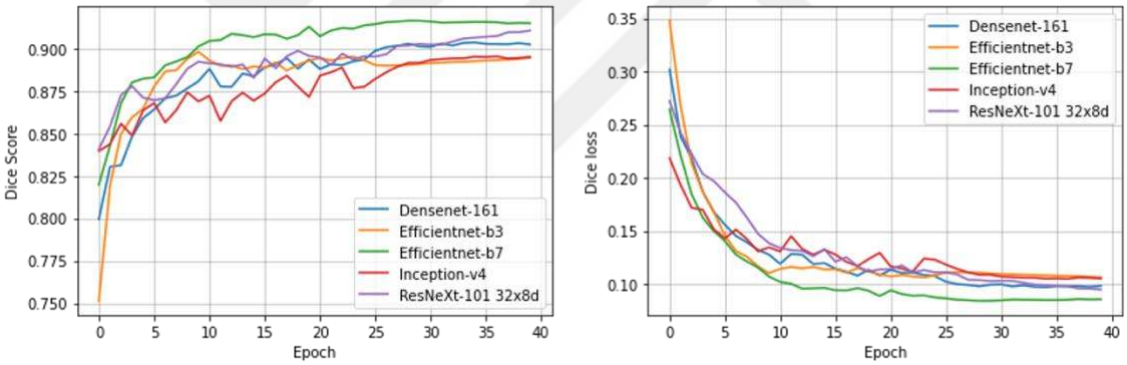


Figure 31: Plots showing dice loss and dice score on validation set during training of U-Net with various backbones after hyperparameter optimization. There is an exponential increase in dice score (and a decrease in dice loss) in the early phases of training, then it saturates as the models converge to an optimal solution.

Table 11: U-Net results on background vs lesion segmentation task with various backbones using best checkpoint, showing Jaccard (IoU) score and Dice (F_1) score for validation and test sets, respectively. Reported test results obtained after full training on training and validation sets.

Backbone	**Jaccard$_{val}$**	**Dice$_{val}$**	**Jaccard$_{test}$**	**Dice$_{test}$**	**Dice$_{test}$ w/ TTA**
EfficientNet-b3	0.831	0.902	0.866	0.925	0.927
Densenet-161	0.836	0.905	0.858	0.921	0.927
Inception-v4	0.821	0.897	0.850	0.915	0.922
EfficientNet-b7	0.855	0.918	**0.867**	**0.926**	**0.929**
ResNeXt-101_32x8d	0.845	0.912	0.863	0.923	0.928

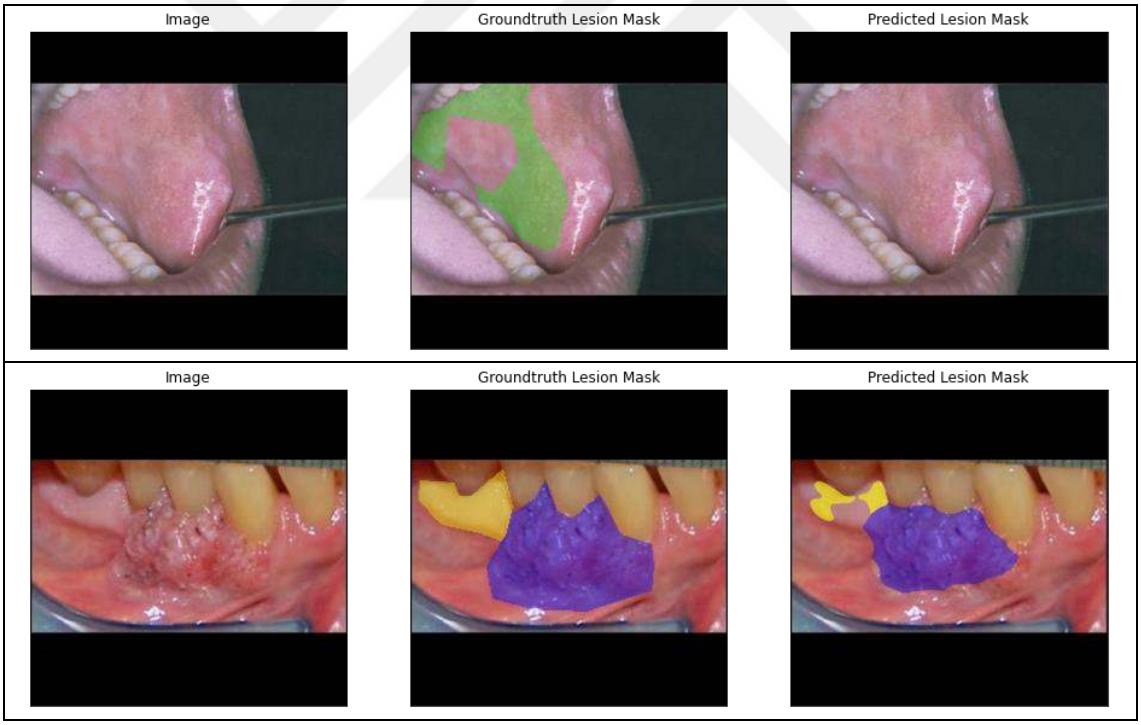
Figure 32: Ground-truth and predicted lesion masks with EfficientNet-b7 model on test set.

Furthermore, the same architectures were evaluated for class-wise segmentation – including background, benign, precancer, cancer classes – of oral lesions. As detailed in Table 12, the models performed poorly on this task compared to binary segmentation and the models were somewhat unstable. Final training of the models on both training and validation sets contributed to a modest improvement in the test set results. The best performing models, EfficientNet-b7 and ResNeXt-101 32x8d with TTA, both achieved Jaccard scores of 0.743 and dice scores of 0.843 on the test set. The predicted and ground-truth masks can be visualized in Figure 33. It is observed that some of the lesions were either missed or failed to be segmented with good accuracy, especially benign class being the most challenging. This is possibly due to subtle transition from normal to abnormal tissue in benign lesions, making their segmentation more difficult. There were also cases

of misclassification of the segmented region as shown in the bottom example in Figure 33.

Table 12: U-Net results on class-wise lesion segmentation task with various backbones, showing Jaccard (IoU) score and Dice (F_1) score for validation and test sets, respectively. Reported test results obtained after full training on training and validation sets.

Backbone	**Jaccard$_{val}$**	**Dice$_{val}$**	**Jaccard$_{test}$**	**Dice$_{test}$**	**Dice$_{test}$ w/ TTA**
EfficientNet-b3	0.650	0.765	0.677	0.797	0.808
Densenet-161	0.611	0.738	0.690	0.798	0.804
Inception-v4	0.658	0.772	0.662	0.781	0.771
EfficientNet-b7	0.678	0.787	0.714	0.823	0.843
ResNeXt-101_32x8d	0.677	0.786	0.739	0.841	0.843



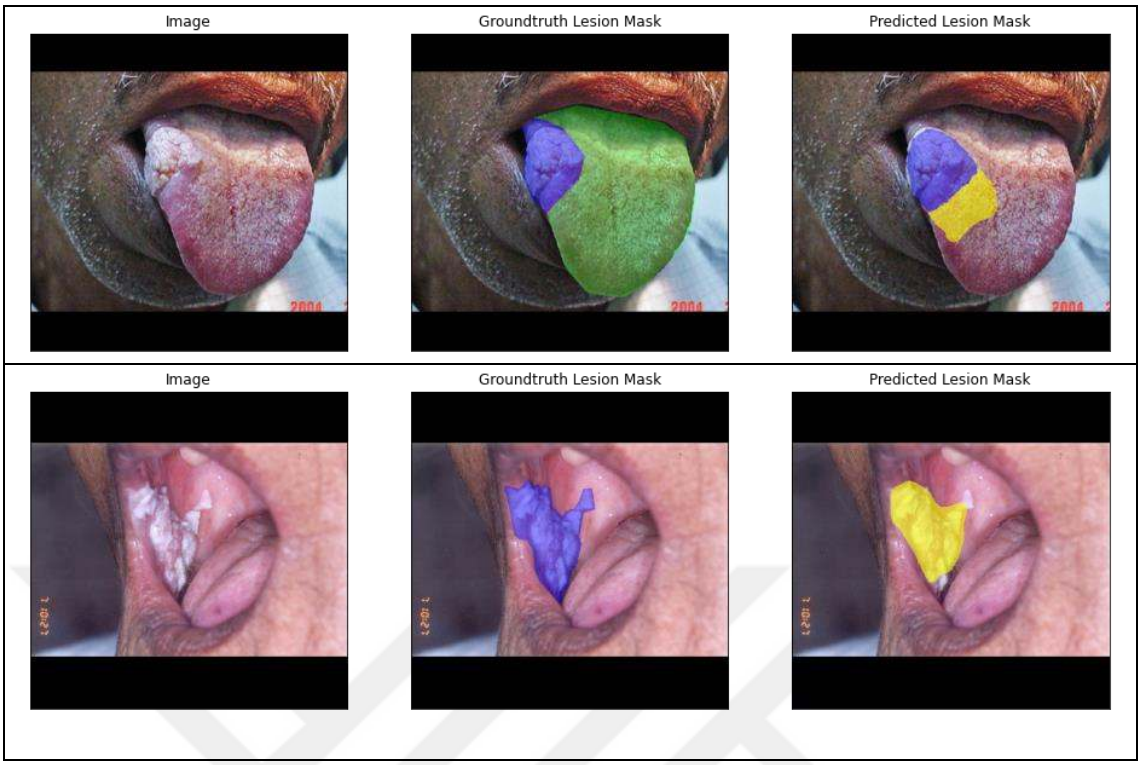


Figure 33: Ground-truth and predicted class-specific lesion masks with ResNeXt-101_32x8d model on test set. Mask colours: green for benign, yellow for precancer, blue for cancer.

To our knowledge, this is the first report of semantic segmentation of oral lesions using photographic images. Although lesion *instances* belonging to the same class are not considered separate instances by definition, the application of semantic segmentation may be limited in certain tasks as it does not distinguish between different instances of lesions. We compare our results with an instance segmentation study by Anantharaman et al. for detection and segmentation of mouth sores (cold sores vs canker sores) (Anantharaman et al., 2019). In this small-scale study with 30 training images, the authors reported an average dice score of 0.744 on the test set that is composed of 10 images. With dice scores of 0.929 on binary segmentation and 0.843 on multi-class segmentation, our work demonstrates feasibility of semantic segmentation for oral diseases using non-standardized photographic images.

4.2 Detection Results

4.2.1 YOLOv5

Four different versions of YOLOv5 were evaluated for one-class lesion detection. Figure 34 shows the plots for AP and AP50 values on validation set during training for

all of the four models (further plots available in **APPENDIX A**). Following hyperparameter optimization on validation set, all models were trained on training and validation set together in an attempt to further boost the performance as a way to compensate for our small dataset. In Table 13, the performance results of each model are reported on validation and test sets with image size of 512 pixels and selected confidence and IoU NMS thresholds.

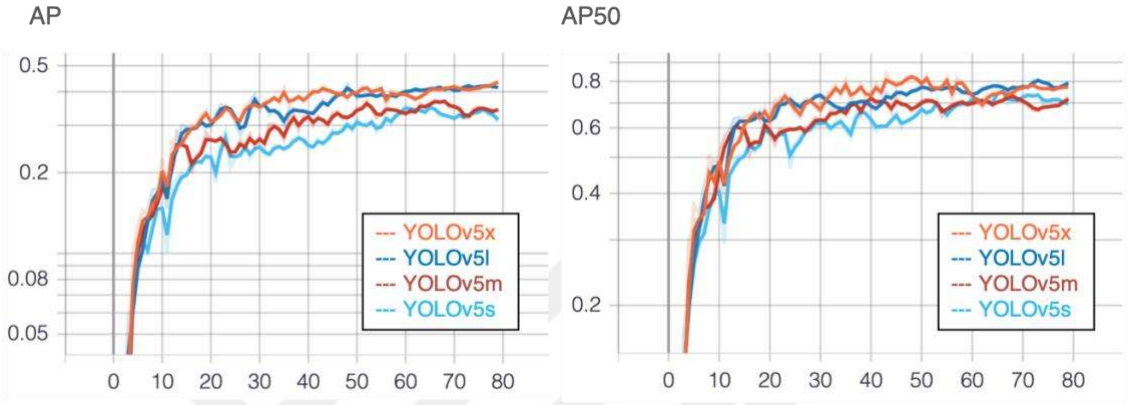


Figure 34: Plots of AP (left) and AP50 (right) on validation set during training for 80 epochs on one-class lesion detection task.

YOLOv5l performed the best among all versions with AP_{val} of 0.63 and AP_{test} of 0.644, followed by YOLOv5x with AP_{val} of 0.607 and AP_{test} of 0.613, YOLOv5m with AP_{val} of 0.582 and AP_{test} of 0.607, and YOLOv5s AP_{val} of 0.553 and AP_{test} of 0.579. In terms of test AP50, YOLOv5l achieves 0.951 on test set, followed by 0.920 with YOLOv5s, 0.902 with YOLOv5x, and 0.896 with YOLOv5m. When TTA is applied, we achieve as high as 0.953 on test AP50 with our YOLOv5l model although this comes at a cost of slight reduction in AP since TTA generally works to increase recall while reducing precision. As expected, bigger models achieved better performance on our dataset similar to COCO results (shown in Table 7), with the exception of YOLOv5x which achieved slightly lower AP compared to YOLOv5l and was more prone to overfitting. This could be due to the relatively small size of our dataset that is better suited to less complex models on this task. We further showed that ensemble of two smaller models such as YOLOv5s and YOLOv5m achieves almost the same AP as YOLOv5l but at a higher speed. Confidence threshold of 0.55 and IoU NMS threshold of 0.3 provided a reasonable trade-off between precision and recall for our use case, but these may be further adjusted to increase recall or precision as needed. Increasing IoU NMS threshold tends to improve recall and consequently AP50 but it comes at a cost of reduced precision.

Table 13: One-class YOLOv5 results for lesion detection with best model checkpoints. AP is shown for both validation and test sets. AP50 and $Speed_{GPU}$ are shown only for test set. $Speed_{GPU}$ measures per image inference speed in ms using one Tesla T4 GPU and includes image pre-processing, inferencing, post-processing and NMS. The AP values were computed after applying confidence threshold and IoU NMS threshold which were set as 0.55 and 0.3, respectively.

Model	**AP_{val}**	**AP_{test}**	**$AP50_{test}$**	**$Speed_{GPU}$(ms)**
YOLOv5s	0.553	0.579	0.920	4.4
YOLOv5m	0.582	0.607	0.896	6.9
YOLOv5l	**0.63**	**0.644**	0.951	10.6
YOLOv5l + TTA	-	0.622	**0.953**	21.2
YOLOv5x	0.607	0.613	0.902	18
YOLOv5x + TTA	-	0.630	0.940	35.3
YOLOv5s & 5m ensemble	0.585	0.637	0.923	9

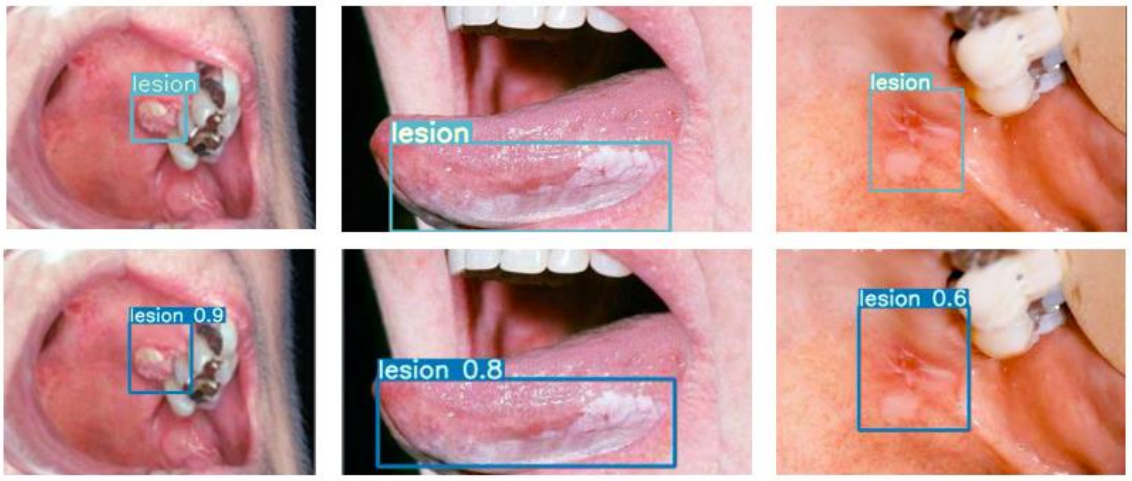


Figure 35: One-class lesion detection results with YOLOv5l on test set. The top row shows the ground-truth boxes, and the bottom row shows the model predictions.

We also evaluated YOLOv5 models on three-class object detection of oral lesions as a way of detecting and classifying oral lesions simultaneously. The results are shown in Figure 36 and Table 14.

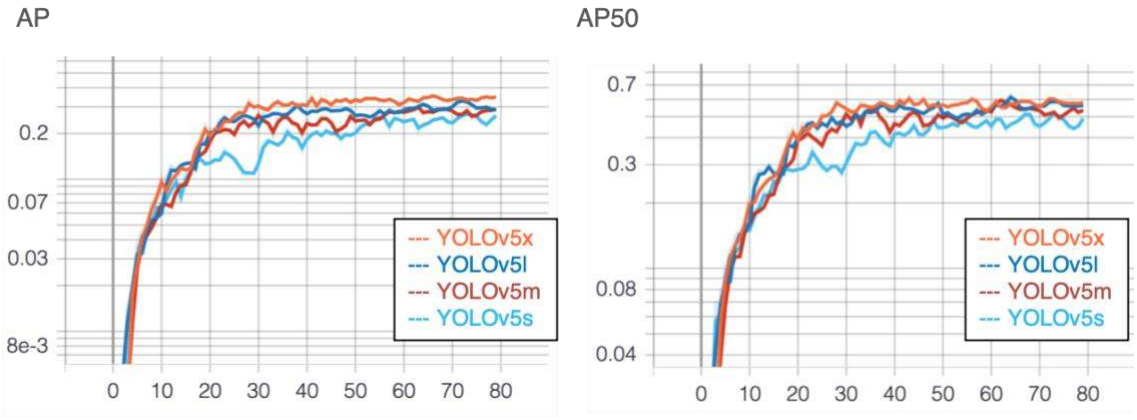


Figure 36: Plots of AP (left) and AP50 (right) on validation set during training for 80 epochs on three-class lesion detection task.

Table 14: Three-class YOLOv5 results for finer lesion detection with best model checkpoints. AP is shown for both validation and test sets. AP50 and $Speed_{GPU}$ are shown only for test set. $Speed_{GPU}$ measures per image inference speed in ms using one Tesla T4 GPU and includes image pre-processing, inferencing, post-processing and NMS. The AP values were computed after applying confidence threshold and IoU NMS threshold which were set as 0.3 and 0.4 respectively.

Model	**AP_{val}**	**AP_{test}**	**$AP50_{test}$**	**$Speed_{GPU}$(ms)**
YOLOv5s	0.463	0.419	0.697	5.2
YOLOv5m	0.467	0.464	0.731	7.4
YOLOv5l	0.473	0.534	0.827	10.7
YOLOv5l + TTA	-	0.530	0.805	21
YOLOv5x	**0.554**	**0.566**	**0.829**	17.4
YOLOv5x + TTA	-	0.536	0.810	35
YOLOv5s & 5m ensemble	0.516	0.478	0.742	10.4

As expected, YOLOv5x performed the best among all versions with mean AP_{val} of 0.554, AP_{test} of 0.566, and $AP50_{test}$ of 0.869 across all classes, followed by YOLOv5l with AP_{val} of 0.473, AP_{test} of 0.534, $AP50_{test}$ of 0.827, YOLOv5m with AP_{val} of 0.467, AP_{test} of 0.464, $AP50_{test}$ of 0.731, and YOLOv5s AP_{val} of 0.463, AP_{test} of 0.419, $AP50_{test}$ of 0.697. In comparison to one-class detection, the models achieved lower AP and much lower recall on both validation and test sets possibly due to the difficulty of learning two tasks (i.e., detection and classification) at the same time with a small and challenging dataset. As shown in Table 15, precancerous lesions have the lowest scores among all

classes, which is in line with our expectations since these lesions are commonly misclassified or go undetected also in clinical practice, making up the most challenging group for healthcare professionals. Although the results are encouraging, future studies with more data are needed to improve simultaneous detection and classification of oral lesions.

Table 15: Per-class precision, recall, AP, and AP50 results of YOLOv5x model on test set.

Class	Precision	Recall	AP	AP50
All	0.701	0.646	0.566	0.829
Benign	0.769	0.69	0.579	0.889
Precancer	0.6	0.6	0.498	0.741
Cancer	0.733	0.647	0.622	0.856

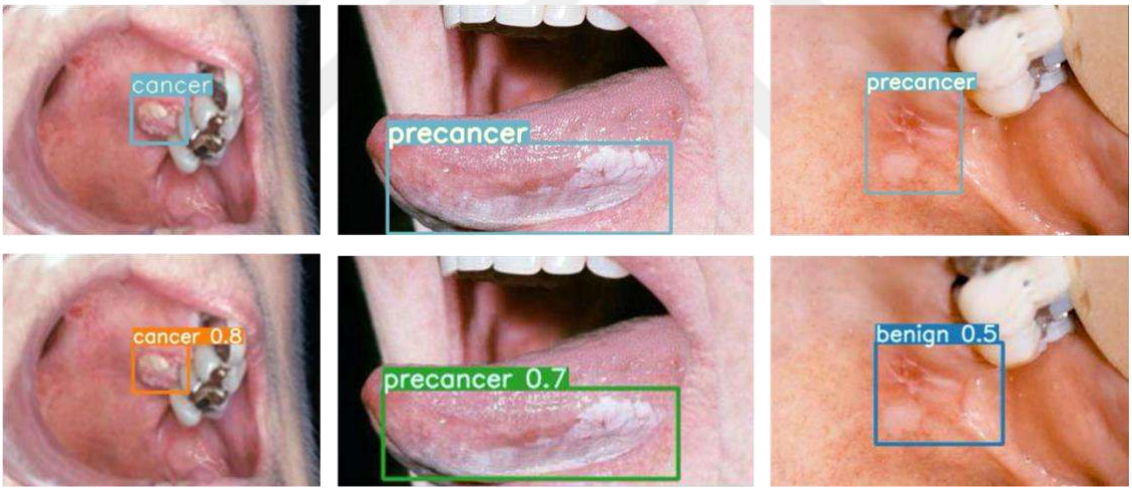


Figure 37: Three-class lesion detection results with YOLOv5x on test set. The ground truth class of the lesions are cancer, precancer, and precancer respectively as shown in the top row. The bottom row shows the model predictions. The lesion in the most right image was misclassified as benign.

To our knowledge, this is the first report of YOLO models for oral lesion detection. The results are very promising especially for one-class detection task despite the size of our dataset and the variability of images. With an inference speed of 10.6 ms per image on Tesla T4 GPU, YOLOv5l offers a great potential for deployment in a real-world application.

4.2.2 Mask R-CNN

Mask R-CNN with ResNet-50, ResNet-101, and ResNeXt-101 FPN backbones were trained with selected hyperparameters as shown in Figure 38 for lesion detection and segmentation task. The model with the biggest backbone ResNeXt-101 converged in a less number of iterations and was more prone to overfitting as the training continued, therefore, learning rate was decayed at both 3000 and 6000 iterations and the best model checkpointing was utilized to tackle overfitting issue.

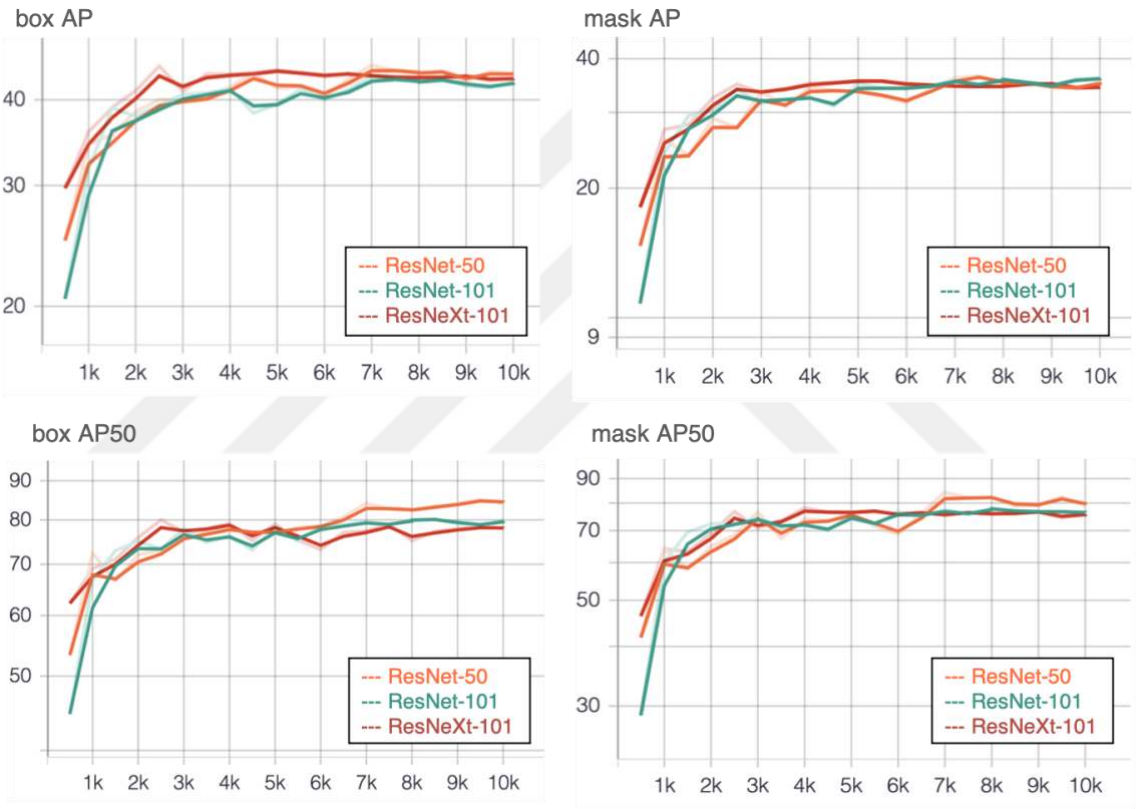


Figure 38: The bounding box and segmentation mask AP and AP50 results on validation set for all backbones. Metrics were computed every 500 iterations due to computational overhead.

The test results are provided in Table 16 for all backbones, both with and without TTA for completeness although applying TTA does not always yield better AP and when it does it comes at a cost of increased inference time by 8x. As summarized in Table 16, the models achieved similar results in general and the best overall results without TTA were obtained by ResNeXt-101 model with a mask AP score of 37.85, box AP score of 43.90, mask AP50 score of 79.74, and box AP50 score of 78.00. The performance of the

model based on lesion scales is given in Table 17, which shows that the model performs worse on small scale lesions although the difference is not of a major concern. The model predictions on some of the test images can be visualized in Figure 39. It can be observed that the model was generally able to detect and segment lesions with good precision even there was multiple lesions in an image (third image) or the lesion was small relative to the entire image (fifth image). Nevertheless, the model failed to detect one lesion completely as shown in Figure 39 (sixth image) given the confidence and NMS thresholds applied. While the results are promising, future studies should focus on data collection to further improve performance. More advanced instance segmentation architectures such as Cascade Mask R-CNN may also bring significant gains in terms of box AP (Cai & Vasconcelos, 2017). Furthermore, we attempted to train Mask R-CNN for three-class lesion detection and segmentation task, yet it was not possible to fit the models given the complexity of the task, therefore no results are provided for this secondary task.

Table 16: The Mask-RCNN results on validation and test sets with ResNet-50, ResNet-101, and ResNeXt-101 FPN backbones. The results are provided with and without TTA. $Speed_{GPU}$ measures per image inference speed in ms using one Tesla T4 GPU and includes image pre-processing, inferencing, post-processing and NMS. The AP values were computed after applying confidence threshold and IoU NMS threshold which were set at 0.5.

Backbone	**box AP_{val}**	**mask AP_{val}**	**box AP**	**box AP50**	**mask AP**	**mask AP50**	**$Speed_{GPU}$ (ms)**
ResNet-50 FPN	43.65	35.3	42.53	80.51	37.23	74.08	46
+ TTA	45.11	36.96	42.65	82.63	37.98	76.19	361
ResNet-101 FPN	42.47	36.03	41.85	81.86	37.70	74.41	56
+ TTA	43.85	35.94	40.54	83.64	37.52	72.96	442
ResNeXt-101 FPN	44.81	35.48	**43.90**	79.74	**37.85**	78.00	89
+ TTA	45.32	35.73	43.35	81.60	37.80	78.92	786

Table 17: Box and mask AP results for ResNeXt-101 model based on area ranges. The ranges were modified as [0**2, 160**2] for AP_{small}, [160**2, 288**2] for AP_{medium}, and [288**2, 10000**2] for AP_{large} according to the distribution of instances in our dataset.

	AP_{small}	**AP_{medium}**	**AP_{large}**	**AP_{all}**	**AR_{all}**
box AP	42.7	45.6	49.9	43.9	56.5
mask AP	38.4	37.9	44.8	37.8	50.0

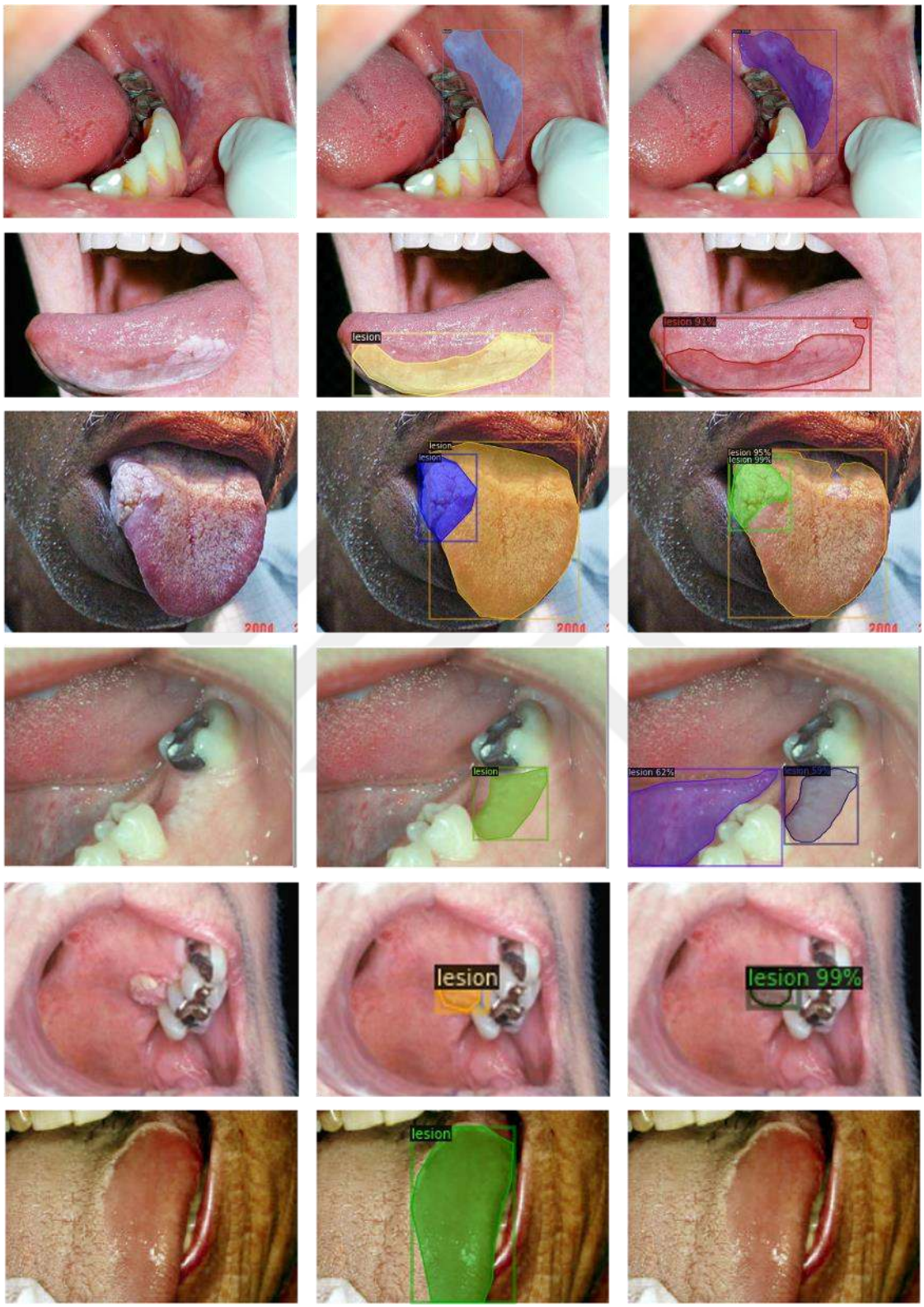


Figure 39: Mask R-CNN predictions on test set images. Original image without annotations (left), ground-truth bounding box and segmentation mask (middle), and predicted bounding box and mask (right) are displayed for each image.

In conclusion, we attempt to compare our results with previous work on oral lesion detection and segmentation. Anantharaman et al. implemented Mask R-CNN for detection and segmentation of mouth sores with a very small dataset (40 images), yet only dice score was reported as a pixel-wise segmentation metric instead of AP scores (Anantharaman et al., 2019). Another study by Welikala et al. used Faster R-CNN, which is same as Mask R-CNN but without the segmentation branch, for detection of oral lesions with different referral scores (Welikala et al., 2020). The authors reported an F_1-score of 41.35% for one-class lesion detection task, a macro-average F_1-score of 32.41% on two-class lesion detection task (referral vs non-referral), and a macro-average F_1 score of 24.50% on four-class detection task (non-referral, refer-other reasons, refer-low risk, refer-high risk) at an IoU threshold of 0.5. For comparison, we compute F_1 scores based on precision and recall results from our experiments. Our best Mask R-CNN implementation achieves F_1-scores of 77.22% and 75.95% on one-class lesion detection task for bounding box and segmentation mask respectively, and YOLOv5 implementation achieves an F_1-score of 85.52% on one-lesion detection task and a macro-average F_1-score of 67.24% on three-class lesion detection task at an IOU threshold of 0.5. However, it should be noted that the comparison may not be appropriate considering the variations in reported metrics and the lack of clarity on Welikala's computation of F_1-score.

4.3 Classification Results

4.3.1 Feature Extraction Results

Image features such as colour histogram, Hu moments, LBP, and GLCM features were extracted from images of cropped lesion regions and evaluated for their classification performance using ML models in order to obtain baseline results. The results are provided in Table 18. SVM model achieved a weighted macro-average F_1-score of 0.47, followed by Random Forest and MLP with an F_1-score of 0.46 and 0.45. The confusion matrix for the SVM model is provided in Figure 40 and shows that the model performs poorly across all classes. As shown in Table 18, a dummy classifier which makes predictions based on class distribution in the training set achieves a macro-average F_1-score of 0.33 on the test set. While the feature extraction-based ML models perform better than a dummy classifier, their performance is not on par with CNN models which are discussed in the next section. The success of feature extraction-based methods

heavily relies on manual engineering and selection of features; therefore, further studies on designing and selecting appropriate features may improve the performance of feature extraction-based methods.

Table 18: Classification results of SVM, Random Forest, and MLP models on the test set using a set of feature extractors such as colour histogram, Hu moments, LBP, and GLCM. Precision, recall, and F_1-score are reported as weighted macro-average.

Model	Precision	Recall	F_1-score
SVM	0.47	0.46	0.47
Random Forest	0.49	0.48	0.46
MLP	0.46	0.45	0.45
Dummy Classifier	0.33	0.33	0.33

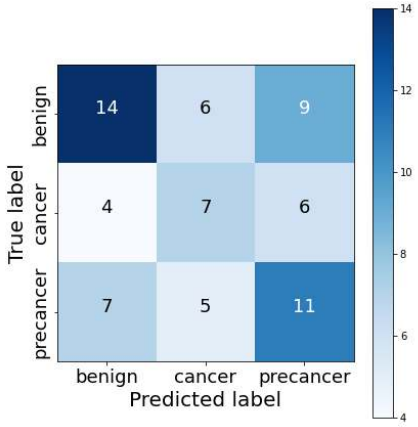


Figure 40: Confusion matrix for SVM model on test set.

4.3.2 CNN Results

Different types of CNN models including an ensemble model of DenseNet-161 and ResNet-152 were evaluated on the test set composed of close-up cropped images of oral lesions and the results are reported in Figure 41 and Table 19. Among all models, Inception-v4 and EfficientNet-b4 achieved the highest scores with a precision of 0.877, a recall of 0.855, and a weighted macro-average F_1-score of 0.858 for Inception-v4 and a precision of 0.869, a recall of 0.855, and a weighted macro-average F_1-score of 0.858 for EfficientNet-b4 on the test set. These are closely followed by our ensemble model and DenseNet-161 that was finetuned in three stages, which achieved a weighted macro-average F_1-score of 0.844 and 0.843 respectively. As detailed in Table 19, finetuning DenseNet-161 and ResNet-152 in three stages helped the models to tackle overfitting and

achieve better performance on the test set. Overall, Inception-v4 and EfficientNet-b4 performed better than all other models, which is expected since these models achieved higher top-1 accuracy on ImageNet dataset as shown in Table 9. While more advanced network architectures of these two models contribute to higher accuracy, another reason for better performance on our test set could be the higher input image size (299 for Inception-v4 and 380 for EfficientNet-b4) compared to the older models such as DenseNet-161 and ResNet-152.

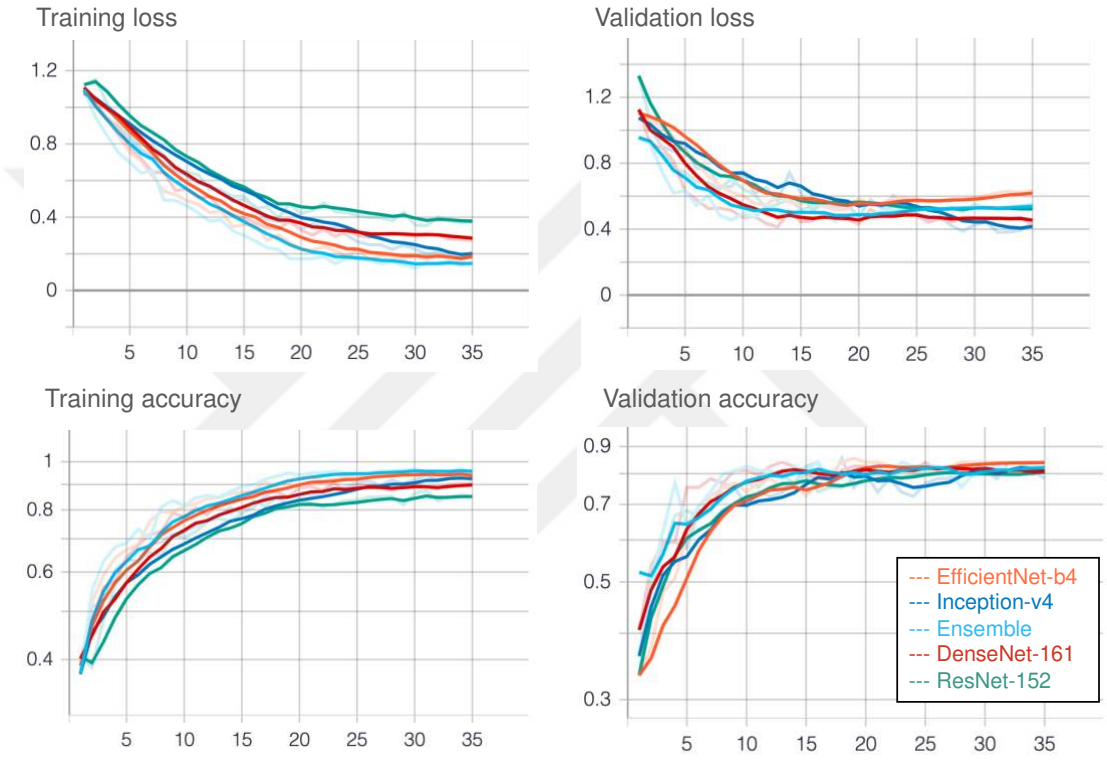


Figure 41: Loss and accuracy plots during training of different CNN models. Only three-staged training results are shown for DenseNet-161 and ResNet-152.

Table 19: Classification results of different CNN models on the test set of cropped lesion regions with best model checkpoints. Precision, recall, and F_1-score are reported as weighted macro-average. Two different training procedures were evaluated for DenseNet-161 and ResNet-152 models: finetuning all layers together and finetuning in three stages* (heads only, x layer onwards, all layers).

Model-Training	Input size	Precision	Recall	F_1-score
Inception-v4	299	0.877	0.855	0.858
EfficientNet-b4	380	0.869	0.855	0.858
DenseNet-161 staged*	224	0.879	0.841	0.844

Ensemble	224	0.849	0.841	0.843
DenseNet-161	224	0.849	0.812	0.816
ResNet-152 staged*	224	0.826	0.812	0.811
ResNet-152	224	0.831	0.812	0.809

Confusion matrices and per-class precision and recall results for the best performing models Inception-v4 and EfficientNet-b4 are provided in Figure 42, Table 20, and Table 21, respectively (results of all models available in Appendix B). Both models have relatively lower precision – 0.72 with Inception-v4 and 0.74 with EfficientNet-b4 – for 'precancer' class compared to benign and cancer classes. As shown in the confusion matrices, some benign and cancerous lesions were misclassified as precancer. Misclassification of cancerous lesions as precancerous is not a consideration of great significance as both types of lesions are considered to be high-risk and usually referred to a medical professional immediately. Misclassification of benign lesions as precancerous, on the other hand, may pose an additional burden on the clinical staff due to increased referrals. Nevertheless, the recall for precancerous lesions is relatively high in both models – 0.91 with Inception-v4 and 0.87 with EfficientNet-b4, which is very promising as these lesions generally have low recall in clinical practice due to their subtle appearance. Among 23 precancerous lesions, only 2 was misclassified as benign with Inception-v4. When the misclassified examples are analysed as displayed in Figure 43, it can be observed that these images are in fact challenging as they look similar to benign cases such as ulcerative lichen planus in our dataset. From clinical perspective, the identification of precancerous and cancerous lesions carries greater importance compared to benign lesions and our models show the capability of identifying these lesions with good recall. In clinical practice, the lack of adequate training and experience of primary care practitioners such as dental care professionals stand as the major reason for overlooked or misclassified oral lesions. In a similar way, deep learning models require large amounts of data that represent all the variations of oral lesions in real world in order to *learn* from the data and perform on par with oral cancer specialists. Hence, increasing the size of the dataset especially for the challenging lesion types and addressing the class imbalance are likely to further improve precision and recall of the models. Moreover, the performance of the models is conditional on the robustness of the ground-truth annotations. Unfortunately, we did not have access to biopsy results for some of the lesion

images which were collected from public resources. Additionally, our dataset was annotated only by a single clinician given the labour-intensive nature of data annotation. In future work, a more systematic data collection process could be implemented to ensure the formation of a more robust dataset.

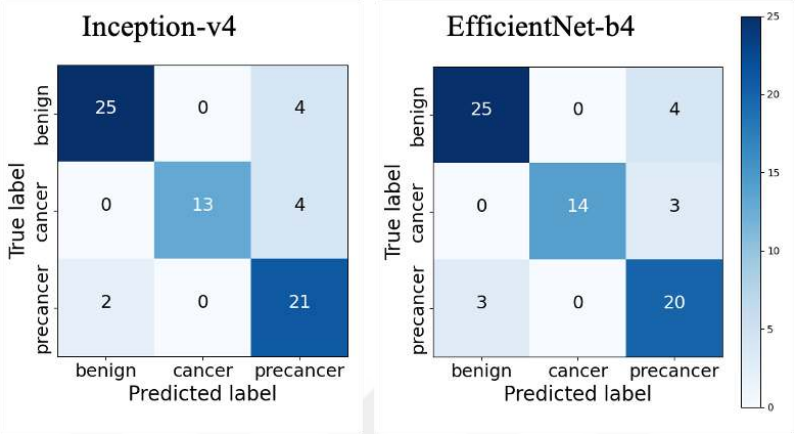


Figure 42: Confusion matrices for Inception-v4 (left) and EfficientNet-b4 (right) models for test set.

Table 20: Precision, recall, and F_1-score results per class on test set for Inception-v4 model.

	Precision	**Recall**	**F1-score**	**Support**
Benign	0.93	0.86	0.89	29
Precancer	0.72	0.91	0.81	23
Cancer	1.00	0.76	0.87	17
Weighted average	0.88	0.86	0.86	69

Table 21: Precision, recall, and F_1-score results per class on test set for EfficientNet-b4 model.

	Precision	**Recall**	**F_1-score**	**Support**
Benign	0.89	0.86	0.88	29
Precancer	0.74	0.87	0.90	23
Cancer	1.00	0.82	0.90	17
Weighted average	0.87	0.86	0.86	69

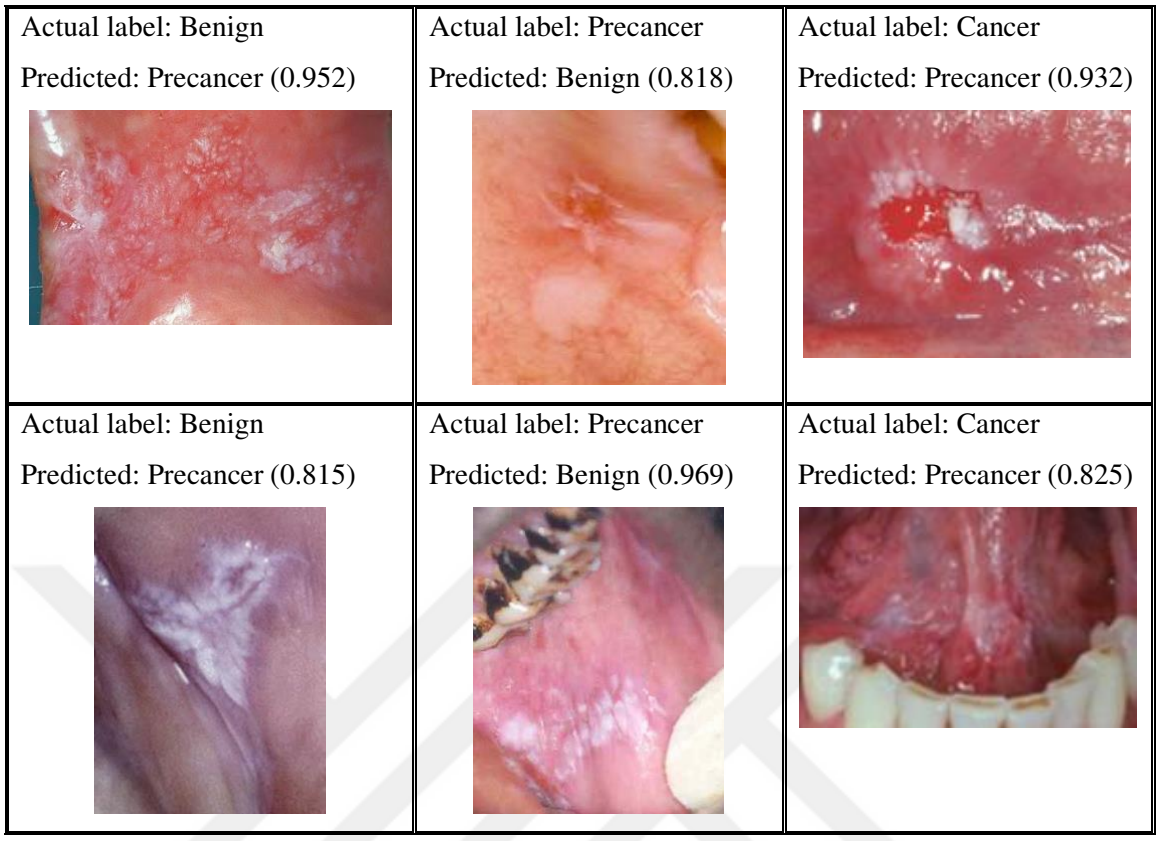


Figure 43: Test set examples that are misclassified by Inception v4 model. The actual (ground-truth) label and the predicted label with corresponding posterior probability are provided for each image.

Feature vectors were extracted from the last layer of the CNN and visualized by t-SNE on a two-dimensional space. As illustrated in Figure 44, the cancer class is well separated from other classes in general whereas the benign and precancer classes are more spread out in the space in both training and test sets, which agrees with our results discussed above.

In order to investigate the impact of lesion surroundings on model performance, we further evaluated the models on test set2 which consists of sole lesion regions cropped as polygons instead of rectangle cropped close-up lesion regions. Polygon cropped lesion images do not contain any other anatomical regions or normal mucosa but only the lesion itself unlike the rectangle cropped images. As detailed on Table 22, the models achieved similar or lower F_1-scores compared to rectangle cropped set and were more prone to overfitting. The results suggest that the surrounding of a lesion especially the colour transition from normal oral mucosa to abnormal tissue is a worthwhile component and facilitates the learning process.

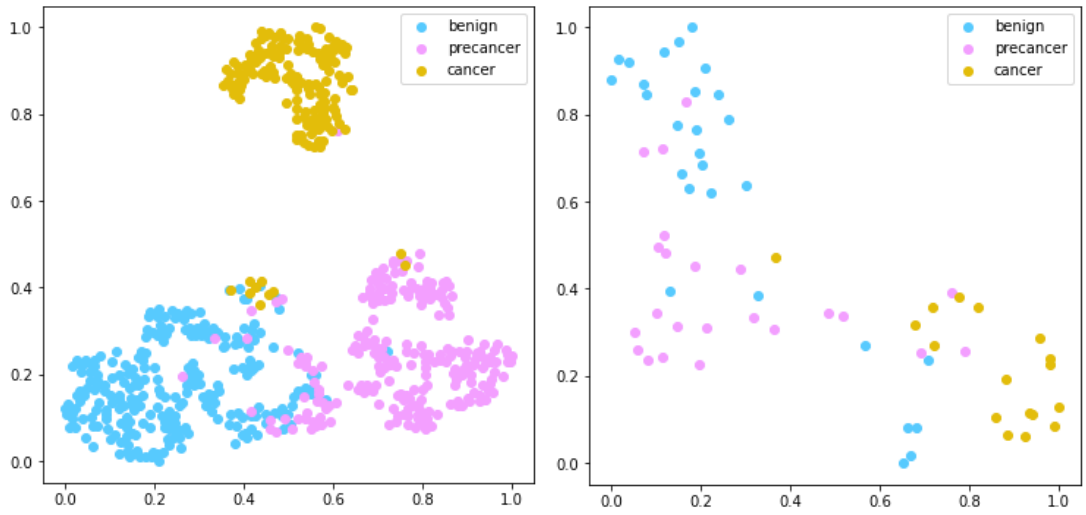


Figure 44: t-SNE visualization in two-dimensions of image feature vectors extracted from the last layer of Inception v4 model. Visualization provided for training set (left) and test set (right).

Table 22: Classification results of different CNN models on the test set of polygon-cropped lesion regions. Precision, recall, and F_1-score are reported as weighted macro-average. Two different training procedures were evaluated for DenseNet-161 and ResNet-152 models: finetuning all layers together and finetuning in three stages* (heads only, x layer onwards, all layers).

Model-Training	Input size	Precision	Recall	F_1-score
Inception-v4	299	0.814	0.812	0.814
EfficientNet-b4	380	0.859	0.855	0.856
DenseNet-161 staged*	224	0.845	0.841	0.838
Ensemble	224	0.818	0.812	0.812
DenseNet-161	224	0.82	0.812	0.811
ResNet-152 staged*	224	0.754	0.739	0.739
ResNet-152	224	0.733	0.71	0.705

We compare our results with the two recently published studies on photographic images of oral lesions. As discussed in section 2.4, Welikala et al. implemented ResNet101 for classifying lesions based on referral decision (Welikala et al., 2020). They reported binary classification of images into ‘non-referral’ or ‘referral’ classes achieving an F_1-score of 78.30% on test set of 204 images. In a separate task, 5-class classification of lesions (‘no lesion’, ‘no referral’, ‘referral for other reasons’, ‘refer – low risk’, ‘refer – cancer / high risk’) was evaluated which produced an F_1 score of 50.57% on the test set.

In a similar study by Fu et al. on a dataset of 5775 images, target lesion patches were cropped based on the detections by an initially trained detector network and then were classified as ‘cancer’ or ‘normal’ by DenseNet-121 (Fu et al., 2020). For this binary classification task, the authors reported 92.3% accuracy on the external validation (test) set and 92.4% mean accuracy on the clinical validation set. Our best performing model, Inception-v4, achieves an F_1-score of 85.8% on a three-class classification task with a dataset of 684 lesion regions which is much smaller than the datasets used in these two studies. A very fine 5-class classification of oral lesions based on referral decision, as investigated by Welikala et al., is challenging and in fact may not be very vital in a clinical setting. Moreover, binary classification of lesions as ‘cancer’ (includes all types of lesions including benign) and ‘normal’ by Fu et al. does not distinguish between benign and precancerous / cancerous lesions, which is as a matter of fact a critical information for making a referral decision. Here, we show that the oral lesions can be classified into three groups based on their risk of progression into oral cancer, with reasonable precision and recall across all classes despite the small size of our dataset and the variability of images. Unlike previous work, we utilized more recent and advanced models for classification such as Inception-v4 and EfficientNet which provided significant improvements in evaluated metrics. Considering both the accuracy and the computational efficiency, EfficientNet-b4 shows great promise for deployment in a mobile or web application.

Additionally, we evaluate EfficientNet-b4 model on cut-out lesion regions that are detected with our pre-trained YOLOv5 model (explained in previous section) on the test set. Among all ground-truth lesion instances, 4 of them were not detected therefore could not be included in this evaluation. When there were multiple overlapping detections for a single lesion, we took the detection with the higher confidence to the next step. Moreover, a few of the same-class lesions which were annotated separately in ground truth were detected in a single bounding box and vice versa, which may have affected the classification results. Ultimately, on a set of 25 benign, 22 precancer, and 16 cancer regions, EfficientNet-b4 achieved a weighted macro-average F_1-score of 0.81 which is slightly lower than our original test set results. This is possibly due to reduced precision of detections by increased recall as we applied TTA to capture most of the lesions if not all. Having larger areas of normal mucosa included in the detected region may have reduced the classification accuracy especially for precancer and cancer classes, despite the taken measures such as data augmentation during training of our models. To address

the issue, recent approaches to improve translation invariance of CNNs can be explored (Kayhan & van Gemert, 2020). Training classification models directly on the test outputs of the detection models might also be beneficial with a dataset large enough to accommodate this task. Future studies with larger datasets are needed to improve robustness and generalization of the models in order to be used in a clinical setting.

4.4 Model deployment

Considering the results of one-class and three-class detection experiments, we find that it is more practicable to implement a lesion detector algorithm with a second-stage classifier to classify the detected lesion. Since the recall of lesions is an important factor for our use case, a two-staged detection and classification would ensure better recall and classification of all types of lesions than a single network performing lesion detection and classification at the same time. Nonetheless, the latter would be computationally more efficient and may outperform a two-staged detection and classification if trained on a large enough dataset to learn both tasks.

For our proposed two-staged detection and classification approach, YOLOv5l was selected for lesion detection task and EfficientNet-b4 for three-class classification of detected lesions. The selected networks achieved overall good performance in terms of both accuracy and inference time, which made them ideal for deployment in a real-world application. The web application deployed with selected models is presented in Figure 45 and Figure 46. It can also be accessed by a mobile phone browser as shown in Figure 47. In addition to uploading saved photographs from library, the application allows taking a photograph and uploading it directly to the platform.

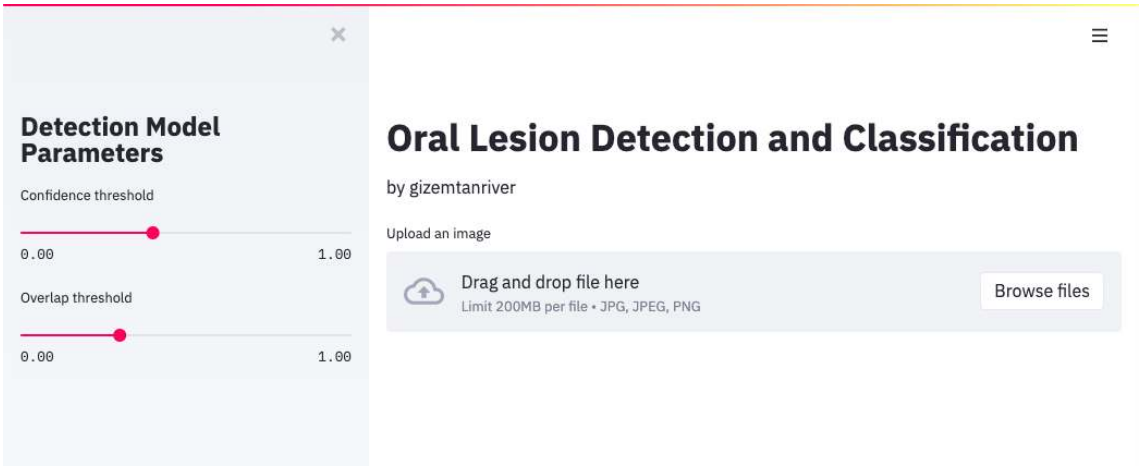


Figure 45: Main screen of our application accessed from a web browser.

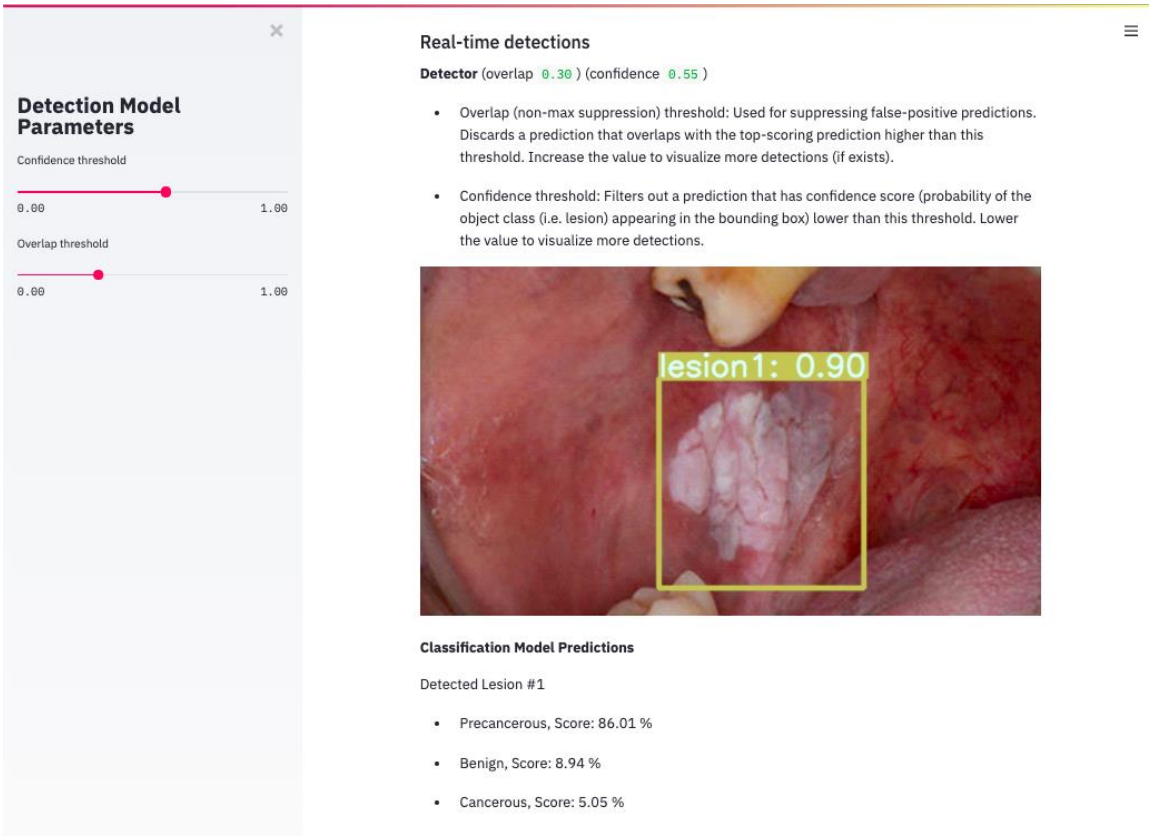


Figure 46: Screenshot from our application accessed from a web browser, showing real-time detections and classification of detected lesions.

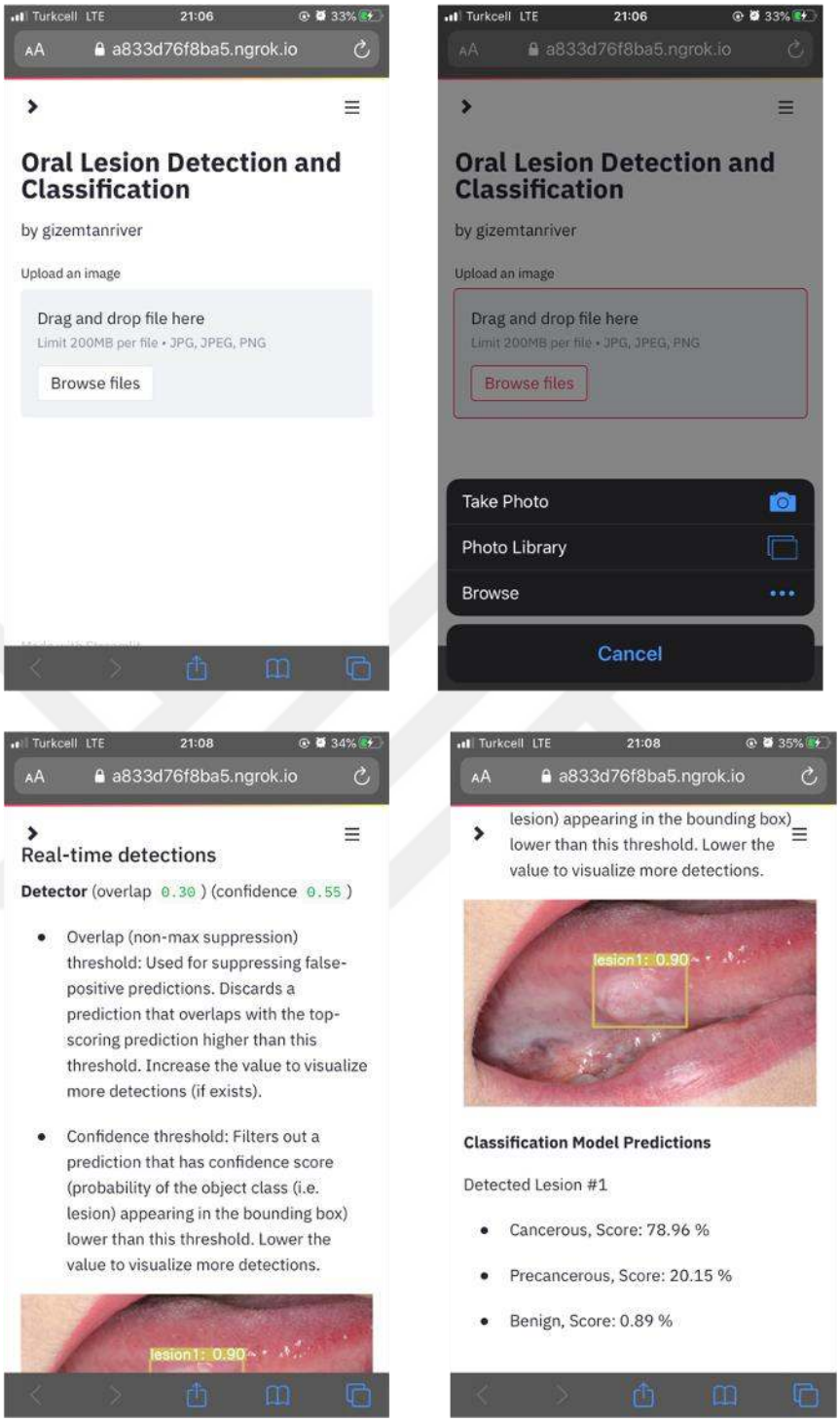


Figure 47: Screenshots from our application accessed from browser using a mobile phone, showing real-time detections and classification of detected lesions.

Chapter 5:
CONCLUSION

In this thesis, we explored the potential applications of various computer vision techniques using deep learning to the oral cancer domain using photographic images. Unlike medical images such as dermoscopy and histopathology, which are typically obtained by using a specialized instrument and are therefore overly standardized, photographic images possess high variability in terms of quality as they can be obtained by different types of devices including smart phones. Besides the photographic variability, oral lesions are commonly hidden or obstructed by other structures present in the image such as teeth and dental tools, which makes segmentation and/or detection of lesion area an essential step for identification of oral lesions. Therefore, we evaluated several segmentation and detection algorithms to isolate region of interest i.e., lesion area from photographic images of oral lesions. For pixel-wise semantic segmentation of oral lesions, we achieved encouraging results with U-Net architecture with EfficientNet-b7 and ResNeXt-101 32x8d backbones, with which we were able to segment a good majority of lesions in our test set successfully. For lesion detection, we studied Mask R-CNN and YOLOv5 architectures which output a bounding box enclosing the detected lesion region (and a segmentation mask in the case of Mask R-CNN). While instance segmentation algorithms such as Mask R-CNN is capable of providing a finer output compared to object detection, they fall behind in terms of inference speed which is of great importance in real-time applications. With the state-of-the-art YOLOv5 architecture, we were able to achieve very promising results at an inference speed of only 10.6 ms per image without TTA. Although the studies on one-class lesion detection and segmentation highlighted the potential of the models, their performance on three-class lesion experiments were not as on par given the greater complexity of the task. Hence, we proposed a two-stage model which aims to detect oral lesions from a given image with a detector network such as YOLOv5 in the first stage and classify the detected lesion as benign, precancerous, or cancer with a classifier network in the second stage. Based on our experiments with various CNN models, EfficientNet-b4 was chosen as the classifier network owing to its accuracy and computational efficiency.

The variable clinical presentation of oral lesions makes their identification challenging for primary care healthcare professionals who are often not adequately

trained or experienced to recognize these lesions (Scully et al., 2008). Furthermore, most cancerous lesions are typically asymptomatic and may appear as innocuous lesions at early stages, leading to late presentation of patients to healthcare centres and ultimately exacerbating diagnostic delay (Markopoulos, 2012). Early detection of precancerous lesions remains as the most effective measure for decreasing mortality and morbidity outcomes from oral cancer (Markopoulos, 2012). Our proposed model enables automated detection and identification of various oral lesion types and paves the way for a low-cost, easy, and non-invasive tool that can support screening processes and improve early diagnosis of oral cancer. It could be also used for remote follow-up of patients who have undergone surgery or received treatment for disease recurrence. Patient can take an image or a video of their oral mucosa using their smart phone and our proposed model can give a warning to the patient and the overseeing specialist if there is any suspicious lesion.

In spite of the promising results obtained in this proof-of-concept study, we acknowledge that there are several limitations to our work. Our dataset is relatively small taking into account the data hungry nature of CNNs and exhibits significant variability both in the quality of images and the presentation of each oral condition. For instance, two separate lesions of the same oral condition may look very different, and likewise two separate lesions resulting from different conditions may look very similar. Despite splitting the dataset into training, validation, and test sets in a randomly stratified fashion, the variations in the dataset can still have an impact on the data distribution of each set. Transfer learning and data augmentation techniques offer great benefits when working with small datasets, yet they cannot be a substitute for more data. A larger dataset with equally presented oral diseases would bring significant gains in building a robust and reliable model and improve generalizability. To account for the variations in image resolution in our dataset, we had to train our models at relatively small resolution. Collecting more data with higher resolution can particularly improve performance of the models that produce finer outputs such as Mask R-CNN. Another limitation is that our dataset was annotated by a single annotator. Since some lesions did not have well-defined boundaries and/or had poor image quality, having a single annotator may have affected the results. In future work, it may be more ideal to collect annotations from multiple clinicians and obtain consensus or majority voting on each annotation whilst forming the dataset. Additionally, we were not able to compare the performance of our models with that of oral cancer specialists or primary care practitioners in this study. Future work

should assess human performance on the modelled tasks, which would provide a reliable reference for the evaluation of our algorithms. Lastly, our image classification model can be extended to include a ‘normal mucosa’ class which may increase the generalizability of the proposed two-staged network in case of false positive detections from the detection network. Another approach for improving the classifier network would be to include clinical and demographical features such as age and gender. This ensemble approach may bring potential benefits in classification performance and can be easily adapted to the data collection process in the context of an application.

BIBLIOGRAPHY

Aggarwal, C. C. (2018). Neural Networks and Deep Learning: A Textbook. In *Artificial Intelligence*. Springer.

Alpaydin, E. (2014). *Introduction to Machine Learning* (3rd ed.). The MIT Press.

Anantharaman, R., Anantharaman, V., & Lee, Y. (2017). Oro Vision: Deep Learning for Classifying Orofacial Diseases. *Proceedings - 2017 IEEE International Conference on Healthcare Informatics, ICHI 2017*, 39–45. https://doi.org/10.1109/ICHI.2017.69

Anantharaman, R., Velazquez, M., & Lee, Y. (2019). Utilizing Mask R-CNN for Detection and Segmentation of Oral Diseases. *Proceedings - 2018 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2018*, 2197–2204. https://doi.org/10.1109/BIBM.2018.8621112

Aubreville, M., Knipfer, C., Oetter, N., Jaremenko, C., Rodner, E., Denzler, J., Bohr, C., Neumann, H., Stelzle, F., & Maier, A. (2017). Automatic Classification of Cancerous Tissue in Laserendomicroscopy Images of the Oral Cavity using Deep Learning. *Scientific Reports*, *7*(1). https://doi.org/10.1038/s41598-017-12320-8

Barkan, O., Weill, J., Wolf, L., & Aronowitz, H. (2013). Fast high dimensional vector multiplication face recognition. *Proceedings of the IEEE International Conference on Computer Vision*, 1960–1967. https://doi.org/10.1109/ICCV.2013.246

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning* (M. Jordan, J. Kleinberg, & B. Scholkopf (eds.)). Springer.

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection*.

Borgli, H., Thambawita, V., Smedsrud, P., Hicks, S., Jha, D., Eskeland, S., Randel, K. R., Pogorelov, K., Lux, M., Dang-Nguyen, D.-T., Johansen, D., Griwodz, C., Stensland, H., Garcia-Ceja, E., Schmidt, P., Hammer, H., Riegler, M., Halvorsen, P., & de Lange, T. (2019). *HyperKvasir, A Comprehensive Multi-Class Image and*

Video Dataset for Gastrointestinal Endoscopy. https://doi.org/10.31219/osf.io/mkzcq

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Burzynski, N. J., Rankin, K. V., Silverman, S., Scheetz, J. P., & Jones, D. L. (2002). Graduating dental students' perceptions of oral cancer education: Results of an exit survey of seven dental schools. *Journal of Cancer Education*, *17*(2), 83–84. https://doi.org/10.1080/08858190209528804

Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., & Kalinin, A. A. (2020). Albumentations: Fast and flexible image augmentations. *Information (Switzerland)*, *11*(2), 125. https://doi.org/10.3390/info11020125

Cadene, R. (2020). *cadene/pretrained-models.pytorch: Pretrained ConvNets for pytorch*. GitHub. https://github.com/Cadene/pretrained-models.pytorch#modelinput_size

Cai, Z., & Vasconcelos, N. (2017). Cascade R-CNN: Delving into High Quality Object Detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 6154–6162. https://doi.org/10.1109/CVPR.2018.00644

Ching, C. S. (2018). MeMoSa: Mobile Mouth Screening Anywhere for Early Detection of Oral Cancer. *Journal of Global Oncology*, *4*(Supplement 2), 56s-56s. https://doi.org/10.1200/jgo.18.33300

COCO Consortium. (2016). *COCO - Common Objects in Context - Detection Evaluation*. https://cocodataset.org/#detection-eval

Coleman, H., Dahlstrom, J., Hylam, D., Allende, A., Nguyen, B., Samra, S., Sundaresan, P., & Wong, E. (2019). Carcinomas of the oral cavity structured reporting protocol. In M. Judge & C. Selinger (Eds.), *RCPA* (2nd ed.).

da Silva, S. D., Hier, M., Mlynarek, A., Kowalski, L. P., & Alaoui-Jamali, M. A. (2012). Recurrent oral cancer: Current and emerging therapeutic approaches. *Frontiers in Pharmacology*, *3 JUL*. https://doi.org/10.3389/fphar.2012.00149

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, *I*, 886–893. https://doi.org/10.1109/CVPR.2005.177

Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, & Li Fei-Fei. (2009). *ImageNet: A large-scale hierarchical image database*. 248–255. https://doi.org/10.1109/cvpr.2009.5206848

Dutta, A., Gupta, A., & Zissermann, A. (2016). *VGG Image Annotator (VIA)*. https://www.robots.ox.ac.uk/~vgg/software/via/

Dutta, Abhishek, & Zisserman, A. (2019). The VIA annotation software for images, audio and video. *MM 2019 - Proceedings of the 27th ACM International Conference on Multimedia*, 2276–2279. https://doi.org/10.1145/3343031.3350535

Epstein, J. B., Güneri, P., Boyacioglu, H., & Abt, E. (2012). The limitations of the clinical oral examination in detecting dysplastic oral lesions and oral squamous cell carcinoma. *Journal of the American Dental Association*, *143*(12), 1332–1342. https://doi.org/10.14219/jada.archive.2012.0096

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, *542*(7639), 115–118. https://doi.org/10.1038/nature21056

Fu, Q., Chen, Y., Li, Z., Jing, Q., Hu, C., Liu, H., Bao, J., Hong, Y., Shi, T., Li, K., Zou, H., Song, Y., Wang, H., Wang, X., Wang, Y., Liu, J., Liu, H., Chen, S., Chen, R., … Xiong, X. (2020). A deep learning algorithm for detection of oral cavity squamous cell carcinoma from photographic images: A retrospective study. *EClinicalMedicine*, *27*, 100558. https://doi.org/10.1016/j.eclinm.2020.100558

Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Martinez-Gonzalez, P., & Garcia-Rodriguez, J. (2018). A survey on deep learning techniques for image and video semantic segmentation. *Applied Soft Computing Journal*, *70*, 41–65. https://doi.org/10.1016/j.asoc.2018.05.018

Ghatwary, N., Ye, X., & Zolgharni, M. (2019). Esophageal Abnormality Detection Using DenseNet Based Faster R-CNN with Gabor Features. *IEEE Access*, *7*, 84374–84385. https://doi.org/10.1109/ACCESS.2019.2925585

Ghodrati, M., Farzmahdi, A., Rajaei, K., Ebrahimpour, R., & Khaligh-Razavi, S. M. (2014). Feedforward object-vision models only tolerate small image variations compared to human. *Frontiers in Computational Neuroscience*, *8*(JUL), 74. https://doi.org/10.3389/fncom.2014.00074

Girshick, R. (2015). Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, *2015 Inter*, 1440–1448. https://doi.org/10.1109/ICCV.2015.169

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 580–587. https://doi.org/10.1109/CVPR.2014.81

Gupta, R. K., Kaur, M., & Manhas, J. (2019). Tissue Level Based Deep Learning Framework for Early Detection of Dysplasia in Oral Squamous Epithelium. *Journal of Multimedia Information System*, *6*(2), 81–86. https://doi.org/10.33851/jmis.2019.6.2.81

Gurumurthy, V. A., Kestur, R., & Narasipura, O. (2019). *Mango Tree Net -- A fully convolutional network for semantic segmentation and individual crown detection of mango trees*.

Hafiz, A. M., & Bhat, G. M. (2020). A survey on instance segmentation: state of the art. *International Journal of Multimedia Information Retrieval*, *9*(3), 171–189. https://doi.org/10.1007/s13735-020-00195-x

Halicek, M., Lu, G., Little, J. V., Wang, X., Patel, M., Griffith, C. C., El-Deiry, M. W., Chen, A. Y., & Fei, B. (2017). Deep convolutional neural networks for classifying head and neck cancer using hyperspectral imaging. *Journal of Biomedical Optics*, *22*(6), 060503. https://doi.org/10.1117/1.jbo.22.6.060503

Hall-Beyer, M. (2017). GLCM Texture: A Tutorial v. 3.0 March 2017. *Arts Research &*

Publications, *2017–03*, 75.

Handuo, Z. (2018, August 20). *You only look once (YOLO) -- (1)*. GitHub. https://zhanghanduo.github.io/post/yolo1/

Haralick, R. M. (1979). Statistical and structural approaches to texture. *Proceedings of the IEEE*, *67*(5), 786–804. https://doi.org/10.1109/PROC.1979.11328

Haralick, R. M., Dinstein, I., & Shanmugam, K. (1973). Textural Features for Image Classification. *IEEE Transactions on Systems, Man and Cybernetics*, *SMC-3*(6), 610–621. https://doi.org/10.1109/TSMC.1973.4309314

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(2), 386–397. https://doi.org/10.1109/TPAMI.2018.2844175

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 770–778. https://doi.org/10.1109/CVPR.2016.90

He, K., Zhang, X., Ren, S., & Sun, J. (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8691 LNCS*(PART 3), 346–361. https://doi.org/10.1007/978-3-319-10578-9_23

Heidari, A. E., Sunny, S. P., James, B. L., Lam, T., Tran, A. V., Yu, J., Ramanjinappa, R. D., K, U., Birur, P., Suresh, A., Kuriakose, M. A., Chen, Z., & Wilder-Smith, P. (2018). Optical Coherence Tomography as an Oral Cancer Screening Adjunct in a Low Resource Settings. *IEEE Journal of Selected Topics in Quantum Electronics*. https://doi.org/10.1109/JSTQE.2018.2869643

Hendrycks, D. (2015). *CONVEX OPTIMIZATION, DUALITY, AND THEIR APPLICATION TO SUPPORT VECTOR MACHINES*.

Hu, M. K. (1962). Visual Pattern Recognition by Moment Invariants. *IRE Transactions on Information Theory*, *8*(2), 179–187. https://doi.org/10.1109/TIT.1962.1057692

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 2261–2269. https://doi.org/10.1109/CVPR.2017.243

Hui, J. (2018, March 7). *mAP (mean Average Precision) for Object Detection*. Medium. https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173

Ilhan, B., Lin, K., Guneri, P., & Wilder-Smith, P. (2020). Improving Oral Cancer Outcomes with Imaging and Artificial Intelligence. *Journal of Dental Research*, *99*(3), 241–248. https://doi.org/10.1177/0022034520902128

Jaremenko, C., Maier, A., Steidl, S., Hornegger, J., Oetter, N., Knipfer, C., Stelzle, F., & Neumann, H. (2015). Classification of confocal laser endomicroscopic images of the oral cavity to distinguish pathological from healthy tissue. *Informatik Aktuell*, 479–485. https://doi.org/10.1007/978-3-662-46224-9_82

Jeyaraj, P. R., & Samuel Nadar, E. R. (2019). Computer-assisted medical image classification for early diagnosis of oral cancer employing deep learning algorithm. *Journal of Cancer Research and Clinical Oncology*, *145*(4), 829–837. https://doi.org/10.1007/s00432-018-02834-7

Jiao, L., & Zhao, J. (2019). A Survey on the New Generation of Deep Learning in Image Processing. *IEEE Access*, *7*, 172231–172263. https://doi.org/10.1109/ACCESS.2019.2956508

Jocher, G., Stoken, A., Borovec, J., ChristopherSTAN, Changyu, L., Hogan, A., Laughing, NanoCode012, yxNONG, Diaconu, L., lorenzomammana, wanghaoyang0106, ml5ah, Doug, Poznanski, J., 于力军 L. Y., changyu98, Rai, P., Ferriday, R., … yzchen. (2020). ultralytics/yolov5: v2.0. *GitHub*. https://doi.org/10.5281/ZENODO.3958273

Jocher, G., Stoken, A., Borovec, J., NanoCode012, ChristopherSTAN, Changyu, L., Laughing, tkianai, Hogan, A., lorenzomammana, yxNONG, AlexWang1900,

Diaconu, L., Marc, wanghaoyang0106, ml5ah, Doug, Ingham, F., Frederik, … Rai, P. (2020). *ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements*. https://doi.org/10.5281/ZENODO.4154370

Kayhan, O. S., & van Gemert, J. C. (2020). On translation invariance in CNNs: Convolutional layers can exploit absolute spatial location. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 14262–14273. https://doi.org/10.1109/CVPR42600.2020.01428

Lane, P. M., Gilhuly, T., Whitehead, P., Zeng, H., Poh, C. F., Ng, S., Williams, P. M., Zhang, L., Rosin, M. P., & MacAulay, C. E. (2006). Simple device for the direct visualization of oral-cavity tissue fluorescence. *Journal of Biomedical Optics*, *11*(2), 024006. https://doi.org/10.1117/1.2193157

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2323. https://doi.org/10.1109/5.726791

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2016). *Feature Pyramid Networks for Object Detection*.

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8693 LNCS*(PART 5), 740–755. https://doi.org/10.1007/978-3-319-10602-1_48

Lingen, M. W., Abt, E., Agrawal, N., Chaturvedi, A. K., Cohen, E., D'Souza, G., Gurenlian, J. A., Kalmar, J. R., Kerr, A. R., Lambert, P. M., Patton, L. L., Sollecito, T. P., Truelove, E., Tampi, M. P., Urquhart, O., Banfield, L., & Carrasco-Labra, A. (2017). Evidence-based clinical practice guideline for the evaluation of potentially malignant disorders in the oral cavity: A report of the American Dental Association. *Journal of the American Dental Association*, *148*(10), 712-727.e10. https://doi.org/10.1016/j.adaj.2017.07.032

Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path Aggregation Network for Instance

Segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 8759–8768. https://doi.org/10.1109/CVPR.2018.00913

Long, J., Shelhamer, E., & Darrell, T. (2014). Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(4), 640–651.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

Macey, R., Walsh, T., Brocklehurst, P., Kerr, A. R., Liu, J. L., Lingen, M. W., Ogden, G. R., Warnakulasuriya, S., & Scully, C. (2015). Diagnostic tests for oral cancer and potentially malignant disorders in patients presenting with clinically evident lesions. In *Cochrane Database of Systematic Reviews* (Vol. 2015, Issue 5). John Wiley and Sons Ltd. https://doi.org/10.1002/14651858.CD010276.pub2

Markopoulos, A. K. (2012). Current Aspects on Oral Squamous Cell Carcinoma. *The Open Dentistry Journal*, *6*(1), 126–130. https://doi.org/10.2174/1874210601206010126

Matsumoto, K. (2011). [Detection of potentially malignant and malignant lesions of oral cavity using autofluorescence visualization device]. *Kōkūbyō Gakkai Zasshi. The Journal of the Stomatological Society, Japan*, *78*(2), 73–80.

Melas-Kyriazi, L. (2020). *lukemelas/EfficientNet-PyTorch: A PyTorch implementation of EfficientNet*. GitHub. https://github.com/lukemelas/EfficientNet-PyTorch

Misra, D. (2019). *Mish: A Self Regularized Non-Monotonic Activation Function*.

MissingLink.ai. (2019). *Convolutional Neural Network Tutorial: From Basic to Advanced*. https://missinglink.ai/guides/convolutional-neural-networks/convolutional-neural-network-tutorial-basic-advanced/

Ojala, T., Pietikäinen, M., & Harwood, D. (1994). Performance evaluation of texture measures with classification based on Kullback discrimination of distributions.

Proceedings - International Conference on Pattern Recognition, *3*, 582–585. https://doi.org/10.1109/ICPR.1994.576366

Ojala, T., Pietikäinen, M., & Mäenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(7), 971–987. https://doi.org/10.1109/TPAMI.2002.1017623

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. Alch\'{e}-Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 32* (pp. 8024--8035). Curran Associates, Inc.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, Mathieu and Prettenhofer, P., Weiss, R., Dubourg, V., & ...others. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(Oct), 2825–2830.

Petersen, P. E. (2005). *GLOBAL DATA ON INCIDENCE OF ORAL CANCER*.

Rahman, M. S., Ingole, N., Roblyer, D., Stepanek, V., Richards-Kortum, R., Gillenwater, A., Shastri, S., & Chaturvedi, P. (2010). Evaluation of a low-cost, portable imaging system for early detection of oral cancer. *Head and Neck Oncology*, *2*(1), 10. https://doi.org/10.1186/1758-3284-2-10

Raman, R., Srinivasan, S., Virmani, S., Sivaprasad, S., Rao, C., & Rajalakshmi, R. (2019). Fundus photograph-based deep learning algorithms in detecting diabetic retinopathy. *Eye (Basingstoke)*, *33*(1), 97–109. https://doi.org/10.1038/s41433-018-0269-y

Rana, A., Yauney, G., Wong, L. C., Gupta, O., Muftu, A., & Shah, P. (2017). Automated segmentation of gingival diseases from oral images. *2017 IEEE Healthcare Innovations and Point of Care Technologies, HI-POCT 2017*, *2017-*

Decem, 144–147. https://doi.org/10.1109/HIC.2017.8227605

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 779–788. https://doi.org/10.1109/CVPR.2016.91

Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement*.

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 6517–6525. https://doi.org/10.1109/CVPR.2017.690

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

Rezvantalab, A., Safigholi, H., & Karimijeshni, S. (2018). *Dermatologist Level Dermoscopy Skin Cancer Classification Using Different Deep Learning Convolutional Neural Networks Algorithms*.

Roblyer, D., Kurachi, C., Stepanek, V., Williams, M. D., El-Naggar, A. K., Lee, J. J., Gillenwater, A. M., & Richards-Kortum, R. (2009). Objective detection and delineation of oral neoplasia using autofluorescence imaging. *Cancer Prevention Research*, *2*(5), 423–431. https://doi.org/10.1158/1940-6207.CAPR-08-0229

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9351*, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

Russell, S. J., & Norvig, P. (2010). *Artificial Intelligence A Modern Approach* (3rd ed.). Prentice Hall.

Scully, C., Bagan, J. V., Hopper, C., & Epstein, J. B. (2008). Oral cancer: Current and future diagnostic techniques. *American Journal of Dentistry*, *21*(4), 199–209.

Seoane, J., Alvarez-Novoa, P., Gomez, I., Takkouche, B., Diz, P., Warnakulasiruya, S., Seoane-Romero, J. M., & Varela-Centelles, P. (2016). Early oral cancer diagnosis: The Aarhus statement perspective. A systematic review and meta-analysis. In *Head and Neck* (Vol. 38, pp. E2182–E2189). John Wiley and Sons Inc. https://doi.org/10.1002/hed.24050

Shamim, M. Z. M., Syed, S., Shiblee, M., Usman, M., & Ali, S. (2019). Automated detection of oral pre-cancerous tongue lesions using deep learning for early diagnosis of oral cavity cancer. *ArXiv*, *abs/1909.0*(September), 1–25. https://doi.org/10.13140/RG.2.2.28808.16643

Shapiro, L., & Stockman, G. (2000). *Computer Vision*. Prentice Hall.

Silverman, S. (1988). Early diagnosis of oral cancer. *Cancer*, *62*(1 S), 1796–1799. https://doi.org/10.1002/1097-0142(19881015)62:1+<1796::AID-CNCR2820621319>3.0.CO;2-E

Song, B., Sunny, S., Uthoff, R. D., Patrick, S., Suresh, A., Kolur, T., Keerthi, G., Anbarani, A., Wilder-Smith, P., Kuriakose, M. A., Birur, P., Rodriguez, J. J., & Liang, R. (2018). Automatic classification of dual-modalilty, smartphone-based oral dysplasia and malignancy images using deep learning. *Biomedical Optics Express*, *9*(11), 5318. https://doi.org/10.1364/boe.9.005318

Svistun, E., Alizadeh-Naderi, R., El-Naggar, A., Jacob, R., Gillenwater, A., & Richards-Kortum, R. (2004). Vision enhancement system for detection of oral cavity neoplasia based on auto fluorescence. *Head and Neck*, *26*(3), 205–215. https://doi.org/10.1002/hed.10381

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, inception-ResNet and the impact of residual connections on learning. *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, 4278–4284.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *07-12-June*, 1–9.

https://doi.org/10.1109/CVPR.2015.7298594

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 2818–2826. https://doi.org/10.1109/CVPR.2016.308

Szeliski, R. (2011). Computer Vision Algorithms and Applications. In *Springer Tracts in Advanced Robotics* (Vol. 73). Springer. https://doi.org/10.1007/978-3-642-20144-8_11

Taha, A. A., & Hanbury, A. (2015). Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool. *BMC Medical Imaging*, *15*(1), 29. https://doi.org/10.1186/s12880-015-0068-x

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. *36th International Conference on Machine Learning, ICML 2019*, *2019-June*, 10691–10700.

Tan, M., Pang, R., & Le, Q. V. (2020). *EfficientDet: Scalable and Efficient Object Detection*. 10778–10787. https://doi.org/10.1109/cvpr42600.2020.01079

Tatehara, S., & Satomura, K. (2020). Non-invasive diagnostic system based on light for detecting early-stage oral cancer and high-risk precancerous lesions—potential for dentistry. In *Cancers* (Vol. 12, Issue 11, pp. 1–15). MDPI AG. https://doi.org/10.3390/cancers12113185

Teixeira, T. (2020). *streamlit/streamlit: Streamlit — The fastest way to build data apps in Python*. GitHub. https://github.com/streamlit/streamlit

Thomas, B., Kumar, V., & Saini, S. (2013). Texture analysis based segmentation and classification of oral cancer lesions in color images using ANN. *2013 IEEE International Conference on Signal Processing, Computing and Control, ISPCC 2013*. https://doi.org/10.1109/ISPCC.2013.6663401

Uthoff, R. D., Song, B., Sunny, S., Patrick, S., Suresh, A., Kolur, T., Gurushanth, K., Wooten, K., Gupta, V., Platek, M. E., Singh, A. K., Wilder-Smith, P., Kuriakose,

M. A., Birur, P., & Liang, R. (2019). Small form factor, flexible, dual-modality handheld probe for smartphone-based, point-of-care oral and oropharyngeal cancer screening. *Journal of Biomedical Optics*, *24*(10), 1. https://doi.org/10.1117/1.jbo.24.10.106003

Uthoff, R. D., Song, B., Sunny, S., Patrick, S., Suresh, A., Kolur, T., Keerthi, G., Spires, O., & Anbarani, A. (2018). *Point-of-care, smartphone-based, dual- modality, dual-view, oral cancer screening device with neural network classification for low-resource communities*. 1–21. https://doi.org/10.5281/zenodo.1214892

Uthoff, R. D., Wilder-Smith, P., Sunny, S., Suresh, A., Patrick, S., Anbarani, A., Liang, R., Song, B., Birur, P., Kuriakose, M. A., & Spires, O. (2018). Development of a dual-modality, dual-view smartphone-based imaging system for oral cancer detection. In R. Raghavachari, R. Liang, & T. J. Pfefer (Eds.), *Design and Quality for Biomedical Technologies XI* (Vol. 10486, p. 30). SPIE. https://doi.org/10.1117/12.2296435

Van der Walt, S., Sch"onberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., & Yu, T. (2014). scikit-image: image processing in Python. *PeerJ*, *2*, e453.

Villa, A., Villa, C., & Abati, S. (2011). Oral cancer and oral erythroplakia: An update and implication for clinicians. In *Australian Dental Journal* (Vol. 56, Issue 3, pp. 253–256). John Wiley & Sons, Ltd. https://doi.org/10.1111/j.1834-7819.2011.01337.x

Wang, C. Y., Mark Liao, H. Y., Wu, Y. H., Chen, P. Y., Hsieh, J. W., & Yeh, I. H. (2020). CSPNet: A new backbone that can enhance learning capability of CNN. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, *2020-June*, 1571–1580. https://doi.org/10.1109/CVPRW50498.2020.00203

Warnakulasuriya, S., & Ariyawardana, A. (2016). Malignant transformation of oral leukoplakia: A systematic review of observational studies. *Journal of Oral Pathology and Medicine*, *45*(3), 155–166. https://doi.org/10.1111/jop.12339

Warnakulasuriya, Saman, & Greenspan, J. S. (2020). *Textbook of Oral Cancer: Prevention, Diagnosis and Management*.

Welikala, R. A., Remagnino, P., Lim, J. H., Chan, C. S., Rajendran, S., Kallarakkal, T. G., Zain, R. B., Jayasinghe, R. D., Rimal, J., Kerr, A. R., Amtha, R., Patil, K., Tilakaratne, W. M., Gibson, J., Cheong, S. C., & Barman, S. A. (2020). Automated Detection and Classification of Oral Lesions Using Deep Learning for Early Detection of Oral Cancer. *IEEE Access*, *8*, 132677–132693. https://doi.org/10.1109/ACCESS.2020.3010180

WHO. (2018). *Oral cancer*. WHO. https://www.who.int/cancer/prevention/diagnosis-screening/oral-cancer/en/

Wieslander, H., Forslid, G., Bengtsson, E., Wählby, C., Hirsch, J. M., Stark, C. R., & Sadanandan, S. K. (2017). Deep Convolutional Neural Networks for Detecting Cellular Changes Due to Malignancy. *Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017*, *2018-Janua*, 82–89. https://doi.org/10.1109/ICCVW.2017.18

Wilder-Smith, P., Lee, K., Guo, S., Zhang, J., Osann, K., Chen, Z., & Messadi, D. (2009). In vivo diagnosis of oral dysplasia and malignancy using optical coherence tomography: Preliminary studies in 50 patients. *Lasers in Surgery and Medicine*, *41*(5), 353–357. https://doi.org/10.1002/lsm.20773

Woo, S., Park, J., Lee, J. Y., & Kweon, I. S. (2018). CBAM: Convolutional block attention module. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11211 LNCS*, 3–19. https://doi.org/10.1007/978-3-030-01234-2_1

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., & Girshick, R. (2019). *Detectron2*. GitHub. https://github.com/facebookresearch/detectron2

Xu, S., Liu, Y., Hu, W., Zhang, C., Liu, C., Zong, Y., Chen, S., Lu, Y., Yang, L., Ng, E. Y. K., Wang, Y., & Wang, Y. (2019). An Early Diagnosis of Oral Cancer based on Three-Dimensional Convolutional Neural Networks. *IEEE Access*, *7*, 158603–158611. https://doi.org/10.1109/ACCESS.2019.2950286

Yakubovskiy, P. (2020). *Segmentation Models Pytorch*. GitHub. https://github.com/qubvel/segmentation_models.pytorch

Yun, S., Han, D., Chun, S., Oh, S. J., Choe, J., & Yoo, Y. (2019). CutMix: Regularization strategy to train strong classifiers with localizable features. *Proceedings of the IEEE International Conference on Computer Vision*, *2019-Octob*, 6022–6031. https://doi.org/10.1109/ICCV.2019.00612

Zhao, S., Luo, Q., & Liu, C. (2020). *Automatic Tooth Segmentation and Classification in Dental Panoramic X-ray Images*. https://doi.org/10.21203/RS.3.RS-89894/V1

Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object Detection with Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems*, *30*(11), 3212–3232. https://doi.org/10.1109/TNNLS.2018.2876865

Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, D. (2020). Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(07), 12993–13000. https://doi.org/10.1609/aaai.v34i07.6999

Appendix A

Further training plots for YOLOv5 experiments are presented in this section.

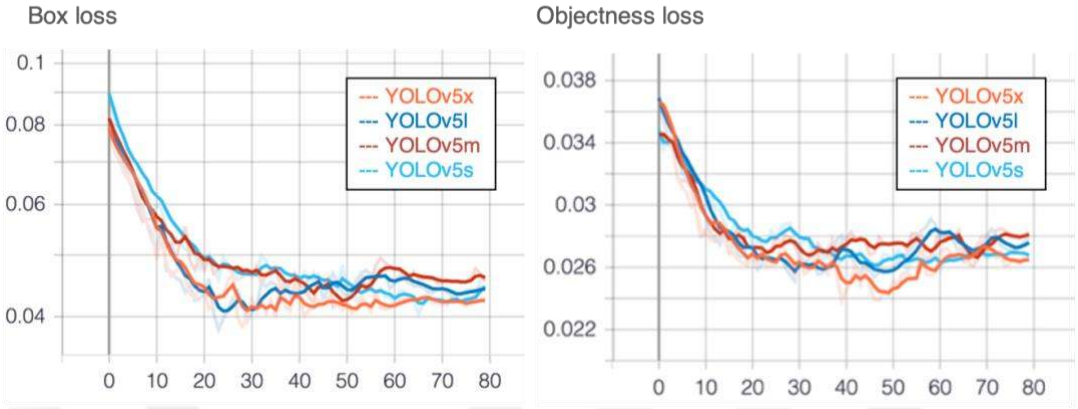


Figure 48: Plots of box loss (left) and objectness loss (right) on validation set during training for 80 epochs on one-class lesion detection task. Note that there is no classification loss for this one-class task.

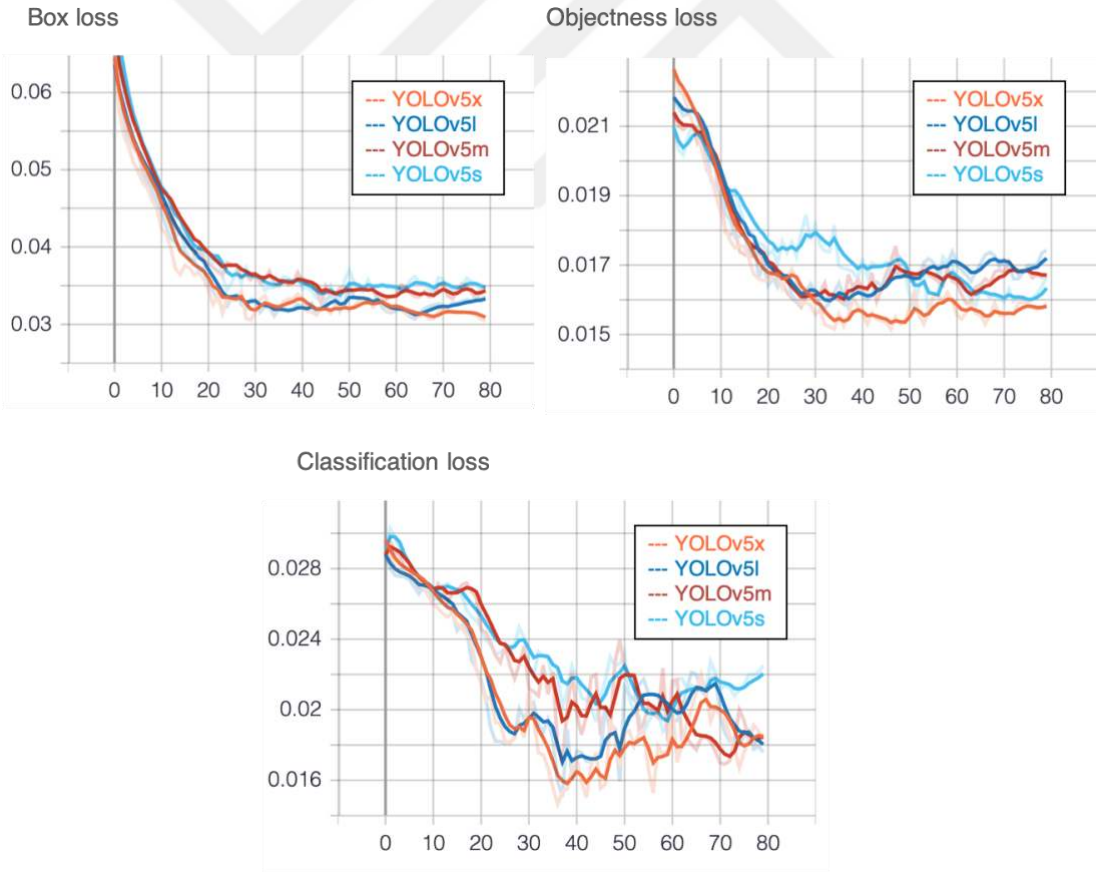


Figure 49: Plots of box loss (top left), objectness loss (top right), and classification loss (bottom) on validation set during training for 80 epochs on three-class lesion detection task. The models struggle with lesion classification in this task.

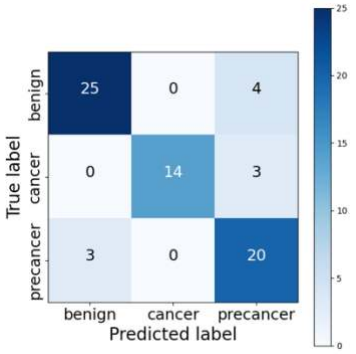
Appendix B

Detailed results of classification experiments are presented in this section.

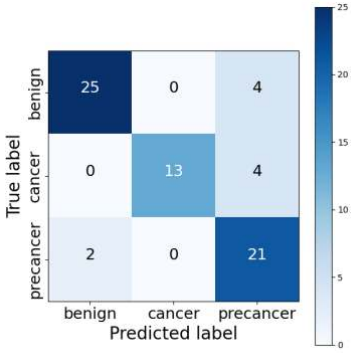
Table 23: Accuracy and F_1-scores of evaluated CNN models on the validation and tests sets of cropped lesion regions with best model checkpoints. F_1-score is reported as weighted macro-average. Two different training procedures were evaluated for DenseNet-161 and ResNet-152 models: finetuning all layers together and finetuning in three stages* (heads only, x layer onwards, all layers).

Model	**Accuracy$_{val}$**	**F$_1$-score$_{val}$**	**Accuracy$_{test}$**	**F$_1$-score$_{test}$**
EfficientNet-b4	0.873	0.874	0.855	0.858
Inception-v4	0.873	0.867	0.855	0.858
Ensemble	0.841	0.839	0.841	0.843
DenseNet-161 staged	0.841	0.838	0.841	0.844
DenseNet-161	0.825	0.821	0.812	0.816
ResNet-152 staged	0.825	0.822	0.812	0.811
ResNet-152	0.81	0.807	0.812	0.809

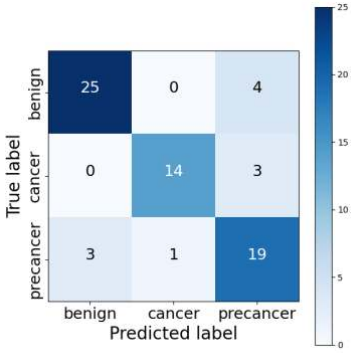
EfficientNet-b4



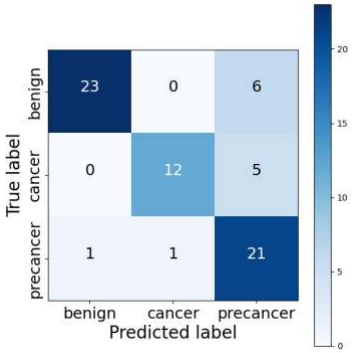
Inception-v4



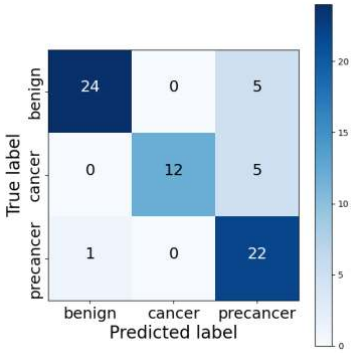
Ensemble



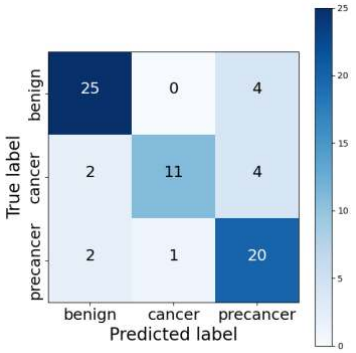
DenseNet-161



DenseNet-161 staged



ResNet-152



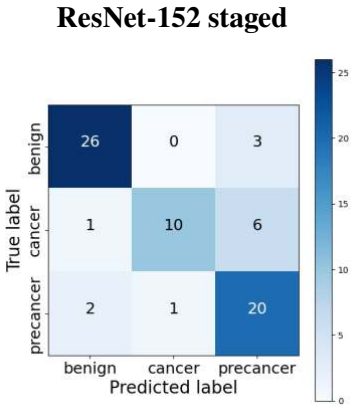


Figure 50: Confusion matrices for all CNN models on the test set.

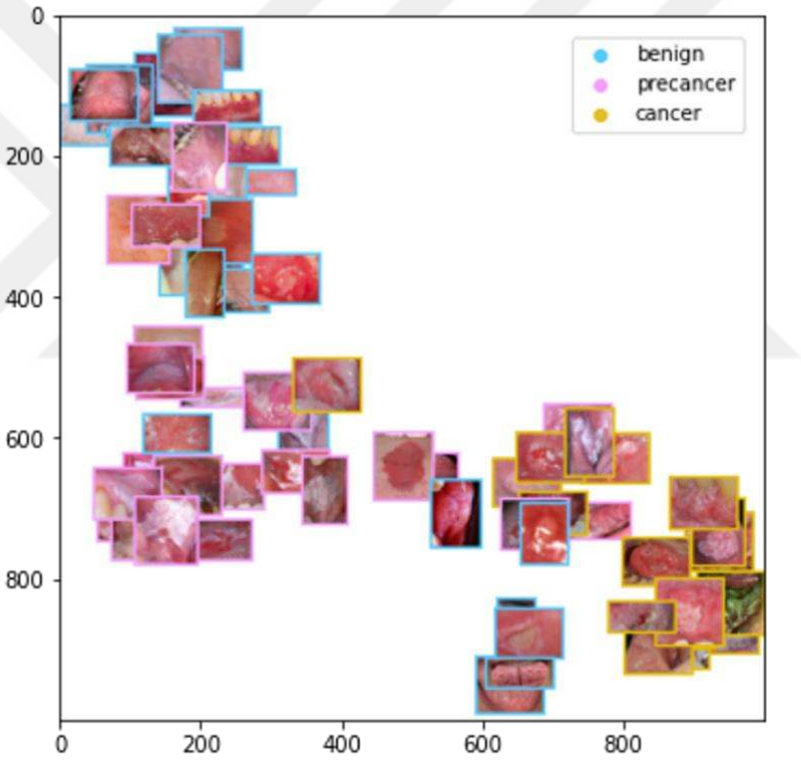


Figure 51: t-SNE visualization of image feature vectors extracted from the last layer of Inception v4 model with corresponding images. The border colour of the images represents the predicted class for the image as shown in the legend.