



**UiT** The Arctic University of Norway

Faculty of Science and Technology  
Department of Physics and Technology

## **Automatic detection of the mental foramen for estimating mandibular cortical width in dental panoramic radiographs**

Isak Paasche Edvardsen

STA-3941 Master's thesis in applied physics and mathematics - 30 stp - December 2021

This thesis document was typeset using the *UiT Thesis L<sup>A</sup>T<sub>E</sub>X Template*.

© 2021 – <http://github.com/egraff/uit-thesis>

# Abstract

Screening tests are vital for detecting diseases, especially at early stages, where efforts can prevent further illness. For example, osteoporosis is a systemic skeletal disease characterized by low bone mass and microarchitectural deterioration of bone tissue, resulting in bone fragility and susceptibility to fracture. Dual-energy x-ray absorptiometry is commonly used to diagnose osteoporosis since it evaluates bone mineral density. It is the most standard method for diagnosing osteoporosis, but it is not immediately available and is commonly used for research due to the high capital cost. Further, dual-energy x-ray absorptiometry is not used for populational-based screening due to its suboptimal ability to predict hip fractures based on measurements. Therefore, it is recommended to adopt a case-finding strategy to identify individuals at risk who benefit from the dual-energy x-ray absorptiometry examination.

Several indices have been developed to estimate bone quality in dental panoramic radiographs to identify individuals at risk of osteoporosis. In particular, the mandibular cortical width index. Studies suggest that dentists can measure the mandibular cortical width to identify individuals at risk and refer them for bone mineral density testing. However, this endeavor is time-consuming and inconsistent due to the bone's unclear borders and the challenge of determining the mental foramen's position, leading to varying measurements between clinicians. Therefore, the dentistry community is investigating how to automate this process effectively and accurately.

In an attempt to address some of these problems, this thesis presents a method to assess the mandibular cortical width index automatically. Four different object detectors were analyzed to determine the mental foramen's position. EfficientDet showed the highest average precision (0.30). Therefore, it was combined with an iterative procedure to estimate mandibular cortical width. The results are promising.





# Acknowledgments

Big thanks to my helpful supervisors, Fred Godtliebsen, Thomas Haugland Johansen, Jonas Nordhaug Myhre, Napat Limchaichana Bolstad, and Anna Teterina, for overseeing and guiding me through this thesis. Extra thanks to Napat and Anna for the immense work you have done.

Lastly, I would like to express my sincerest gratitude to my family and friends for being the most reliable supporters.

Kom arbeidslyst og treng deg på. Her skal du motstand finne.  
*Flåklypa Grand Prix (1975)*

# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgments</b>	<b>iii</b>
<b>List of Figures</b>	<b>viii</b>
<b>List of Tables</b>	<b>x</b>
<b>List of Algorithms</b>	<b>x</b>
<b>List of Abbreviations</b>	<b>xi</b>
<b>I Introduction</b>	<b>1</b>
<b>1 Motivation and objective</b>	<b>3</b>
1.1 Thesis Outline . . . . .	5
1.2 Mathematical Nomenclature . . . . .	5
<b>II Background and Methodology</b>	<b>7</b>
<b>2 Dental Panoramic Radiographs</b>	<b>9</b>
2.1 The Mental Foramen . . . . .	10
2.2 Related work . . . . .	12
<b>3 Machine Learning</b>	<b>14</b>
3.1 Neural Network . . . . .	15
3.1.1 The Perceptron . . . . .	15
3.1.2 Feedforward Neural Networks . . . . .	16
3.2 Convolutional Neural Networks . . . . .	18
3.2.1 Convolution . . . . .	18
3.2.2 Pooling . . . . .	20
3.2.3 Convolutional layers . . . . .	20

3.2.4	Transposed convolution . . . . .	21
3.3	Learning the Parameters . . . . .	21
3.3.1	Optimization techniques . . . . .	23
3.4	Regularization . . . . .	24
3.4.1	Weight Regularization . . . . .	24
3.4.2	Batch Normalization . . . . .	25
3.4.3	Data Augmentation . . . . .	26
3.4.4	Dropout . . . . .	26
3.5	Performance Evaluation . . . . .	26
3.5.1	Accuracy . . . . .	27
3.5.2	Precision . . . . .	27
3.5.3	Recall . . . . .	28
3.5.4	Intersection-over-Union . . . . .	30
<b>4</b>	<b>Deep learning and Object Detection</b>	<b>32</b>
4.1	Pretrained Models and Backbones . . . . .	32
4.2	Object Detection . . . . .	33
4.2.1	Non-maximum Suppression . . . . .	34
4.2.2	Two-stage Detectors . . . . .	34
4.2.3	Single-stage Detectors . . . . .	36
4.2.4	RetinaNet . . . . .	37
4.2.5	CenterNet . . . . .	38
4.2.6	EfficientDet . . . . .	39
<b>5</b>	<b>Tromsø Survey 7 Data Set</b>	<b>42</b>
<b>III</b>	<b>Results and Discussion</b>	<b>44</b>
<b>6</b>	<b>Experiments and Results</b>	<b>46</b>
6.1	Experimental Setup . . . . .	47
6.1.1	Faster R-CNN . . . . .	47
6.1.2	RetinaNet . . . . .	48
6.1.3	CenterNet . . . . .	48
6.1.4	EfficientDet-D0 . . . . .	48
6.1.5	Procedure to Estimate MCW . . . . .	49
6.2	Results . . . . .	51
6.2.1	Detecting the MF . . . . .	51
6.2.2	Estimating MCW . . . . .	63
<b>IV</b>	<b>Conclusion</b>	<b>65</b>
<b>7</b>	<b>Conclusion and Final Thoughts</b>	<b>67</b>

CONTENTS	vii
7.1 Future work . . . . .	69
<b>Bibliography</b>	<b>70</b>
<b>V Appendix</b>	<b>77</b>
<b>Appendix</b>	<b>79</b>
7.2 Extra Figures . . . . .	80

# List of Figures

1.1	Proposed system . . . . .	4
2.1	Schematic view of dental panoramic radiography . . . . .	10
2.2	Dental panoramic radiograph region where essentials are marked	12
3.1	Feedforward Neural Network . . . . .	17
3.2	Convolution . . . . .	19
3.3	Figurative description of gradient decent . . . . .	22
3.4	Conceptual sketch of the dropout strategy . . . . .	26
3.5	ROC curve . . . . .	29
3.6	PR curve . . . . .	30
3.7	Figurative description of IoU. . . . .	31
4.1	R-CNN architecture . . . . .	35
4.2	Faster R-CNN architecture . . . . .	35
4.3	YOLO architecture . . . . .	37
4.4	FPN structures . . . . .	40
6.1	Predictions from EfficientDet D0 in complex scenarios . . . . .	53
6.2	Predictions from EfficientDet D0 in complex scenarios . . . . .	54
6.3	Predictions from EfficientDet D0 as datapoints evaluated by expert 1 . . . . .	55
6.4	Example of predictions with low IoU . . . . .	56
6.5	Predictions from EfficientDet D0 as data points evaluated by expert 1 . . . . .	57
6.6	Predictions from EfficientDet D0 as datapoints evaluated by expert 2 . . . . .	59
6.7	Crops showing insufficient size for evaluation of the MF's position. . . . .	60
6.8	Incorrect predictions from EfficientDet D0 on highly complex scenarios . . . . .	62
6.9	Incorrect predictions from EfficientDet D0 on highly complex scenarios . . . . .	62
6.10	Example results for the automatic estimation of MCW. . . . .	63

6.11	Figure depicts two cases where the measuring algorithm need improvements. . . . .	64
7.1	Predictions from EfficientDet D0 in easy scenarios . . . . .	80
7.2	Predictions from EfficientDet D0 in easy scenarios . . . . .	81
7.3	Predictions from EfficientDet D0 as datapoints evaluated by expert 2 . . . . .	82

# List of Tables

1.1	Mathematical nomenclature . . . . .	6
6.1	Results for experimental set-up 1 . . . . .	51
6.2	Results for experimental set-up 2 . . . . .	52
6.3	Evaluation of 101 complex images by two experts . . . . .	61
6.4	Combined evaluation of 101 complex images . . . . .	61

# List of Algorithms

1	Measuring mandibular cortical width algorithm . . . . .	50
---	---	----



# Abbreviations

**AI** Artificial intelligence

**BMD** Bone mineral density

**CNN** Convolutional neural network

**DPR** Dental panoramic radiograph

**e.g.** *exempli gratia* (latin for "for example")

**FCN** Fully connected network

**FPN** Feature pyramid network

**i.e.** *id est* (latin for "that is")

**IoU** Intersection-over-Union

**MCW** Mandibular cortical width

**MF** Mentale foramen

**MLP** Multilayer perceptron

**NMS** Non-maximum suppression

**NN** Neural network

**ReLU** Rectified Linear Unit

**RPN** Region proposal network

## **Part I**

# **Introduction**





# Motivation and objective

Dental panoramic radiography (DPR) is a valuable diagnostic tool in dental practice and has long been one of the most standard means for dental imaging by dentists and oral surgeons due to its many advantages. For example, annually it is taken approximately 16 million dental panoramic radiographs in the general dental service in England and Wales (Rushton and Horner, 1996), 10 million are taken in Japan (Taguchi et al., 2006), and 5.55 million are taken in Norway (Levernes et al., 2014).

DPRs provide invaluable information about oral health and are used in various applications, including diagnosis of caries, periodontitis, and periapical pathologies. In addition, studies suggest oral health and general health mutually affect each other; research found a significant association between osteoporosis and alveolar bone loss, periodontal diseases, and tooth loss (Bernal et al., 2018).

The use of AI in the medical field has increased in recent years (Patel et al., 2009). However, AI solutions have not yet entered routine dental practice extensively (Schwendicke et al., 2020), despite the fact that dentistry is suited for AI;

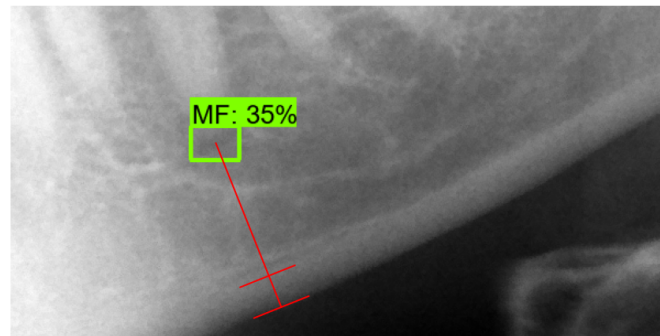
- Imagery is the foundation of most patients' dental voyage, from screening to treatment.
- Dentistry frequently applies different imagery procedures from the same anatomical region of the same individual. Further, data are often collected

over multiple periods.

- Common dental conditions are relatively prevalent. Therefore, creating datasets with multiple “affected” cases can be accomplished with little effort.

This thesis provides a system for identifying the mental foramen using state of the art object detection tools. In addition, we present an automatic system for estimating the mandibular cortical width based on previously located mental foramen (see figure 1.1). Specifically, four different deep learning models for object detection will be tested. Tromsø Survey 7 provides the raw data examined in this thesis: the dental panoramic radiographs. In addition, with the help of clinical staff, a dataset with bounding box annotations suitable for object detection has been created.

The dental health care community provides services to the public, and ongoing research seeks practical and precise methods to improve today’s systems. This thesis is assigned to study the possibilities in DPRs and may contribute to systems being adopted as part of Tromsø’s dental health services.



**Figure 1.1:** Proposed system for automatic measurements of MCW.

In addition to aiding the estimation of mandibular cortical width, finding the position of the mental foramen is important in several clinical settings. Knowing its anatomy and anatomical variations is cardinal to clinicians to minimize complications related to nerve procedures (Laher et al., 2016). In addition, the mental foramen is used as a landmark when measuring mandibular cortical width (MCW), and several studies suggested that the MCW might be helpful for osteoporosis triage screening (Kinalski et al., 2020; Calciolari et al., 2015). However, as Devlin et al. pointed out, measuring the MCW is time-consuming and not precise when performed manually by dentists. Therefore, automatic measurements of MCW are a new pathway to osteoporosis triage screening (Devlin et al., 2007). Therefore, the first step in developing such a tool for automatic measurement would be automatic detection of the mental foramen.

To summarize, the objective of this thesis are:

1. Exploring the feasibility of detecting the mental foramen in DPRs with pre-trained object detection models.
2. Explore the possibilities for an automatic measurement tool of mandibular cortical width.

## 1.1 Thesis Outline

**Part I** expresses the motivation behind object detection in oral medicine and thesis formalities.

**Part II** provides fundamental theory and concepts of dental panoramic images, anatomy, and machine learning. A brief review of related work and relevant theory regarding modern object detection is also given. In addition, we present properties of the data set concerning how it is collected and created and the architectures chosen for testing.

**Part III** describes the experiments conducted involving how the architectures were set up; results are presented and discussed.

**Part IV** concludes the thesis, considering the hypothesis and experimental results. In addition, views on future work are given.

## 1.2 Mathematical Nomenclature

There is a wide variation in the mathematical notation in physics, statistics, and mathematics, which all contribute to this thesis. In Table 1.1, the notation in this thesis is outlined.

**Table 1.1:** Table of mathematical terminology used in this thesis.

<b>Numbers and Arrays</b>	
$a$	A scalar
$\mathbf{a}$	A vector
$\mathbf{A}$	A matrix
$\mathbf{o}$	A vector having all elements equal to zero
<b>Sets and Indexing</b>	
$\mathbb{R}$	The set of real numbers
$\mathbb{N}$	The set of natural numbers. $\mathbb{N} = \mathbb{N}_1 = \{1, 2, \dots\} \subset \mathbb{R}$
$\mathbf{a}_i$	Element $i$ of vector $\mathbf{a}$ , with indexing starting at 1
$A_{i,j}$	Element in matrix $A$ at row $i$ and column $j$
$\{0, 1, \dots, n\}$	Set containing all integers between 0 and $n$
$\{\mathbf{x}^{(i)}\}_{i=1}^N$	Set containing elements of $\mathbf{x}$ with index between 1 and $N$
$(a, b]$	Real interval excluding $a$ but including $b$
<b>Calculus and Linear Algebra</b>	
$\nabla_{\mathbf{x}} y$	Gradient of $y$ with respect to $\mathbf{x}$
$\sum_{i=1}^n a_i$	Sum of elements in $\mathbf{a}$ having index between 1 and $n$
$\sum_i a_i$	Sum of all valid elements in $\mathbf{a}$
$\ \cdot\ $	Euclidean distance
$\mathbf{A} \odot \mathbf{B}$	Element-wise (Hadamard) product of $\mathbf{A}$ and $\mathbf{B}$
$\mathcal{O}$	Order of a function. $\mathcal{O}(n^2)$ is quadratic order
<b>Functions and Statistical Theory</b>	
$\log(x)$	Natural logarithm of $x$
$\exp(x) = e^x$	Exponential of $x$
$\sigma(x)$	Logistic sigmoid. $\sigma(x) = \frac{1}{1+\exp(-x)}$
$\circ$	Function composition. $(g \circ f)(x) = g(f(x))$
$\%$	Modulus. $a \% b = a - \left[\frac{a}{b}\right] * b$
$*$	Convolution. The discrete convolution operator is defined in equation 3.7
$A \cap B$	Union of $A$ and $B$ . Sum of all elements in set $A$ and set $B$
$A \cup B$	Intersection of $A$ and $B$ . Elements in set $A$ also included in set $B$
$\mathbb{E}_{X \sim p}[\cdot]$	Expectation with respect to a stochastic variable $X$ from a distribution $p$
<b>Data Sets</b>	
$\mathbf{x}^{(i)}$	$i$ -th example (sample) from data set
$\mathbf{y}^{(i)}$	Target (label) associated with $\mathbf{x}^{(i)}$ in supervised learning
$\mathbb{D}$	Set containing the complete training data
$\mathbb{B} \subseteq \mathbb{D}$	Subset of the complete training data set. A batch

## **Part II**

# **Background and Methodology**







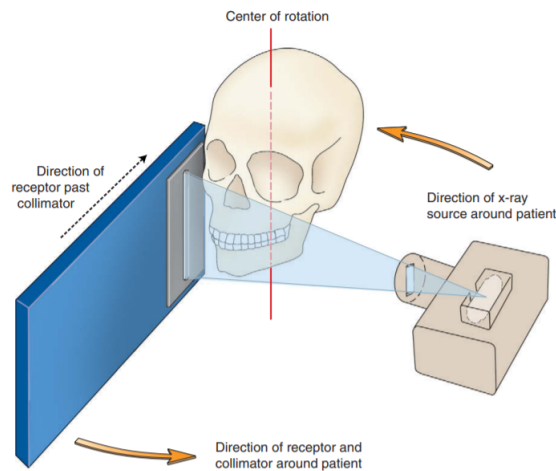
# Dental Panoramic Radiographs

Dental panoramic radiography is a process for capturing multiple images and combining them into a single image of facial structures. Dental panoramic radiographs (DPRs) address a comprehensive view of the jaw. In many situations, DPRs assist in providing information on the jaw's status prior to further examination decisions. For example, evaluation of trauma including jaw fractures, location of third molars, extensive dental or osseous disease, tooth development and eruption, and developmental anomalies (White and Pharoah, 2014).

Panoramic imaging is usually used as the initial evaluation image to provide critical insight or determine the need for other projections (White and Pharoah, 2014). There are several advantages of using panoramic imaging. First, it is a quick and convenient radiographic technique exposing a low radiation dose. It gives insight into the overall evaluation of dentition, dentomaxillofacial<sup>1</sup> trauma, and developmental disturbances. DPRs are prudent, yet there are drawbacks. Compared to intraoral radiographs, the resolution is lower and fine details are not provided. Producing a DPR requires accurate patient positioning to avoid errors and artifacts (ghost images). Finally, size distortion across the image is irregular, making linear measurements unreliable unless the patient is positioned correctly. The latter is also referred to as magnification and leads

1. relating to the jaw and face.

to image distortion. Magnification is influenced by the angulating radiation beam aligning the horizontal plane, x-ray source-to-object distance, and the position of objects inside the focus area. Thus, a patient's position during an x-ray examination can alter the magnification (Paasche Edvardsen, 2021).



**Figure 2.1:** As the x-ray tube head moves around one side of the patient, the receptor assembly moves on the opposite side, figure adapted from White and Pharoah (2014).

Ghost images and double images appearing over the resulting DPR are commonly seen from this technique (figure 2.1) because anatomical structures are projected twice on the receptor as the x-ray beam moves around the patient's head. Therefore, anatomical structures overlapped by a ghost image can not be distinguished by their edges.

## 2.1 The Mental Foramen

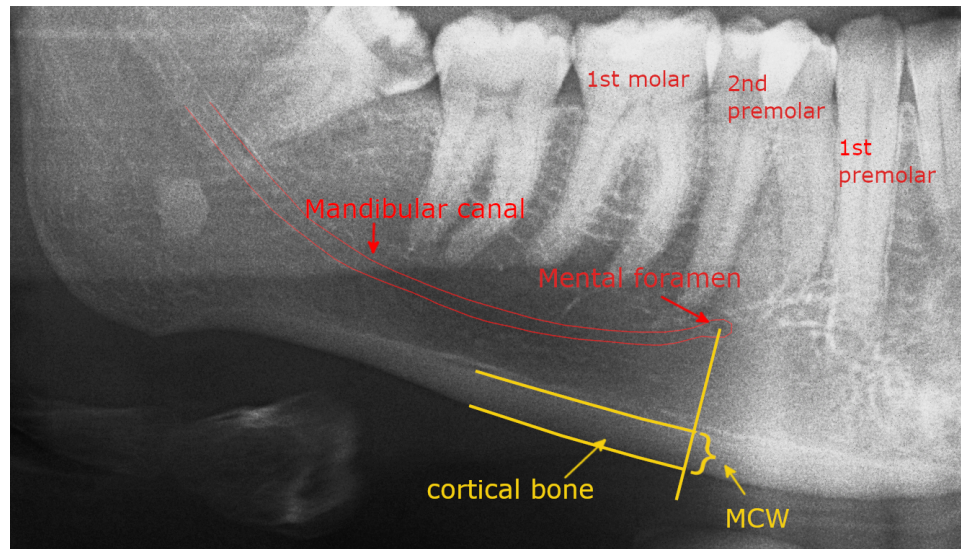
The mental foramen (MF) is a clinically significant landmark for clinicians from several disciplines, such as dentists, oral maxillofacial surgeons, emergency physicians, and plastic and reconstructive surgeons (Laher et al., 2016). For example, to perform a mental nerve block, a type of anesthesia placed in the MF's region, the accurate determination of the mental MF's position is paramount. Otherwise, nerve and blood vessels can be injured.

The mental foramen is most commonly located in the projection of the root apex of the second premolar (see figure 2.2) or between the first and second premolar apex. However, irregular tooth alignments or missing pieces make it more challenging to determine the MF's location (Hasan, 2012). The existence

of a single MF is the most familiar. However, variations like supernumerary (accessory MF), curling, looping or missing MF are also met. Accessory MF can occur because the mental nerve splits into several nerve fibers before the development of MF, resulting in double, triple, or quadruple mental foramina. However, accessory MF is a more common phenomenon than absence (Hasan, 2012). The accessory mental foramen is present in approximately 1-6% of cases in different populations, and it was found that mental foramen is clearly visible only on 50-65% of DPRs, while it was detectable in approximately 87-95% of DPRs (Greenstein and Tarnow, 2006). Jacobs et al. reported detection of the MF in 94% of 545 DPRs. However only 49% were considered clearly visible by two independent observers (oral radiologist).

In this study, we are interested in the cortical width below the mental foramen, subsequently termed the mental index (MI) or mandibular cortical width (MCW). Ledgerton et al. (1999) first exhibited the possibility of MCW as a beneficial screening tool in identifying postmenopausal women with undetected low skeletal bone mineral density BMD or osteoporosis.

In later years, osteoporosis studies have been conducted using different DPRs indices and have been defined and validated by the scientific dentistry community (Taguchi et al., 1996; Vlasidis et al., 2007; López López et al., 2011; Passos et al., 2012; Hastar et al., 2011; Parlani et al., 2014; Calciolari et al., 2015; Tofangchiha et al., 2017; Chandak et al., 2017; on Osteoporosis et al., 2001).



**Figure 2.2:** Visualization of a DPR region with essential markings such as the mental foramen, mandibular canal, and cortical bone. It is shown that MCW is measured between the bone's border along the line drawn through the MF perpendicular to the tangent of the lower edge of the bone. Image adapted from: [https://commons.wikimedia.org/wiki/File:Panoramic\\_radiograph\\_-\\_Orthopantomogram.jpg](https://commons.wikimedia.org/wiki/File:Panoramic_radiograph_-_Orthopantomogram.jpg)

## 2.2 Related work

Studies on automatic image analysis from DPRs have been conducted in recent years, and it is well known that it represents a challenge due to the inherent complexity of DPRs. The challenge implicates identifying and recognizing specific structures and their morphometry. The latter typically involves MCW, PMI (Benson et al., 1991), or M/M (Calciolari et al., 2015) indices. In this section, some studies are presented regarding their proposed methods and disadvantages. The studies give a perspective on the challenge and ongoing research.

Before considering an automatic system, Arifin et al. created a manual computer-aided system for measuring MCW based on gradient analysis of edges in 2006. In addition, high pass filtering was utilized, which is very noise sensitive depending on a set threshold. As the dentists had to determine the MF's position manually, Arifin et al. believed the experience of the examiners may greatly influence their decision, resulting in poor intra- and interexaminer agreement.

Other studies (Abdi et al., 2015; Kavitha et al., 2013; Naik et al., 2016) have focused on automatic segmentation of the mandible. The approaches involved, for example, horizontal integral projections, a modified Canny edge detector,

morphological operations, thresholding, and active contour models. Methods relying on the isolation of the cortical bone region are prone to obstacles due to the bone's irregular shape. Active contour models, or snakes (Cootes et al., 1995), require a clear distinction of image intensity levels so that the snakes can follow the border of the mandible.

Aliaga et al. has considered all these factors when developing an automatic system for computing mandibular indices in DPRs. The resulting algorithm computed indices inside two regions of interest that tolerate flexibility in sizes and locations, making this process robust enough. However, as a part of locating the MF, they use morphological operations and report that the proposed approach fails in 5% of 310 cases to detect the MF.

Lee et al. used transfer learning for screening of osteoporosis in DPRs, with a limited dataset (680 images). The highest overall accuracy achieved was reported 84%. Their results showed that transfer learning with pre-trained weights and fine-tuning techniques could be helpful and reliable in the automated screening of osteoporosis patients.

In this thesis, we are inspired by the works of Aliaga et al. and Lee et al..

# /3

## Machine Learning

Machine learning is a field of study that has been around for decades and continues to evolve around mathematics, statistics, and computer science. It Machine learning commonly involves classification, prediction, segmentation, and decision-making problems. Machine learning is considered a subfield of artificial intelligence (AI). In recent years remarkable results across applications have been shown (Alpaydin, 2014). The expansive access to data has been a significant part of machine learning's success. In addition, immense machine learning models, particularly deep learning models<sup>1</sup> consisting of tens of millions of parameters, have shown outstanding results due to the significant developments of graphical processing units (GPU), that significantly reduce the training time of such models (Shi et al., 2016).

The key idea behind machine learning is the ability to solve tasks without explicitly designing a rule-based system to do so. Instead, machine learning resolves an assignment by learning from data and adapting to the present task. Hence, the data is often referred to as training data and is essential for machine learning to function. Concerning the data available, machine learning methods are usually grouped into four main categories, namely supervised, semi-supervised, unsupervised, and reinforcement learning. These categories define how a machine learning system, or model, learns the data. Regardless, the data supplied to the model is processed to give an output, then a loss function will evaluate the output to quantify the error made, hence evaluating

1. Deep learning refers to neural networks with numerous layers.

how sound the output was. With this error, the model can compute possible corrections, e.g., if the output was sound, little or no corrections are needed; if the output was terrible, the model would adjust itself to improve the output. However, this assumes the data consists of pairs of input and outputs so that the loss function can quantify the error, which brings us back to the four categories of training data:

1. **Supervised learning:** the data is labeled, i.e., consist of input and output pairs. For each input example, a desired output has been provided, commonly called ground truth or label—for example, a dog's picture is the input and the label "dog" is the output. The loss function compares the model's prediction and label. Hence, the model trains to reproduce the correct output for each input.
2. **Unsupervised learning:** the training data is unlabeled; hence only input examples exist.
3. **Semi-supervised:** the data consist of a mix of the two previously mentioned cases, i.e., some labels are missing.
4. **Reinforcement learning:** the model acts as an agent; hence the inputs are surrounding information, and the outputs are actions.

We point out that this thesis will employ supervised learning.

## 3.1 Neural Network

At the core of deep learning are neural networks, also known as multilayer perceptrons MLP. An introduction to neural networks begins with describing its basic building block, the perceptron, before proceeding to multilayer perceptrons and convolutional neural networks.

### 3.1.1 The Perceptron

The human brain is a complex information-processing machine composed of  $10^{11}$  processing units, namely neurons (Alpaydin, 2014). The perceptron (Rosenblatt, 1958) is a simplified model of a neuron, or rather, the synaptic connection between neurons. The perceptron maps an input  $\mathbf{x}$  to an output  $y$  by performing an inner product between the inputs and weights  $\mathbf{w}$ , and adding a bias parameter  $b$ , resulting in a *potential* (equation 3.1). Consequently, the potential composes a simple linear model for vectorial data. An activation



function,  $g(\cdot)$ , evaluates the potential, and the output is hence called the activation. The activation function is usually a non-linear function.

$$\mathbf{w}^T \mathbf{x} + b = y \quad (3.1)$$

where  $\mathbf{w}$  are the learnable weight parameters reflecting influence of neurons' synaptic connection, and  $b$  reflects the threshold deciding if the collective influence of inputs is sufficient to make the neuron fire.

Different activation functions can be used to manipulate the potential for different purposes. For example, the output can be interpreted as a pseudo-probability when the output falls between zero and one, which is achieved using a sigmoid function (equation 3.2). Other popular activation functions are the rectified linear unit (ReLU) (Nair and Hinton, 2010) (equation 3.4) and the tanh activation (equation 3.3). ReLU is widely used in modern neural networks due to its simplicity and its mathematical convenience. Tanh is similar to the sigmoid. However, it restricts all output values to the range  $[-1, 1]$ .

$$g_{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (3.2)$$

$$g_{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.3)$$

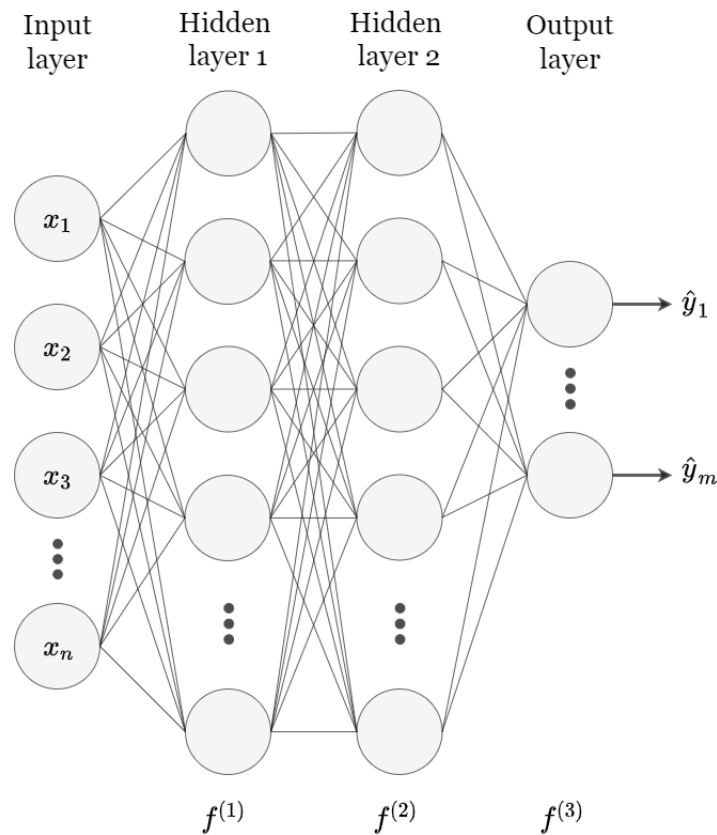
$$g_{ReLU}(x) = \max\{0, x\} \quad (3.4)$$

A single perceptron has limitations, as it cannot be used for non-linear regression. However, a MLP can be constructed to overcome this limitation. As the name suggests, a MLP is several stacked perceptrons (nodes or neurons), often called feedforward neural networks (NN).

### 3.1.2 Feedforward Neural Networks

The goal of a feedforward neural network is to approximate some function  $f^*$  (Goodfellow et al., 2016). Using a mapping  $\mathbf{y} = f(\mathbf{x}; \boldsymbol{\theta})$ , it attempts to

accomplish the goal by learning the value of the parameters  $\theta$  that best approximate the function  $f^*$  (learning will become clear in section 3.3). Accordingly, information flows through the function from the input  $x$  to the output  $y$  via the intermediate computations used to define  $f$ . Feedforward neural networks are called networks because they are typically represented by composing together many different functions (Goodfellow et al., 2016). The functions are most commonly structured in chains, with an input layer, hidden layers, and a final output layer. The hidden layers are between the input and output layer. Data represented as a vector  $x^{(0)}$ , with  $n$  features, are passed to the input layer. Following the input layer, every node in a layer receives the output of all nodes in the previous layer (see figure 3.1).



**Figure 3.1:** Visualization of a feedforward neural net with  $n$  inputs, 2 hidden layers and  $m$  outputs.

A general feedforward neural network can mathematically be described by equation 3.5.

$$f(x; \theta) = f^{(L)} \circ f^{(L-1)} \circ \dots \circ f^{(1)}(x) \quad (3.5)$$

Since each layer  $f^{(l)}$  consists of  $k_l$  nodes with  $n$  weights  $\mathbf{w}$ , and a bias term  $b$ , these parameters can be collected in a weight matrix expressing weights between all neurons in neighboring layers and a bias vector expressing all biases in a layer. Hence all trainable parameters (that can be learned) in a layer are collected in  $\theta^{(l)} = \{\mathbf{W}^{(l)}, \mathbf{b}^{(l)}\}$ .

Specifically,  $\mathbf{W}^{(l)} \in \mathbb{R}^{k_l \times k_{l-1}}$  denotes the weight matrix concerning layer  $f^{(l)}$ . The number of neurons in layer  $f^{(l)}$  and  $f^{(l-1)}$  are denoted  $k_l$  and  $k_{l-1}$ , respectively. Every node evaluates the potential in equation 3.1 with an activation function  $g(\cdot)$ , in parallel in every layer:

$$f^{(l)}(\mathbf{x}^{(l-1)}; \theta^{(l)}) = g(\mathbf{W}^{(l)} \mathbf{x}^{(l-1)} + \mathbf{b}^{(l)}), \quad l = 1, \dots, L \quad (3.6)$$

The output layer produces a vector  $\hat{\mathbf{y}}$  with  $m$  outputs (activations), which is the resulting of the network, i.e.,  $\hat{\mathbf{y}} \approx f^*(\mathbf{x})$ . The number of layers  $L$  is referred to as the *depth* of the network, and the dimensionality of the hidden layers determines the *width* of the network.

## 3.2 Convolutional Neural Networks

A convolutional neural network (CNN) (LeCun et al., 1989) is designed to process grid-like data structures, e.g., images which can be considered as a 2-D grid of pixels. Convolutional networks have been remarkably successful in practical applications Goodfellow et al. (2016), therefore convolutional neural networks will be described thoroughly before introducing additional machine learning methods.

### 3.2.1 Convolution

"Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers" (Goodfellow et al., 2016). Consequently, an understanding of the convolutional operation is vital.

Let the input array  $I(x, y)$  be an image, and  $K(x, y)$  be the convolution *filter*<sup>2</sup>.

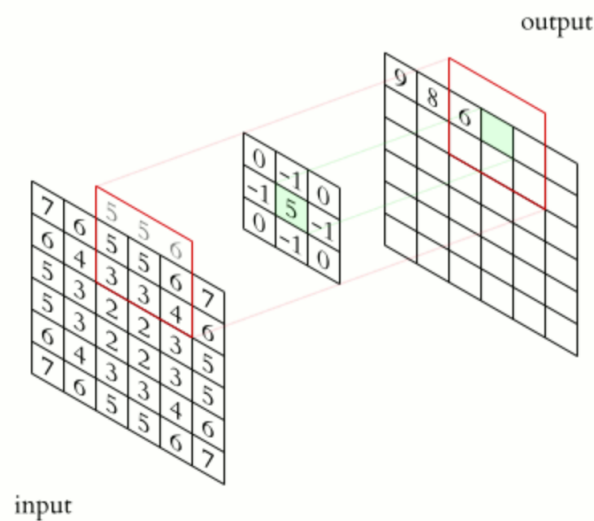
2. Frequently also termed Kernel, which we will avoid as it can be confused with distinct types of "kernels."

The two-dimensional discrete convolution is defined as:

$$(I * K)(x, y) = \sum_m \sum_n I(x - m, y - n)K(m, n) \quad (3.7)$$

Summation limits are omitted because they depend on the *type* of convolution executed. That is, *valid*, *same* or *full*, which decides the output dimension (see (Goodfellow et al., 2016, p. 337-345)).

The convolution operation can be viewed as sliding a filter or over an input image (see figure 3.2) and at each location producing an output sum of the product of a set of weight parameters (elements of the filter) and pixels (elements of the input) contained in a neighborhood. The output is a set of linear activations commonly called *feature map* or *activation map*, and the convolution operator can obtain specific features such as edges by applying a suitable filter. Stride is the number of spatial increments the filter is moved. Motivation for using strides greater than one is data reduction, which is a substitute for subsampling.



**Figure 3.2:** Visualization of a *full* two-dimensional discrete convolution of an input and a filter (in the middle), to produce an output. Image adapted from [https://commons.wikimedia.org/wiki/File:2D\\_Convolution\\_Animation.gif](https://commons.wikimedia.org/wiki/File:2D_Convolution_Animation.gif).

The motivation behind convolution is three essential ideas that can benefit machine learning systems: sparse interactions, parameter sharing, and equivariant representations.

Sparse interactions imply that filters of few parameters can obtain meaningful

features. Parameter sharing implies that the weights of the filter are used at multiple input locations, thus being shared across the spatial dimensions of the image. Consequently, the memory occupied by a CNN may be significantly less than that of an NN of comparable size. Finally, equivariance to translation is a property that allows the output to change in the same way as the input (if it changes).

### 3.2.2 Pooling

Pooling is a fundamental operation in almost every convolutional neural network and is typically executed after the activation function is applied to the set of linear activations (feature map) from the parallel convolutions. Pooling replaces the outputs at a specific location with a summary statistic of the nearby outputs. The motivation is to intensify the presence of features in the feature map. For example, average pooling and max pooling operations report the average and the maximum output within a rectangular neighborhood, respectively. In addition, pooling assists in making the representation roughly invariant to translation.

### 3.2.3 Convolutional layers

It was stated that convolutional networks are neural networks (equation 3.6) that use convolution instead of general matrix multiplication in a layer. Doing so unveils equation 3.8:

$$f^{(l)}(\mathbf{x}^{(l-1)}; \boldsymbol{\theta}^{(l)}) = g(\mathbf{K}^{(l)} * \mathbf{X}^{(l-1)} + \mathbf{b}^{(l)}) \quad (3.8)$$

Distinct from equation 3.6 is that we introduced the convolution filter  $\mathbf{K}^{(l)}$ . Hence,  $\boldsymbol{\theta}^{(l)} = \{\mathbf{K}^{(l)}, \mathbf{b}^{(l)}\}$ . Nevertheless,  $\mathbf{b}$  remains the bias vector, and  $\mathbf{X}^{(l-1)}$  is still the output of previous layers.

If the initial input array at  $l = 1$  is two-dimensional data (height, width), then  $\mathbf{X}^{(l-1)} = \mathbf{X}^{(0)} \in \mathbb{R}^{h \times w}$ , and  $\mathbf{b}^{(l)} \in \mathbb{R}^{k_l}$ , where  $k_l$  is a hyperparameter denoting the number of filters in the layer  $l$ . It is noted that the case differs slightly for multidimensional inputs, i.e., color channels, 3-D images, and batches of inputs. The reader is referred to (Goodfellow et al., 2016, p. 347-358) for elaborated details.

### 3.2.4 Transposed convolution

Transposed convolution (also called fractionally-strided convolution, deconvolution, or learned upsampling) is a version of the convolution operation. It is needed to train CNNs (that have more than one layer). The motivation behind transposed convolution is to return to the input from a summarized input. That is, moving from a low dimension to a higher one. Therefore, transposed convolution is used for upsampling and involves an unraveled (summarized) input matrix and a transposed matrix  $Y^T$ , where  $Y = a(K)$  and  $K$  is the convolution filter. Details of training a CNN is beyond the scope of this thesis, but can be found in (Goodfellow et al., 2016, p. 345-350)

## 3.3 Learning the Parameters

This chapter stated early that machine learning models could learn parameters from data, and many parameters have been introduced until now; therefore, the learning process will be explained here. Note that a hyperparameter is not a learnable parameter; the user explicitly defines the hyperparameter.

Feedforward neural networks approximate an ideal function, that is, finding the best solution from all feasible solutions. Hence, an optimization problem. The latter is solved by an iterative method, a mathematical procedure that uses an initial value to generate a sequence of improving approximate solutions. The procedure is called training in a machine learning setting. Training entails improving all parameters  $\theta$  in the network to *minimize* a loss function  $\mathcal{L}(\theta)$ , evaluated on the data  $\mathcal{X} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^N$  (supervised setting). The reader is reminded that a loss function quantifies the *difference* between the predicted output  $\hat{\mathbf{y}} = f(\mathbf{x}; \theta)$  and the ground truth  $\mathbf{y}$ . The difference measure is called *the loss*, and minimizing any function is to find its global minimum concerning the input (the parameters):

$$\theta^* = \arg \min_{\theta} J(\theta) \quad (3.9)$$

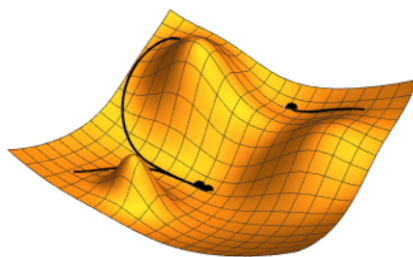
here we use  $J(\theta)$ , which is an *objective function* (a more general function, not necessarily dependent on labels)

In attempt to solve equation 3.9, the backpropagation algorithm (Rumelhart et al., 1986) is usually employed. In essence, it performs gradient decent with respect to the loss function (see equation 3.10). An appropriate loss function must be defined for the particular task that the model is intended to solve.

$$\theta_{new} = \theta_{old} - \mu \cdot \nabla_{\theta} \mathcal{L}(\theta; \mathcal{X}) \quad (3.10)$$

Here, the present parameters  $\theta$  is updated, and  $\mu$  is a hyperparameter called learning rate.

Gradient descent can be compared to walking down a mountain (see figure 3.3) with fog hanging. Then one must measure steepness (time-consuming) and walk (hopefully downhill) a distance before remeasuring. The path taken down reflects the sequence of parameters the algorithm will explore. The hill's steepness represents the loss. The derivative reflects measuring steepness. Finally, the walked distance (*steps*) reflects the learning rate.



**Figure 3.3:** Visualization of gradient decent. Image adapted from [https://commons.wikimedia.org/wiki/File:Gradient\\_descent.gif](https://commons.wikimedia.org/wiki/File:Gradient_descent.gif).

In equation 3.10, the parameters are updated by computing the gradient for the entire dataset  $\mathcal{X} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^N$ , which is computationally expensive for extensive datasets. Stochastic gradient descent (Robbins and Monro, 1951) resolves the issue by randomly partitioning the dataset into  $N$  batches  $\mathcal{B}$ , i.e.,  $\{\mathcal{B}_{(i)}\}_{i=1}^N = \mathcal{X}$ , and computing each batch's gradient. Equation 3.10 becomes:

$$\theta_{new} = \theta_{old} - \mu \cdot \nabla_{\theta} \mathcal{L}(\theta; \mathcal{B}_i) \quad (3.11)$$

where  $\theta_{old}$  originates from  $\mu \cdot \nabla_{\theta} \mathcal{L}(\theta; \mathcal{B}_{i-1})$ . Extensive datasets are more common today; hence stochastic gradient descent is more common.

From the loss surface in figure 3.3, we see that 2 out of 3 cases converged to a the global minimum and 1 got stuck in a local minimum. The situation is highly dependent on the learning rate. If the learning rate is too low, unnecessary training time is needed to localize a minimum. The optimizer also risks getting stuck in a local minimum, or the training time is up before reaching the minimum. On the other hand, if the learning rate is too high, the optimizer may oscillate around the global minimum, preventing convergence.

### 3.3.1 Optimization techniques

Gradient descent was briefly explained because today's optimization algorithms are motivated by gradient descent and stochastic gradient descent. Example of today's optimization algorithms includes; adaptive moment estimation (Adam) (Kingma and Ba, 2014), Nesterov accelerated gradient descent (NAG) (Nesterov, 1983), momentum stochastic gradient descent (Qian, 1999a) commonly just called *momentum*, and root mean squared propagation (RMSprop) (Hinton et al., 2012a). These algorithms try to regulate the learning rate to efficiently find a global minimum.

Momentum builds on top of stochastic gradient descent by including a *momentum term*  $\gamma$  (hyperparameter). Equation 3.11 becomes:

$$\begin{aligned} v_{new} &= \gamma v_{old} + \mu \cdot \nabla_{\theta} \mathcal{L}(\theta) \\ \theta_{new} &= \theta - v_{new} \end{aligned} \quad (3.12)$$

here,  $v_{old}$  is the previous momenta that originates from  $\mu \cdot \nabla_{\theta} \mathcal{L}(\theta_{old})$ . The goal is to accelerate stochastic gradient descent in the right direction

The Adam optimizer takes the concept further and monitors (exponential moving) averages of the gradient (now denoted as  $\mathbf{m}$ ) and the square of the gradients (now denoted as  $\mathbf{v}$ ). From a statistical view,  $\mathbf{m}$  and  $\mathbf{v}$  are called the first moment and uncentered second moment. They are:

$$\begin{aligned} \mathbf{m}_i &= \beta_1 \mathbf{m}_{i-1} + (1 - \beta_1) \nabla_{\theta_i} \mathcal{L}(\theta_i) \\ \mathbf{v}_i &= \beta_2 \mathbf{v}_{i-1} + (1 - \beta_2) (\nabla_{\theta_i} \mathcal{L}(\theta_i))^2 \end{aligned} \quad (3.13)$$

where  $\beta_1$  and  $\beta_2$  are hyperparameters that influence how fast the averages decay. Further,  $\mathbf{m}_i$  and  $\mathbf{v}_i$  are bias-corrected:

$$\tilde{\mathbf{m}}_i = \frac{\mathbf{m}_i}{1 - \beta_1^i} \quad \tilde{\mathbf{v}}_i = \frac{\mathbf{v}_i}{1 - \beta_2^i}$$

The resulting update rule is:

$$\theta_{i+1} = \theta_i - \mu \frac{\tilde{\mathbf{m}}_i}{\sqrt{\tilde{\mathbf{v}}_i} + \epsilon} \quad (3.14)$$

where  $\mu$  is the learning rate and  $\epsilon$  is a small hyperparameter to avoid numerical instabilities.

The advantage with Adam is that significant steps (order of  $\mu$ ) are taken when the gradients do not vary considerably, i.e.,  $\sqrt{\tilde{\mathbf{v}}_i} \approx \tilde{\mathbf{m}}_i$ . In addition, small steps



are taken when the gradients vary rapidly, i.e.,  $\sqrt{\tilde{v}_i} > \tilde{m}_i$ .

## 3.4 Regularization

An essential challenge for machine learning is to assemble models that make correct and precise decisions on unseen data, that is, a generalized model. After all, we do not want to spend time training a different model for the same problem. Therefore, strategies designed to reduce the test error can be included in the training to achieve a more generalized model. These strategies are called regularization. Overfitting a model is an action that decreases its generalizability. Essentially, the model has only learned the exact training data and not the distribution. An overfitted model is usually recognized when the training loss is very low, but test loss is significantly higher. Common regularization strategies avoiding overfitting employ batches, batch normalization (Ioffe and Szegedy, 2015), weight normalization, and dropout (Hinton et al., 2012b). Overfitting can be alleviated by collecting more training data, i.e., observed data. However, that can be costly, time-consuming, or not possible. So, for now, we assume that as much high-quality data is available and focus on regularization techniques.

### 3.4.1 Weight Regularization

Decades before the advent of deep learning, regularization was used for polynomial regression (Goodfellow et al., 2016). Weight regularization is a tool for adjusting function complexity to reduce overfitting. We assume the function  $f = 0$  is the simplest among all functions. Therefore, we can measure the complexity of a function by its distance from zero. How precisely to measure the distance is out of the scope of this thesis, but one way is to add a penalty term to the loss function (equation 3.15). So that, if the weight vector grows too large, the algorithm might minimize the weights norm contra minimize training error.

$$J(\theta) = \mathcal{L}(\theta) + \alpha\Omega(\theta) \quad (3.15)$$

where  $\alpha \in [0, \infty)$  is hyperparameter that weights the relative impact of the norm penalty term  $\Omega$ , relative to the standard loss function  $\mathcal{L}$ .

A measure of the parameters' size is needed (or a subset of  $\theta$ ), that is some norm. Now,  $\mathbf{w} \subseteq \theta$  denotes all weights that should be affected by the norm

penalty. Two common ways to calculate the size of  $\mathbf{w}$  are:

$$\Omega(\boldsymbol{\theta}) = \frac{1}{2} \|\mathbf{w}\|_2^2 \quad (3.16)$$

$$\Omega(\boldsymbol{\theta}) = \|\mathbf{w}\|_1 = \sum_i |w_i| \quad (3.17)$$

where  $\|\cdot\|_2$  is the Euclidean norm. Equation 3.16 is called  $L^2$  regularization (or *weight decay*) and equation 3.17 is called  $L^1$  regularization. From a statistical viewpoint,  $L^2$  regularization is known as *ridge* regression, and linear regression models regularized with  $L^1$  are known as *lasso* regression.

$L^2$  norm is commonly favored since it places an outsize penalty on large weight vector components so that the final model distributes weight evenly across more attributes. On the other hand,  $L^1$  penalties lead to models concentrating weights on a small set of attributes by clearing the other weights to zero, called feature selection (Wen et al., 2016).

### 3.4.2 Batch Normalization

Very deep models involve the composition of several functions (layers). The gradient tells how to update each parameter in a layer, assuming that other layers remain unchanged. However, all layers are updated simultaneously in practice, and unexpected changes happen. The authors (Santurkar et al.) argue that the changes are internal covariate shifts in the layers' learned distribution and concluded that batch normalization (often shortened as *batch norm*) alleviates the problem by parametrizing the underlying optimization, yielding improved optimization.

Batch norm is applying equation 3.18 to each layers input, where  $\hat{\mathbf{x}}^{(i)}$  is the batch normalized feature vector of  $\mathbf{x}^{(i)}$  and  $\mathbb{E}_{\mathcal{B}}[\cdot]$  and  $Var_{\mathcal{B}}[\cdot]$  denote the expectation and variance with respect to the set  $\mathcal{B}$ .

$$\hat{\mathbf{x}}^{(i)} = \frac{\mathbf{x}^{(i)} - \mathbb{E}_{\mathcal{B}}[\mathbf{x}^{(i)}]}{\sqrt{Var_{\mathcal{B}}[\mathbf{x}^{(i)}]}} \quad (3.18)$$

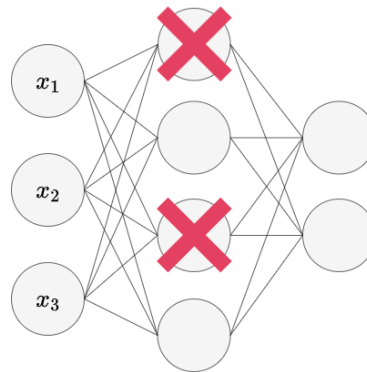
Batches were already mentioned in section 3.3. Because the batch given to the model varies each time, employing batches approximates optimization with the complete distribution (training data), including stochasticity. Therefore, employing batches is a regularization strategy, likely increasing the model's ability to generalize. Employing batches and batch normalization is considered sufficient strategies.

### 3.4.3 Data Augmentation

Another way to increase the model's generalization ability is to artificially generate augmented examples of the data. Examples of data augmentation typically done for image data are vertical and horizontal flipping, rotating, scaling, or adding noise (Goodfellow et al., 2016). Augmentation makes sense if the training data only captures one possible case, e.g., images of planes heading in one direction should be augmented because planes are generally heading in all directions (seen from the ground). However, it is not beneficial to train the model on augmentations that are not physically valid or not beneficial in the learning process.

### 3.4.4 Dropout

Finally, dropout is a simple idea that skips units in the network while training with a probability  $p$ . Dropout can be considered artificially implementing stochasticity to avoid overfitting during training. Figure 3.6 depicts the idea of applying the dropout as regularization. The motivation is to force units to learn and operate independently, not in the context of specific parallel units. During testing, all units are employed.



**Figure 3.4:** Conceptual sketch of how the dropout strategy, with  $p = 0.5$ , applies to a neural network.

## 3.5 Performance Evaluation

Until now, ways of making a model perform better have been described. However, suitable measures are required to estimate a model's performance.

The reader is reminded that a loss function computes the difference between

the desired output and the approximation. Further, the approximation error or total loss is the sum of losses for every individual instance. Therefore, one might assume that the lower the loss, the better the model's performance. The issue with this concept is that loss does not generalize across models, and loss is misleading if the model is overfitted. Furthermore, the loss is difficult to interpret, necessitating other performance metrics. Determining which performance metrics to employ depends on the goal. As the goal of this thesis includes supervised object detection, the luxury of comparing the output of a trained model with the ground truth is available. Thus, such performance metric is computed during testing. The following sections introduce standard performance metrics, some used primarily for object detection, others used for various applications. Nevertheless, a supervised setting is assumed.

### 3.5.1 Accuracy

Accuracy is a well-known excellent metric. For classification problems, accuracy is just the proportion of examples for which the model produces the correct output (Goodfellow et al., 2016). If the output is correct, it is called positive; otherwise, it is called negative. However, if the classes are heavily unbalanced, accuracy is misleading.

A part of the goal of this thesis is to evaluate whether the MF is present (positive) or not (negative) within regions of DPRs, that is, a two-class classification problem. Four outcomes are then possible; true-positive (TP), false-positive (FP), true-negative (TN), and false-negative (FN). TP and TN indicate that the model has *correctly* predicted the positive and negative classes, respectfully. On the other hand, FP and FN indicate that the model has *incorrectly* predicted the positive and negative classes, respectfully. False-positive and false-negative are frequently called Type 1 error and Type 2 error, respectively. Accuracy is defined in equation 3.19.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.19)$$

### 3.5.2 Precision

Precision is another metric that tries to answer the following; "what proportion of positive outputs was actually correct?". Thus, precision is the proportion of all predicted positives that were correctly classified, and high precision, i.e., high true positive rate, relates to the low false-positive rate. Precision is defined

in equation 3.20.

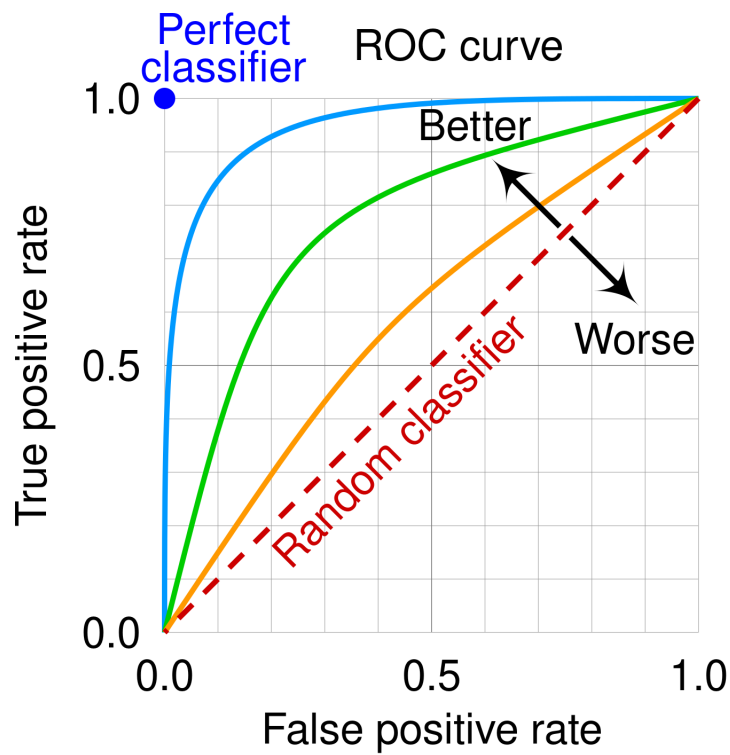
$$Precision = \frac{TP}{TP + FP} \quad (3.20)$$

### 3.5.3 Recall

Mathematically, recall is defined in equation 3.21. Recall, commonly called false-positive rate, answers what proportion of actual positives was identified correctly.

$$Precision = \frac{TP}{TP + FN} \quad (3.21)$$

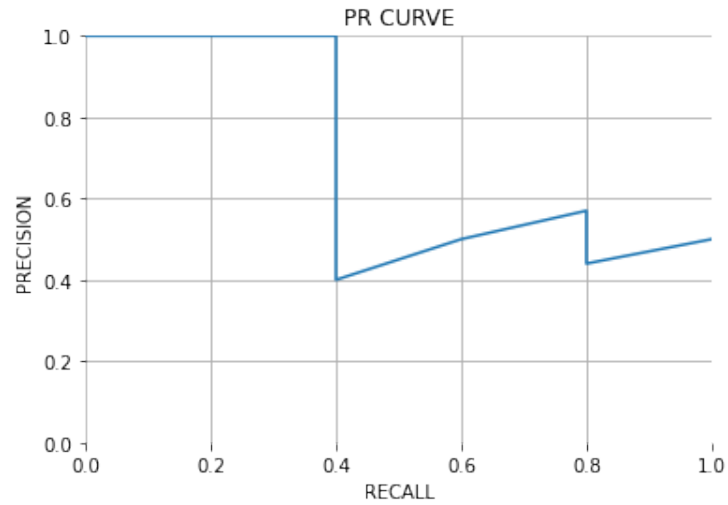
Precision and recall must be examined to evaluate a model's effectiveness fully. However, improving precision commonly reduces recall and vice versa. Hence, precision and recall are somewhat in conflict. A threshold is set to evaluate if a prediction is classified as a true positive. Lowering the threshold classifies more items as positive, thus increasing both FPs and TPs. A ROC curve plots precision vs. recall at different classification thresholds. The area under the ROC curve (AUC ROC) provides an overall measure of the quality of the model's predictions irrespective of the chosen classification threshold.



**Figure 3.5:** Receiver operating characteristic curve (ROC) shows true positive rate vs. false positive rate at different classification thresholds. Image adapted from: <https://commons.wikimedia.org/wiki/File:Roc-draft-xkcd-style.svg>

A single precision-recall (PR) curve is obtained by plotting precision vs. recall points for one classification threshold. The area under the PR curve (AUC RR) is commonly called the average precision (AP). Mean AP (mAP) is either calculated as the average AP for all classes or average AP for all thresholds. However, in some contexts and this thesis, they represent the same thing.

*AP is averaged over all categories. Traditionally, this is called “mean average precision” (mAP). We make no distinction between AP and mAP (and likewise AR and mAR) and assume the difference is clear from context. (Lin et al., 2014)*




**Figure 3.6:** Predictions are ranked in descending order according to the predicted confidence score for a given threshold. A zigzag pattern arises in the precision; it goes down with FPs and goes up again with TPs. Interpolating is usually done to reduce the impact of the pattern before calculating AP.

### 3.5.4 Intersection-over-Union

To decide if a predicted bounding box is a TP, one must compare it to the ground truth. The intersection-over-union (IoU) metric determines the amount of overlap between two boxes compared to their size. TPs are defined based on the IoU being greater than or equal to a threshold  $\tau$ ;  $IoU(\hat{\mathbf{y}}^{(i)}, \mathbf{y}^{(i)}) > \tau$ , where  $\tau$  is a the defined threshold. The IoU between two bounding boxes  $A$  and  $B$  is defined in equation 3.22.

$$IoU(A, B) = \frac{area(A \cap B)}{area(A \cup B)} \quad (3.22)$$

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


(a) Calculation of IoU as describe in equation 3.22. Image adapted from [https://commons.wikimedia.org/wiki/File:Intersection\\_over\\_Union\\_-\\_visual\\_equation.png](https://commons.wikimedia.org/wiki/File:Intersection_over_Union_-_visual_equation.png).



(b) The figure shows a poor IoU = 0.40, a good IoU=0.73, and excellent IoU = 0.92. The poor IoU would not be considered a true positive if the threshold was 0.5. Image adapted from: [https://commons.wikimedia.org/wiki/File:Intersection\\_over\\_Union\\_-\\_poor,\\_good\\_and\\_excellent\\_score.png](https://commons.wikimedia.org/wiki/File:Intersection_over_Union_-_poor,_good_and_excellent_score.png).

**Figure 3.7:** Visualization of how the IoU is calculated and evaluated.



# /4

## Deep learning and Object Detection

Object detection plays a vital role within the field of computer vision research. As a result, there is a continual improvement in automated object detection models. In recent years, machine learning models, particularly CNNs, have shown excellent performance and are often the foundation of most object detection systems. With the increase in popularity and performance, the models increase in size, and the most profound models consist of tens of millions of parameters. Before presenting the workings behind such models, the issue of training such models must be addressed.

### 4.1 Pretrained Models and Backbones

Training a deep neural network involves tweaking millions of parameters to learn a mapping. Generally, these parameters are randomly initialized before training. Consequently, too few parameters will be updated if the training data is not large enough, and the network will not learn the essential attributes required to produce proper predictions. A standard solution is to initialize the network with trained weights from a similar problem instead of random initialization. Hence the model is pre-trained. For example, we may find that learning to recognize apples might help us recognize pears, motivating *transfer*

*learning*. Transfer learning can be expressed as a domain adoption, e.g., (from apples to pears). Fine-tuning is *one* transfer learning technique where all pre-trained weights are initialized and updated. Other techniques involve initializing some network layers with pre-trained weights and regarding them as non-trainable. Commonly, these layers are called "frozen layers" or "frozen".

If a model builds on top of another model, or more typically parts of it, the parts are called *backbone* in the complete model. The backbone (pre-trained) serves as a feature extractor, and the complete model trains on the features. Today's deep neural networks include other well-known networks as the backbone, examples of backbones are VGG, ResNet, and AlexNet.

## 4.2 Object Detection

*Object detection* is a term that combines object localization and image classification, which in terms try to identify the location of one or more objects and predict the class of the objects (Brownlee, 2019). Consequently, object detection looks for structures and shapes in an image and processes this data to identify them. For example, this may include reporting current objects, annotating them with bounding boxes, or distinguishing the categories by labeling each pixel corresponding to a current class (Goodfellow et al., 2016). The latter is referred to as segmentation. This thesis focuses on annotating the mental foramina, a single class, with bounding boxes. Object detection models that output bounding boxes with a corresponding score for the class will be called detection models from now. Therefore, segmentation models are not considered further. However, object detectors explored in chapter III will be described in this section.

The process of detecting objects can be split into two parts: proposing regions then classifying and regressing bounding boxes. Standard detection models either employ proposal-based or grid-based methods to propose regions. Methods for proposing regions have a speed-accuracy tradeoff. Regardless, detection models (typically) represent objects by defining bounding boxes around the objects' region, and undefined regions are described as "background". The predicted bounding box is usually obtained as a result of the last layer in the architecture. The bounding box are parameterized using either corner coordinates or center coordinates, along with width and height, i.e.,  $\theta_{Box} = \{\theta_x, \theta_y, \theta_w, \theta_h, c\}$ . The confidence score is denoted  $c$ . The parameterization can be extended to multiple classes effortlessly.

### 4.2.1 Non-maximum Suppression

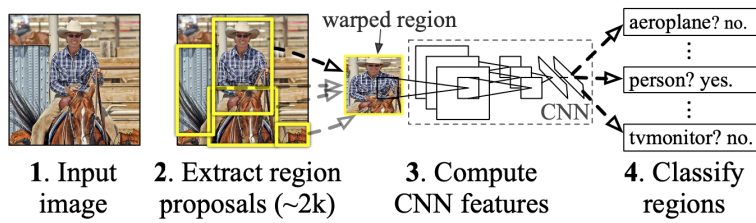
Non-maximum suppression (NMS), is used in virtually all state-of-the-art object detection pipelines (Rosenhahn and Andres, 2016). The reason being that detection models will typically propose numerous bounding boxes for each object. Nevertheless, only one of these covers the object in the most accurate way possible. As a solution to this problem, NMS eliminates all bounding boxes that do not match the best-predicted bounding box of an object. This is accomplished by arranging every prediction by confidence score, then, starting with the prediction with the highest confidence score, calculating the IoU between that prediction and all the predictions with a high confidence score. The only accept predictions are for which the IoU falls below a user-defined threshold and the confidence score exceeds a user-defined threshold.

### 4.2.2 Two-stage Detectors

Region proposal models are known as two-stage detectors because the pipeline constitutes, as the name suggests, two stages. The first stage proposes regions that may contain objects. Then, the proposed regions are fed into a network in the second stage. Proposal regions are sometimes called anchors, and the network evaluates the presence and category of objects in the proposal regions. A variety of methods have been used to generate region proposals. Earlier methods applied a sliding window, which involved predefined locations and testing all possible aspect ratios and window sizes. Unfortunately, testing all possible locations is computationally infeasible, and the sliding window approach is computationally expensive and inaccurate. In addition, problems with this approach would involve objects flowing across windows and choosing window size.

Later, models such as R-CNN (Girshick et al., 2014) used a Selective Search algorithm (Uijlings et al., 2013) in the first stage instead of a sliding window. The Selective Search algorithm identifies approximately 2000 regions where it believes objects are likely to be present. Selective Search achieves this with a classical approach, involving segmentation using pixel intensities and graph-based methods. In addition, the algorithm yields arbitrary shaped and positioned proposal regions, which in terms produces more accurate enclosing of objects.

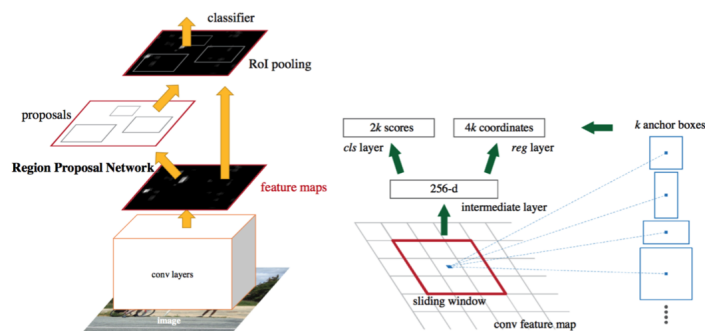
Fast R-CNN (Girshick, 2015) was released shortly after R-CNN and was a significant improvement over the original, and then Faster R-CNN (Ren et al., 2015) followed after the Fast R-CNN paper. In the first stage, Faster R-CNN proposed an end-to-end trainable detector, which meant that the region proposal algorithm was replaced with a *region proposal network*, i.e., RPN, that



**Figure 4.1:** R-CNN pipeline with its main components. Image credits: Girshick et al. (2014).

learned to predict desirable proposals. The reason was that Selective Search was functional but took a long time and constituted a bottleneck. In theory, a network that learned to predict higher quality regions would make more accurate predictions. The Faster R-CNN use an RPN (built in a fully trainable convolutional structure) as the backbone. Faster R-CNN allows the RPN to learn which proposed regions were accepted as objects during the second stage, leading to fewer, more accurate region proposals.

The authors (Ren et al.) observed that convolutional features maps used by a region-based detector could also generate proposal regions. Therefore, using a region-based object detector as a backbone (feature extractor), the authors proposed an RPN by adding two additional convolution layers to the backbone. The layers would encode convolution map position into a feature vector, and at each position, output an objectness score and regressed bounds for  $K$  region proposals relative to various scales and aspect ratios.



**Figure 4.2:** Main components in the Faster R-CNN model (left), and a conceptual sketch of the region proposal network (right). Image credit: Girshick (2015).

In order to generate region proposals, a small network fully connected to a spatial window slides over the feature map extracted by the last shared convolutional layer. Each sliding window is mapped to a lower-dimension vector. Instead of ROI pooling at this stage, the vector is fed into two small

FCNs, yielding box regression and box classification.

Ren et al. defines the objective function as follows:

$$L = \frac{1}{N_{cls}} \sum_{i=1}^N L_{cls}(\hat{c}^{(i)}, c^{(i)}) + \frac{\lambda}{N} \sum_{i=1}^N c^{(i)} L_1^{smooth}(\mathbf{l}^{(i)} - \mathbf{g}^{(i)}) \quad (4.1)$$

where  $N_{cls}$  and  $N$  are batch size and number of anchors, respectively. The confidence score associated with an anchor (proposed region) is denoted  $\hat{c}^{(i)}$ , and  $c^{(i)}$  is an indicator function comparing anchor and ground truth (1 if the anchor contains an object, 0 otherwise). The parameterized predicted bounding boxes and the associated ground truth bounding boxes are denoted  $\mathbf{l}$  and  $\mathbf{g}$ , respectively. The cross entropy (log) loss function (object vs. not object) is denoted  $L_{cls}$ , and is defined as:

$$L_{cls}(\hat{c}^{(i)}, c^{(i)}) = -c^{(i)} \log(\hat{c}^{(i)}) - (1 - c^{(i)}) \log(1 - \hat{c}^{(i)}) \quad (4.2)$$

$L_1^{smooth}$  is the smooth  $L_1$  loss:

$$L_1^{smooth}(\psi) = \begin{cases} 0.5\psi^2 & , |\psi| < 1 \\ |\psi| - 0.5 & , otherwise \end{cases} \quad (4.3)$$

The bounding boxes are parameterized as followed:

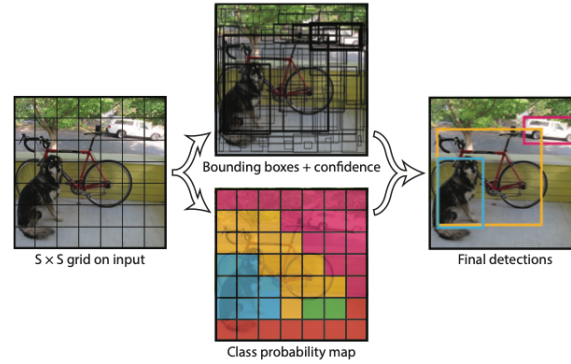
$$\begin{aligned} l_x &= (\theta_x - x_a)/w_a, & l_y &= (\theta_y - y_a)/h_a \\ l_w &= \log(\theta_w/w_a), & l_h &= \log(\theta_h/h_a) \end{aligned} \quad (4.4)$$

where  $\theta_x, \theta_y, \theta_w$ , and  $\theta_h$  denote box center coordinates along with width and height, respectively. The predicted x-coordinate is denoted  $\theta_x$  and  $x_a$  is the anchor x-coordinate. The other parameters are denoted in the same fashion, including the parameterized ground truth bounding boxes  $\mathbf{g}$ .

### 4.2.3 Single-stage Detectors

Single-stage detectors such as the Single Shot Multibox detector (Liu et al., 2016) or YOLO (Redmon et al., 2016) are models that skip the region proposal stage of two-stage detectors. Single-stage detectors seek an image understanding by looking at an image just once (one stage). Commonly, such models use a predefined grid (see figure 4.3). The grid divides the image, where each grid cell is responsible for detecting objects in that region of the image. These types

of models are commonly known for faster inference, typically at the cost of precision.



**Figure 4.3:** YOLO architecture with its main components. Image adapted from Redmon et al. (2016).

#### 4.2.4 RetinaNet

RetinaNet is a single-stage detector, Lin et al. proposed Focal Loss (equation 4.5) that adds a modulating factor to the standard cross entropy loss in equation 4.2. The motivation was due to the large class imbalance encountered during training that overwhelms the cross entropy loss. The modulating factor down-weight the loss assigned to well-classified examples and thus focus training on difficult examples.

The cross entropy loss in equation 4.2 can be written as:

$$L = \begin{cases} -\log(\hat{c}^{(i)}), & \text{if } c^{(i)} = 1 \\ -\log(1 - \hat{c}^{(i)}), & \text{otherwise} \end{cases}$$

we define  $c$  as:

$$c = \begin{cases} \hat{c}^{(i)}, & \text{if } c^{(i)} = 1 \\ 1 - \hat{c}^{(i)}, & \text{otherwise} \end{cases}$$

the focal loss introduced is then:

$$FL(c) = -\log(c)(1 - c)^\gamma \quad (4.5)$$

where  $\gamma \geq 0$  is with tunable parameter that smoothly adjusts the rate at which easy examples are down-weighted.

The objective function of RetinaNet is defined as followed:

$$L = L_1^{smooth}(\mathbf{l}^{(i)} - \mathbf{g}^{(i)}) + FL(c) \quad (4.6)$$

### 4.2.5 CenterNet

CenterNet is a single-stage detector, the authors of CenterNet (Duan et al.) suggest an approach that represents each object as a triplet of keypoints, specifically one center keypoint and a pair of corners. The authors' intuition was that if a predicted bounding box regarding a class has a high IOU with the ground-truth box, it is highly likely that the center keypoint will be predicted as the same class, and vice versa. Accordingly, during inference, after a proposal is generated as a pair of corner keypoints, we determine if the proposal is indeed an object by checking if there is a center keypoint of the same class falling within its central region. To improve keypoint detection, the authors proposed two methods, center pooling, and cascade corner pooling, to enhance keypoint information. Without going into too much detail, we present the three loss functions that constitute the final objective function.

The training loss is a penalty-reduced pixel-wise logistic regression with local loss:

$$L_k = \frac{-1}{N} \sum_Y \begin{cases} (1 - \hat{Y})^\alpha \log(\hat{Y}), & \text{if } Y = 1. \\ (1 - Y)^\beta (\hat{Y})^\alpha \log(1 - \hat{Y}), & \text{otherwise.} \end{cases} \quad (4.7)$$

1. When  $Y = 1$  (prediction in the vicinity of ground truth heat map), and if the classification  $\hat{Y}$  score is close to one, the focal loss will decrease the influence of the loss. The opposite holds for difficult examples, as  $\alpha$  increases.
2. If  $Y \neq 1$ , and  $\hat{Y}$  is close to zero,  $(\hat{Y})^\alpha$  makes the overall loss zero, further, if  $\hat{Y} \approx 1$ , there is no drop in values, because a Gaussian kernel is used to compute  $Y$  as a heatmap, so the values are considered positive candidates.

Additionally a local offset loss is added for each centerpoint. All classes  $c$  share the same offset prediction:

$$L_{off} = \frac{1}{N} \sum_p |\hat{O}_{\tilde{p}} - (\frac{p}{R} - \tilde{p})| \quad (4.8)$$

Finally,  $L_1$  loss is added at the centerpoints to regress the width and height of the bounding boxes:

$$L_{size} = \frac{1}{N} \sum_{k=1}^N |\hat{S}_{pk} - s_k| \quad (4.9)$$

where  $\hat{S}_{pk}$  denote the predicted dimension of the bounding box and  $s_k$  are ground truth bounding box dimension.

Overall training objective is then:

$$L = L_k + \lambda_{size} L_{size} + \lambda_{off} L_{off} \quad (4.10)$$

where  $\lambda_{size}$  and  $\lambda_{off}$  are hyperparameters.

## 4.2.6 EfficientDet

The EfficientDet is a single-stage detector (Tan et al., 2020), that is one of the most recent state of the art model architectures and overcomes two problems regarding multi-scale feature fusion and model scaling. The latter usually sacrifice either accuracy or efficiency of the object detection model. The act of feature fusion is the combination of features from different layers, and in the simplest case, it is a summation or concatenation of features. The issue arises when the features of different resolutions, are fused; then, the features usually contribute unequally to the resulting fused feature.

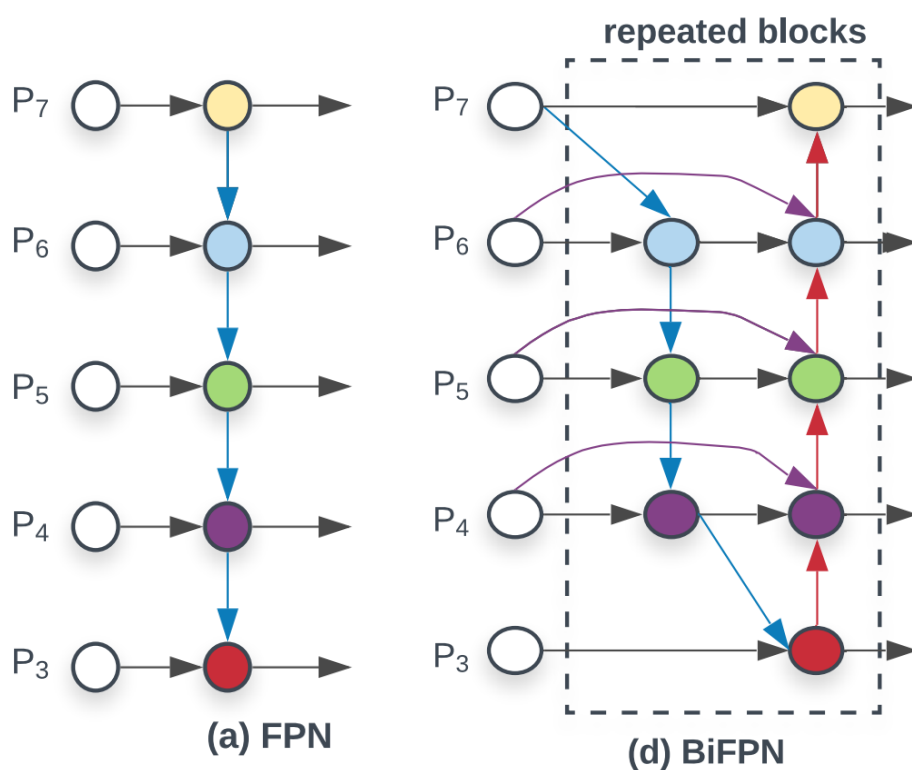
To overcome these problems, EfficientDet utilizes compound scaling, inspired by EfficientNets (Tan and Le, 2019), to jointly scale up the backbone regarding input resolution, width, and depth of the network, and box/class prediction network. In addition, EfficientDet suggests a weighted bi-directional feature pyramid network (BiFPN), yielding easy, fast multi-scale features fusion (see figure 4.4).

### 4.2.6.1 BiFPN

The main ideas behind BiFPN are efficient bidirectional cross-scale connections and weighted feature fusion.

Given a list of multi-scale features  $\mathbf{P}^{in} = (P_{l_1}^{in}, P_{l_2}^{in}, \dots)$ , where  $P_{l_i}^{in}$  represents the feature at level  $l_i$ , and  $\mathbf{P}^{out} = f(\mathbf{P}^{in})$  is the transformation that aggregate different features, then the conventional top-down FPN aggregate features across scales in the following manner:





**Figure 4.4:** Figure showing FPN in a top-down fashion (left), and BiFPN (right). Image credits: Tan et al. (2020).

$$\begin{aligned}
 P_7^{out} &= Conv(P_7^{in}) \\
 P_6^{out} &= Conv(P_6^{in} + Resize(P_7^{out})) \\
 &\dots \\
 P_3^{out} &= Conv(P_3^{in} + Resize(P_4^{out}))
 \end{aligned}$$

where *Resize* and *Conv* usually is a up- or down-sampling operation for resolution matching, and convolutional operation for feature processing, respectively.

Tan et al. proposed the following steps to optimize cross-scale connection: 1) Remove nodes only having one input link. 2) Add an extra link from the original input node to the output node if they are at the same level. 3) Treat each top-down and bottom-up (bidirectional) path as one feature network layer, and repeat the layer. The result removes nodes that contribute less to the feature network aiming to fuse different features, fuse more features without

adding much cost, enable more high-level feature fusion. Further, to address the problem of unequally feature contribution, the fused features are weighted:

$$O = \sum_i \frac{w_i}{\epsilon + \sum_j w_j} I_i \quad (4.11)$$

where  $w_i$  is a learnable weight, and ensured to be larger or equal to 0 by applying a ReLU activation function. To avoid numerical instability,  $\epsilon = 0.0001$ .

#### 4.2.6.2 Compound Scaling

Tan et al. propose a compound scaling method, using a coefficient  $\phi$  to jointly scale up all dimensions of the backbone network, BiFPN network, class/box prediction network, and resolution. The depth of BiFPN, is linearly scaled as  $\phi + 3$ , and the width is exponentially scaled as  $64 \times 1.35^\phi$ . The width of the box/class prediction networks are scaled as BiFPN, but the depth is linearly increased as  $3 + \phi/3$ . Input image resolution is linearly increased as  $512 + \phi \cdot 128$ . We point out that EfficientDet D0 implies that  $\phi = 0$ .

# /5

## Tromsø Survey 7 Data Set

The data set used for experimentation in this thesis consists of dental panoramic x-ray images taken during the seventh survey of the Tromsø Study. The chapter will describe the collection and modification of the data set and its basic properties.

Tromsø Study is a population-based study carried out in a repeated cross-sectional fashion (33). A total of seven surveys have been conducted. All residents of Tromsø who are at least 40 years of age and live in the municipality were invited to participate in Tromsø 7 (33). Tromsø 7 consisted of a questionnaire-based survey and clinical examination, including DEXA measurements and dental panoramic radiographs. A letter inviting potential participants and a questionnaire were sent by mail. The seventh Tromsø study enrolled 10,009 men and 11,074 women aged 40-99. Accordingly, the participation rate was 62.4% and 67.0% (33), respectively.

A total of 3970 DPRs were collected following the clinical dental examination. The DPRs consist of 2821 by 1376 pixels and were in TIF format, with a DPI (dots per inch) of 257. Knowing the DPI makes it possible to convert between pixels and physical size. In addition, two regions of interest were automatically cropped out for every image at an exact location. The resulting crops were 300 × 600 (height, width) pixels.

During the examination, high patient positioning caused some crops to be too low in the jaw and was therefore discarded. In addition, distorted images

and images with obstructing artifacts were also rejected. Finally, the image was rejected if the experts did not recognize the mental foramen's position. Out of 7940, 5197 crops were usable and annotated by the experts using VIA (Dutta and Zisserman, 2019). The data was divided into 4157 training and 1040 test images. It should be pointed out that to save time, the experts divided the workload, not annotating the same image. However, to establish the IoU between them, 706 images were annotated by both experts. The resulting average IoU calculated between 0.5 and 0.95 was 0.678.

## **Part III**

# **Results and Discussion**



# /6

## Experiments and Results

In this chapter, we present and discuss the results of our experiments. We present a feasibility study indicating that the proposed pipeline is, in fact, able to fine-tune object detectors to be adequate in detecting the mental foramen in x-ray images. Architectures presented in chapter ?? (section 4.2) will first be tested, and relevant performance metrics will be presented. In addition, the model with the highest average precision (AP) will be further examined by investigating predictions from two scenarios. The testing and fine-tuning were performed on a GeForce RTX 2080 Ti 11 GB GPU. In addition, as a result of using an object detector along with previous research (Paasche Edvardsen, 2021), an automated method capable of detecting and measuring bone width at an appropriate location will be reviewed.

In the first section of this chapter, a description of how each model was set up will be explained. The results will be presented and discussed in the second section in the second section.

The following models, pre-trained on the COCO dataset (Lin et al., 2014), were "fine-tuned" to our dataset using Tensorflow framework (Abadi et al., 2015):

1. Faster R-CNN with ResNet50 (He et al., 2016) as the backbone.
2. CenterNet with HourGlass104 (Newell et al., 2016) as the backbone.
3. EfficientDet D0 with EfficientNet-B0 (Tan and Le, 2019) as the backbone.

4. RetinaNet (Lin et al., 2017) with ResNet50 as the backbone.

Now, "trained" implies "fine-tuned". We put "fine-tuned" in quotation marks as the COCO dataset and the dataset used in this thesis are from different domains. In addition, the backbones will not be stated when referring to the models as it is cumbersome.

## 6.1 Experimental Setup

For experiments on object detectors, the IoU threshold  $\phi_{IoU}$  and confidence score threshold  $\phi_c$  used during NMS were set to 0.5 and virtually 0 for all models except CenterNet, which do not use NMS. Although a brief explanation of NMS was given in section 4.2.1, the reader is reminded that  $c$  is the threshold that discard predictions with a score  $c < \phi_c$ . Setting this parameter to 0 means all proposals are accepted at the beginning of NMS. We assume it is beneficial in challenging scenarios where the predicted scores can be poor. Each model was trained with two different configurations (set-up 1 and set-up 2), and the results are presented in table 6.1 and table 6.2 (one for each configuration).

The batch size is set to six for all experiments (unless something else is specified), and we train for 30 epochs. Since the training data consists of 4157 examples, processing six simultaneously (a batch) results in  $\sim 693$  gradient updates (training steps), to cycle through the training data once (one epoch). Therefore, to train for 30 epochs with a batch size of six requires a total of  $\sim 21000$  steps. Empirically, using the moving average of the trained parameters has shown to be better than using trained parameters directly. However, we do not employ a moving average in any experiment due to technical limitations.

### 6.1.1 Faster R-CNN

**Set-up 1:** The SGD optimizer (Qian, 1999b) was used with momentum 0.9 and  $L_2$  regularization (decay =  $4 \times 10^{-4}$ ). The learning rate grows linearly from  $1 \times 10^{-2}$  to  $4 \times 10^{-2}$  for 2000 steps, then transitioned down using a cosine decay rule (Loshchilov and Hutter, 2016). ReLu activation is employed between convolutional layers. The anchor generator used aspect ratios (1/2, 1, 2) at scales (1/4, 1/2, 1, 2). The training images had a 50% probability of being flipped horizontally.

**Set-up 2:** From the first set up we change to the following, the rest is unchanged: Adam optimizer ( $\epsilon = 1 \times 10^{-7}$ ) with learning rate  $2 \times 10^{-4}$  that dropped to  $1 \times 10^{-4}$  at epoch 6,  $8 \times 10^{-5}$  at epoch 10, and  $4 \times 10^{-5}$  at epoch 15.



### 6.1.2 RetinaNet

**Set-up 1:** The SGD optimizer (Qian, 1999b) was used with momentum 0.9 and  $L_2$  regularization (decay =  $4 \times 10^{-4}$ ). The learning rate grows linearly from  $1 \times 10^{-2}$  to  $4 \times 10^{-2}$  for 2000 steps, then transitioned down using a cosine decay rule (Loshchilov and Hutter, 2016). Synchronized batch normalization was added after every convolution with batch norm decay of 0.99 and  $\epsilon = 1 \times 10^{-3}$ . ReLU activation was employed but capped at 6. Standard smooth L1 was the localization loss, and focal loss with  $\alpha = 0.25$  and  $\gamma = 2$  was the classification loss. The anchor generator used aspect ratios (1/2, 1, 2). The training images had a 50% probability of being flipped horizontally. The feature pyramid use minimum level 3 and maximum 7.

**Set-up 2:** From the first set up we change to the following, the rest is unchanged: Adam optimizer (Kingma and Ba, 2014), where the learning rate grows linearly from  $2 \times 10^{-4}$  to  $2 \times 10^{-3}$  for 2100 steps, then transitioned down using a cosine decay rule.

### 6.1.3 CenterNet

**Set-up 1** The Adam optimizer was used ( $\epsilon = 1 \times 10^{-7}$ ) for training with constant learning rate  $9.9 \times 10^{-4}$ . For the penalty-reduced pixel-wise logistic regression with focal loss,  $\alpha$  and  $\beta$  was set to 2 and 4, respectively. The loss is scaled by  $\lambda_{size} = 0.1$  and  $\lambda_{off} = 1.0$ . The training images had a 50% probability of being flipped horizontally, cropped, contrast adjusted or brightness adjusted.

**Set-up 2:** From the first set up we change to the following, the rest is unchanged: The Adam optimizer was used ( $\epsilon = 1 \times 10^{-7}$ ) for training with learning rate  $5 \times 10^{-4}$  for 30 epochs, that dropped 10× at epoch 18 and 24.

### 6.1.4 EfficientDet-Do

**Set-up 1:** Adam optimizer ( $\epsilon = 1 \times 10^{-7}$ ) with learning rate  $2 \times 10^{-2}$  for 30 epochs, that dropped 10× at epoch 18 and 24. Synchronized batch normalization was added after every convolution with batch norm decay of 0.99 and  $\epsilon = 1 \times 10^{-3}$ . Swish-1 (Ramachandran et al., 2017) (commonly called SiLu) activation was employed. Standard smooth L1 was the localization loss, and focal loss with  $\alpha = 0.25$  and  $\gamma = 1$  was the classification loss. The anchor generator used aspect ratios (1/2, 1, 2, 4). The training

images had a 50% probability of being flipped horizontally. The feature pyramid use minimum level 3 and maximum 7.

**Set-up 2:** From the first set up we change to the following, the rest is unchanged: Adam optimizer ( $\epsilon = 1 \times 10^{-7}$ ) with learning rate  $2 \times 10^{-4}$  that dropped to  $1 \times 10^{-4}$  at epoch 6,  $8 \times 10^{-5}$  at epoch 10, and  $4 \times 10^{-5}$  at epoch 15. Random cropping was added as well, and batch size was increased to 8.

### 6.1.5 Procedure to Estimate MCW

The procedure to estimate the mandibular cortical width (MCW) is briefly described in in algorithm 1. For more extended details, the reader is referred to the original article (Paasche Edvardsen, 2021). The procedure is improved by including the trained object detector. In addition, to find the bone's lowest edge, Paasche Edvardsen proposed gray-scale dilation applied to the variance image before applying the Canny edge detector (Canny, 1986); the dilation was omitted to find the bone's edge more accurately. The reason being dilation would expand the structure of bone, pushing the edge away from its original position by a small degree. Further, the stop criterion in algorithm 1 is a user-defined threshold that represents the percentage of the line segment  $L$  overlapping with black pixels in the binary image  $I_b$  (see figure 6.11 b). The threshold was set to 0.7 in this work.

After algorithm 1 terminates, the width of the bone is the distance between the parallel lines: the initial line and the resulting line. The distance is calculated with equation 6.1.

$$d = \frac{|c_1 - c_2|}{\sqrt{1 + m^2}} \quad (6.1)$$

where  $c_1$  and  $c_2$  is the y-intercept of the lines and  $m$  is the slope.

---

**Algorithm 1:** Bone width measuring method adapted and from Paasche Edwardsen (2021) and improved with an object detector.

---

**Find bone's lowest edge:**

1. Find MF's location  $P$  with an object detector
2. Convert image to gray-scale and apply median filtering with kernel size 11
3. Apply a variance filter with kernel size 5, and follow with Canny Edge detector
4. Use morphology to remove objects smaller than 150 pixels with a neighborhood of 500 pixels
5. Use probabilistic Hugh transform (Kiryati et al., 1991) to retrieve possible line segments representing the lower bone edge, save line segment  $L$  closest to  $P$

**- Find bone's upper edge (part 1):**

1. Convert image to gray-scale and apply variance filter with kernel size 8
2. Follow with exposure equalization to obtain  $I_V$
3. Apply a uniform filter with kernel size 11 to  $I_V$  to obtain  $I_M$
4. Calculate the binary image  $I_b$

$$I_b = \begin{cases} 1, & \text{if } I_M - I_V \leq \sigma^2 \\ 0, & \text{otherwise} \end{cases}$$

where  $\sigma^2$  is the variance of  $I_V$

**Find bone's upper edge (part 2):**

- Initialize:

Place line segment  $L$  on  $I_b$

**while** *stop criterion not fulfilled* **do**

  | Move  $L$  towards  $P$

**end**

---

## 6.2 Results

Accuracy is a vital concern when studying object detection models' performance. Typically, object detection models have a speed vs. accuracy trade-off which has significant consequences for the application. However, a post x-ray examination analysis does not require instant results, therefore speed is overlooked and we focus on maximizing accuracy.

### 6.2.1 Detecting the MF

The different model's performances are listed in table 6.1 and 6.2. From the table, it is clear that EfficientDet D0 performed better regarding average precision. This is true for both cases when 0.50 and 0.75 IoU were the threshold for a prediction labeled as true positive. We point out that EfficientDet D0 only use a fraction of the number of parameters compared to the other models. However, CenterNet is very close to similar results, and RetinaNet had higher average recall regarding 100 detections. In addition, we notice that the second configuration of every model produced better mean average precision than the first.

Model	mAP	mAP@0.5IoU	mAP@0.75IoU	AR@100
Faster R-CNN	0.24	0.68	0.069	0.33
CenterNet	0.22	0.68	0.064	0.34
EfficientDet D0	0.23	0.70	0.007	0.21
RetinaNet	0.21	0.62	0.010	0.46

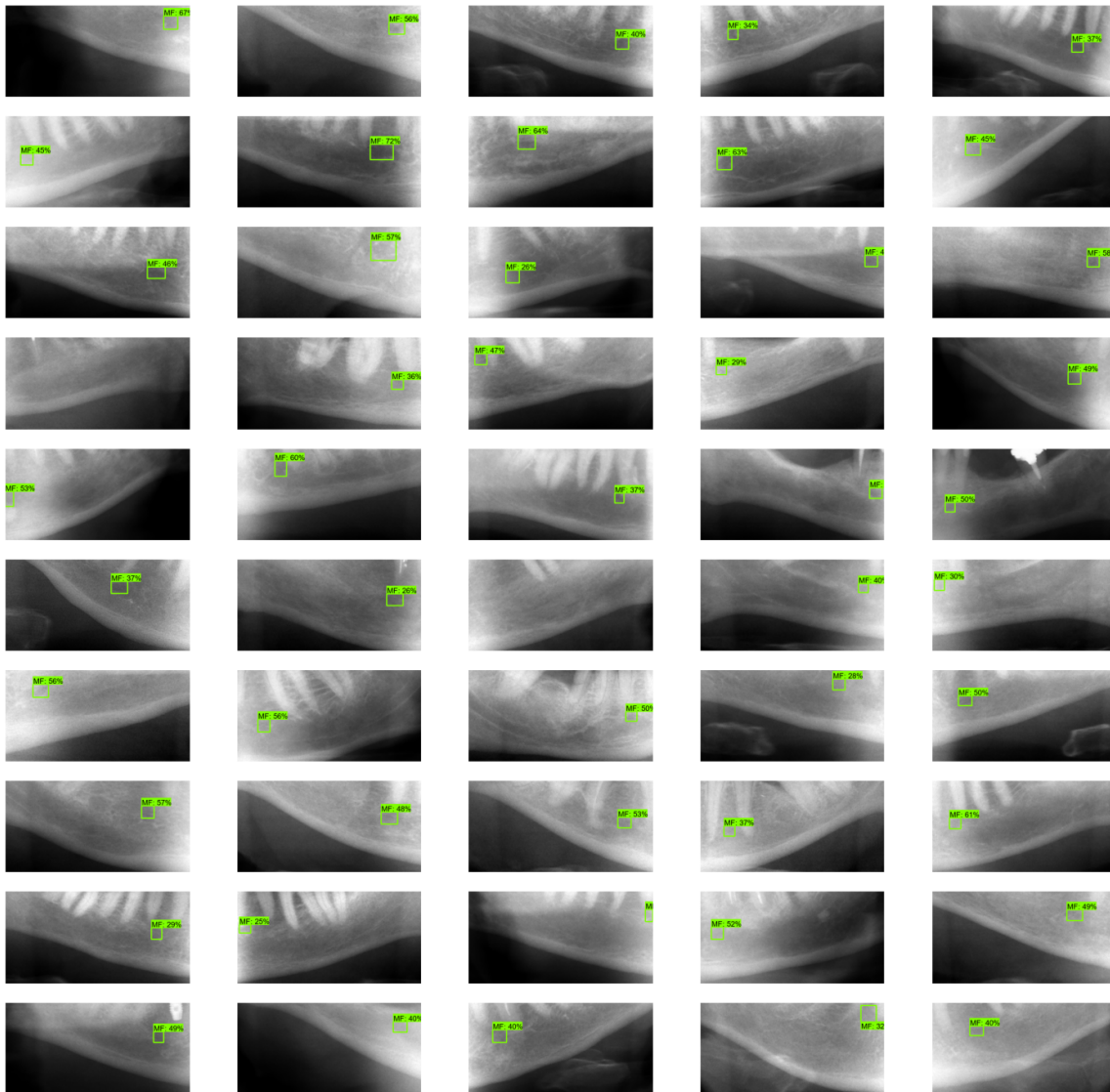
**Table 6.1:** Test results from the object detector with **experimental set-up 1** of the object detectors presented in chapter 4.2 using the Tromsø 7 dataset described in 5.

Model	mAP	mAP@0.5IoU	mAP@0.75IoU	AR@100
Faster R-CNN	0.25	0.72	0.08	0.39
CenterNet	0.28	0.75	0.13	0.39
EfficientDet D0	<b>0.30</b>	0.79	0.14	0.43
RetinaNet	0.23	0.64	0.010	0.47

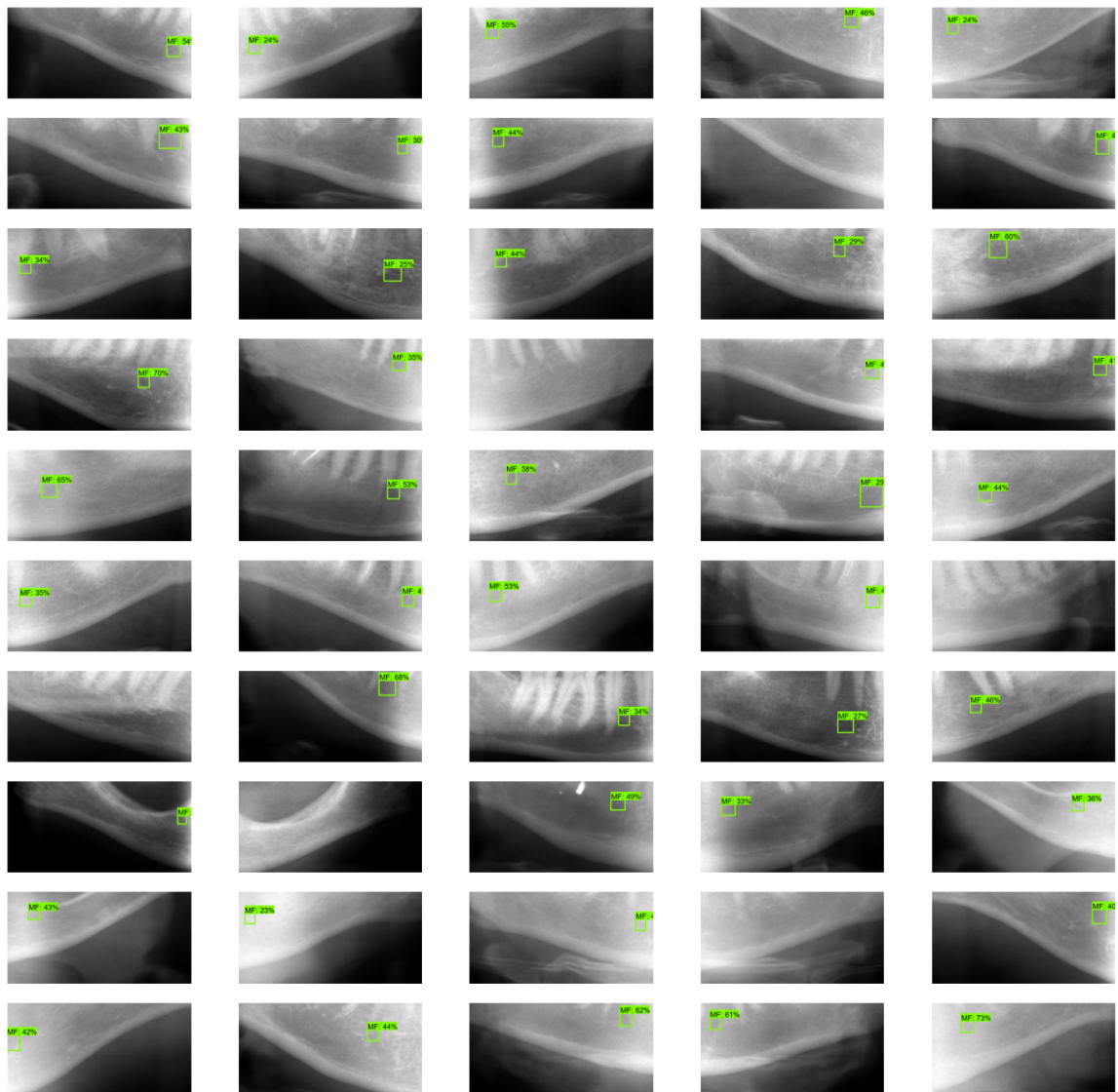
**Table 6.2:** Test results from the object detector with **experimental set-up 2** of the object detectors presented in chapter 4.2 using the Tromsø 7 dataset described in chapter 5.

To better understand EfficientDet D0's predictions, two experts in clinical dentistry, hereby referred to as *the experts*, have handpicked 100 images where the MF is distinguishable, hereby referred to as *easy images*, and 101 images where it is challenging to locate, hereby referred to as *complex images*. The 100 easy images with a bounding box prediction are visualized in figures 7.1 and 7.2 in the appendix (chapter V). The 101 complex images with a bounding box prediction are visualized in figures 6.1 and 6.2 below.

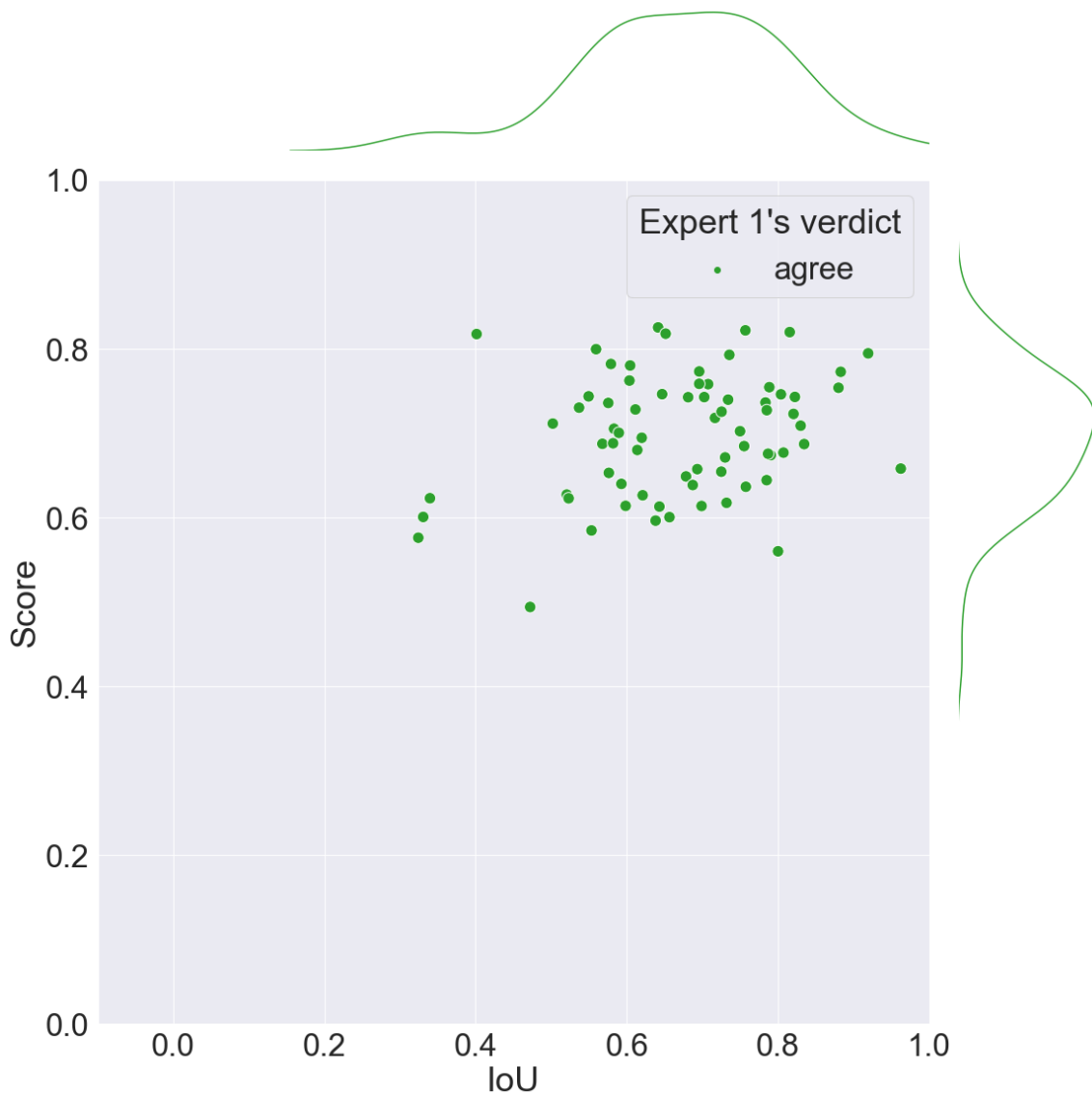
The experts agreed with all 100 predictions on the easy images. On the other hand, the experts did not agree with all 101 predictions on the complex images. Both cases are visualized as data points with prediction scores vs. IoU and expert opinion labels in figures 6.3, 6.5, and 6.6. However, only 66/101 complex images are visualized in figures 6.5, and 6.6 since the remaining images were far too complex to annotate with a ground truth bounding box, and was not a part of the dataset. Therefore, the IoU could not be calculated.



**Figure 6.1:** Figure visualizing 50 out of 100 predictions from EfficientDet D0 on the complex images.



**Figure 6.2:** Figure visualizing 50 out of 100 predictions from EfficientDet D0 on the complex images. It is evident that detecting the MF is challenging as we see several undetected cases and predictions too close to the tooth's root.



**Figure 6.3:** Figure visualizing predicted score vs IoU. Expert 1 has manually inspected the results indicating whether they agree with the predicted results.

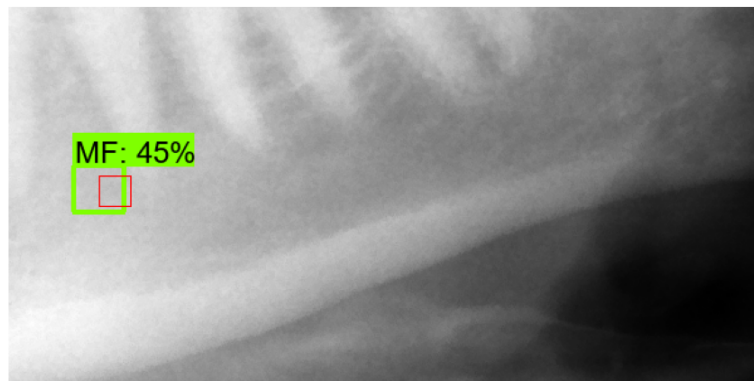
From figure 6.3, we see that all predictions on the easy images (figures 7.1 and 7.2 in appendix) agree with the expert's opinion (the figure is the same regarding expert 2, see 7.3). The score on the y-axis is the output from the sigmoid function, and IoU on the x-axis is calculated between the predicted bounding boxes and ground truths; the same applies to the following figures unless something else is specified. Here, the mean and variance of the IoU between the predictions and ground truths are 0.671 and 0.0173, respectively.



The mean and variance of the score are 0.695 and 0.00577, respectively.

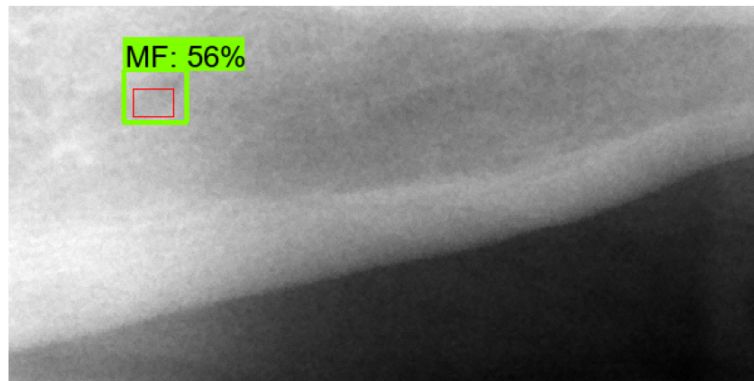
Interestingly, the experts agree with the predictions with IoU less than 0.5. An example is shown in figure 6.4, where the prediction is bigger and contains the ground truth. The mental foramen's border and size are challenging to see for an untrained eye and, in this case, also the model. However, even if the IoU is poor, when the  $\text{IoU} > 0$ , there is a connection to the ground truth bounding box by definition. Consequently, the result can be a good suggestion.

IoU (overlap) = 0.31



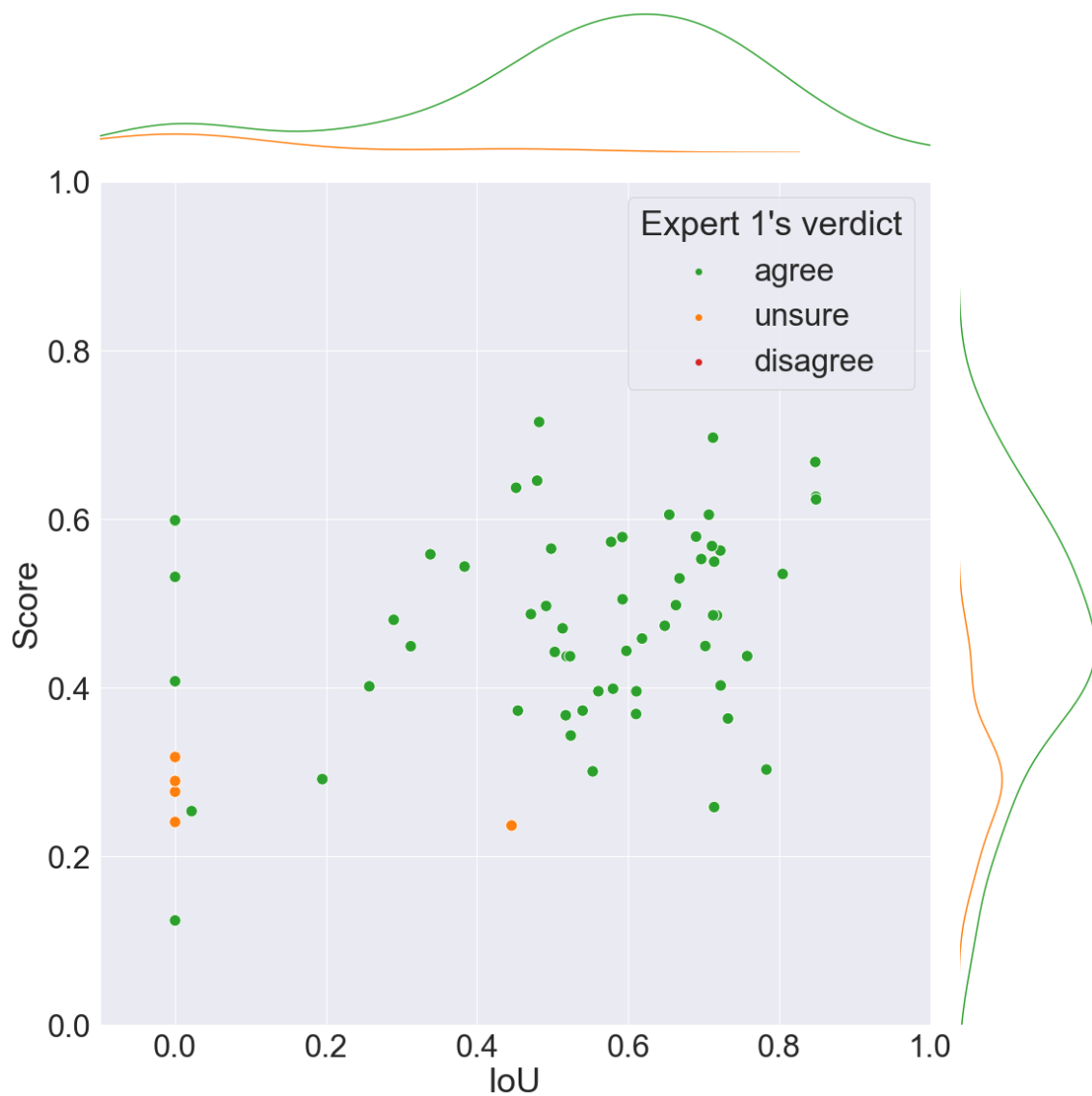
(a) Example of a predicted bounding box (in green) and corresponding ground truth bounding box (in red). The IoU is 0.31 in this situation

IoU (overlap) = 0.34



(b) Example of a predicted bounding box (in green) and corresponding ground truth bounding box (in red). The IoU is 0.34 in this situation

**Figure 6.4:** An example showing that a prediction with low IoU can still give a good suggestion for the MF's position



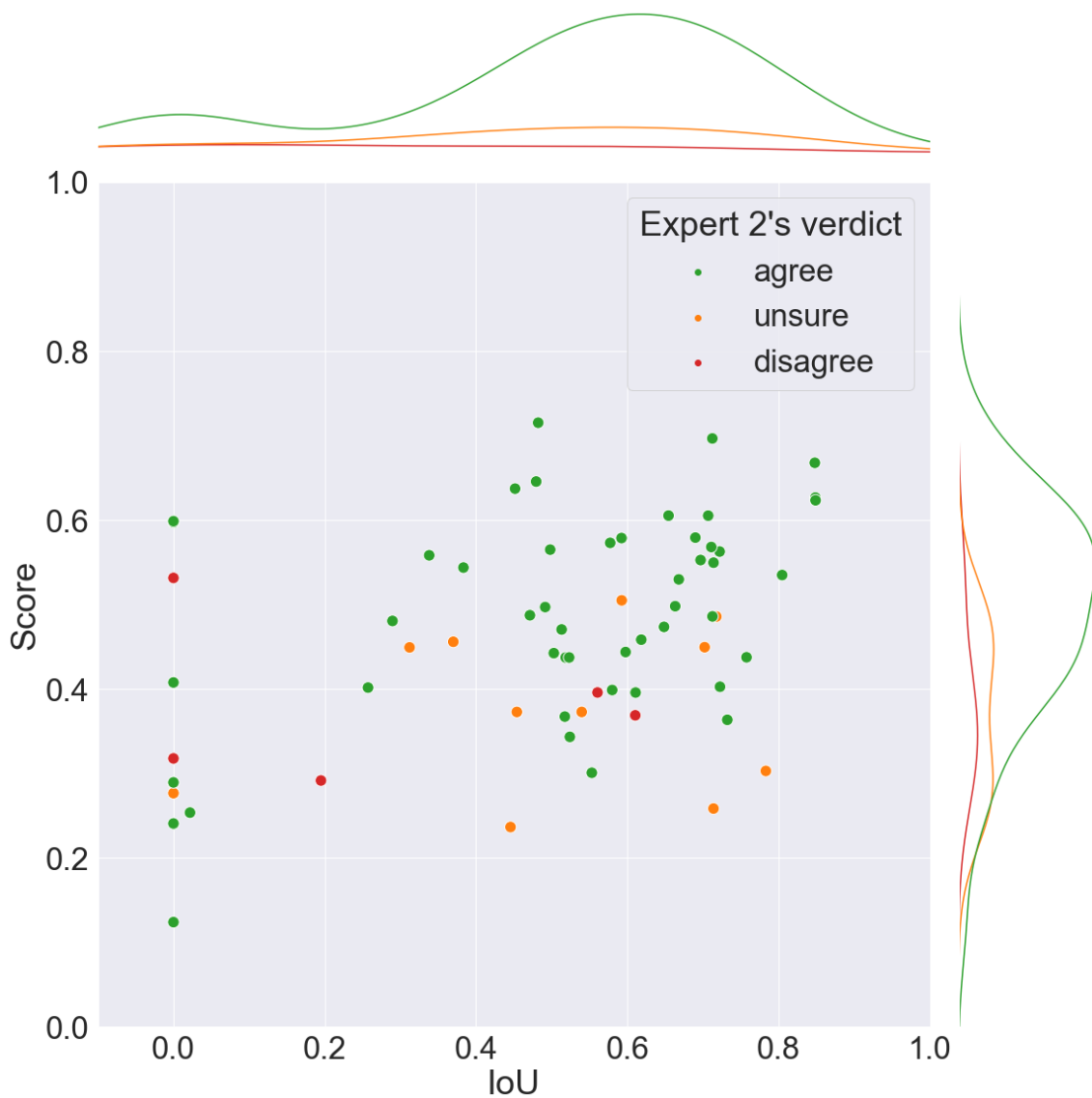
**Figure 6.5:** Figure visualizing predicted score vs IoU. Expert 1 has manually inspected the results indicating whether they agree with the predicted results.

From figure 6.5, we see that the expert does not fully agree with all predictions on the complex images (6.1, 6.2). Here, the mean and variance of the IoU between the predictions and ground truths are 0.489 and 0.0666, respectively. The mean and variance of the score are 0.425 and 0.0207, respectively.

Interestingly, several predictions have relatively low IoU, and the expert agrees with these; the reason was covered regarding the last figure (6.3). Most striking

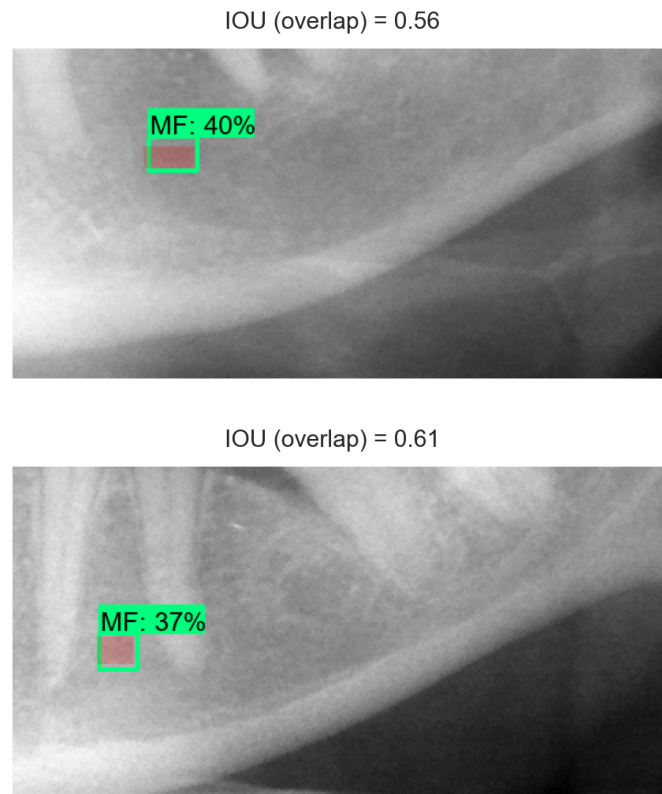
is that even when the IoU is 0, the expert agrees or is unsure. There is also one case with  $\text{IoU} > 0.4$  where the expert is uncertain. The latter is because the predicted region contains part of the tooth's root apex, which also is a dark region similar to the MF. Further, the image crop is too small to state which root apex the predicted region contains. Recall from section 2.1 that it is unlikely that the MF is located in the projection of the root apex of the first premolar; this helpful information was cropped out of the image.

It may seem contradictory to agree with a prediction when the IoU is 0. However, it should be stated that ground truths are not absolute. With this in mind, the zero IoU predictions have in common that they are on the mandibular canal next to the ground truth. Therefore, these predictions are also likely to contain the MF, so the expert accepts them. However, regions marked as "unsure" are again because they contain the root apex but simultaneously lay on the mandibular canal.



**Figure 6.6:** Figure visualizing predicted score vs IoU. Expert 2 has manually inspected the results indicating whether they agree with the predicted results.

A glance at figure 6.6 shows that the second expert disagrees with several predictions on complex images. In addition, 16 points differ from figure 6.5. Surprisingly, two predictions with relatively high IoU ( $> 0.5$ ) were in disagreement with expert 2. Looking at these cases in figure 6.7, we discover that the ground truth bounding boxes were the best guess placed on the crops. However, the whole image was used for evaluation here, as the image region was insufficient. Therefore, the expert disagreed with the predictions on the crops.



**Figure 6.7:** In such highly complex cases, changing the contrast in the complete image helps ease the evaluation of the MF's position. Here prediction is in green and ground truth are filled with red.

Predictions in agreement with no overlap were discussed concerning figure 6.5. However, expert two disagrees with one prediction with 0 IoU, where expert one agrees. Further, one prediction with no overlap where expert one was unsure, expert two disagreed. Both predictions lie on the mandibular canal, emphasizing the challenge of locating the MF.

It was stated earlier that not all the predictions on complex images could be visualized as data points. Therefore the evaluations are summarized in table 6.3. Inter-rater reliability, or agreement, can be measured using the kappa statistic (Landis and Koch, 1977). Kappa is frequently used in biostatistics. The kappa statistic is more robust than a percent agreement calculation, as it considers the possibility of the agreement occurring by chance Szкло and Nieto (2014). However, there is a dispute surrounding the kappa statistic as it is difficult to interpret. Nevertheless, using three categories ("agree", "unsure", and "disagree"), kappa = 0.18, a slight agreement (Landis and Koch, 1977).

	Expert 2 (agree)	Expert 2 (not sure)	Expert 2 (disagree)
Expert 1 (agree)	67	12	7
Expert 1 (unsure)	7	5	2
Expert 1 (disagree)	0	1	0

**Table 6.3:** Evaluation of 101 complex images by two experts

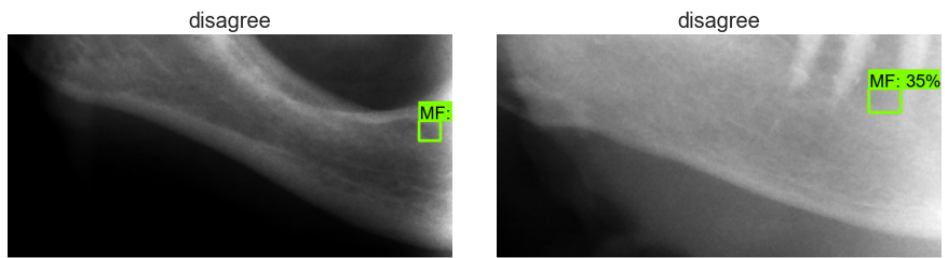
From table 6.3, we see that expert one disagreed once, even though it did not show in figure 6.5. If we assume the experts combined agree with the predictions, if one or both were unsure, and disagree with the predictions when one of them disagreed (see table 6.4, kappa would be 0.44 meaning a moderate agreement (Landis and Koch, 1977)).

	Expert 1 (agree)	Expert 2 (disagree)
Expert 2 (agree)	91	7
Expert 2 (disagree)	0	3

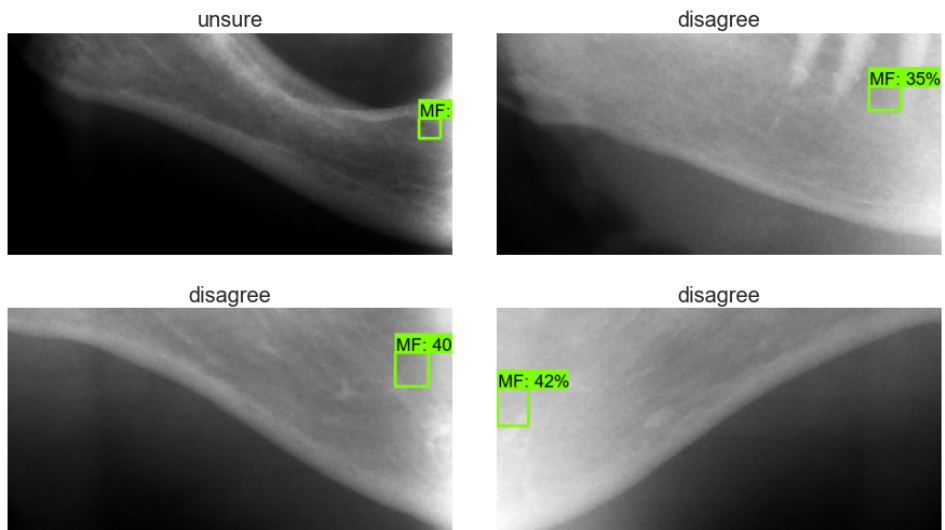
**Table 6.4:** Combined evaluation of 101 complex images

It seems confusing that the agreement was moderate and slight while the experts agreed in most cases. There were only a few cases when the algorithm marked mental foramen wrong. When an event is rare, its proportion in a study sample is low, and the kappa value tends towards 0 (Szklo and Nieto, 2014).

As stated before, not all of the complex images that were handpicked for inference had ground truth bounding boxes. The reason was that the experts could not to locate the MF when creating ground truth bounding boxes. These highly complex images were given to the model, and the experts evaluated the results (see table 6.3). It was stated that there was one case where expert one disagreed with the prediction while expert two were unsure. In addition, there was one case where expert one was unsure but leaned towards disagreeing. In the same case, expert two disagreed with the prediction. These cases are visualized in figures 6.8 and 6.9. For all cases shown in figures 6.8 and 6.9, the experts concluded that the model annotated a part of the tooth's root apex or the experts could not see the MF, and therefore disagreed.



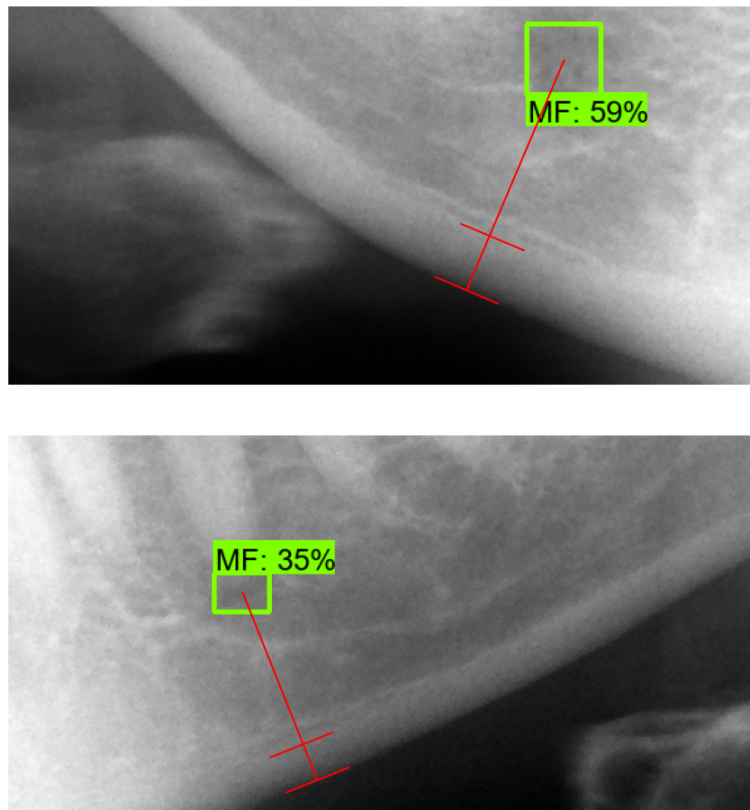
**Figure 6.8:** Figure displaying incorrect predictions from EfficientDet D0 judged by the first expert.



**Figure 6.9:** Figure displaying incorrect predictions from EfficientDet D0 judged by the second expert.

### 6.2.2 Estimating MCW

The proposed algorithm-1 operates *fully* automatically given an image region. In figure 6.10, we see that the algorithm indeed does a good job of locating the MF, and estimating the bone thickness in a completely automatic fashion. Out of 100 random images (not necessarily in training or test dataset), the algorithm produced an output 93 times, 20 of which were not visually satisfactory.

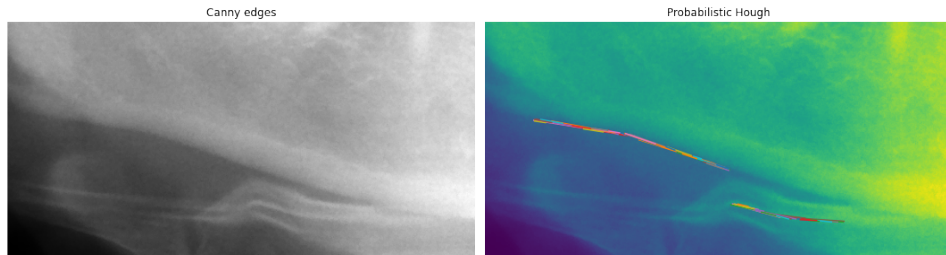


**Figure 6.10:** Figure illustrating results from algorithm 1. We observe optimistic results; the algorithm has stopped in a sweet spot. That is, just under porous textures.

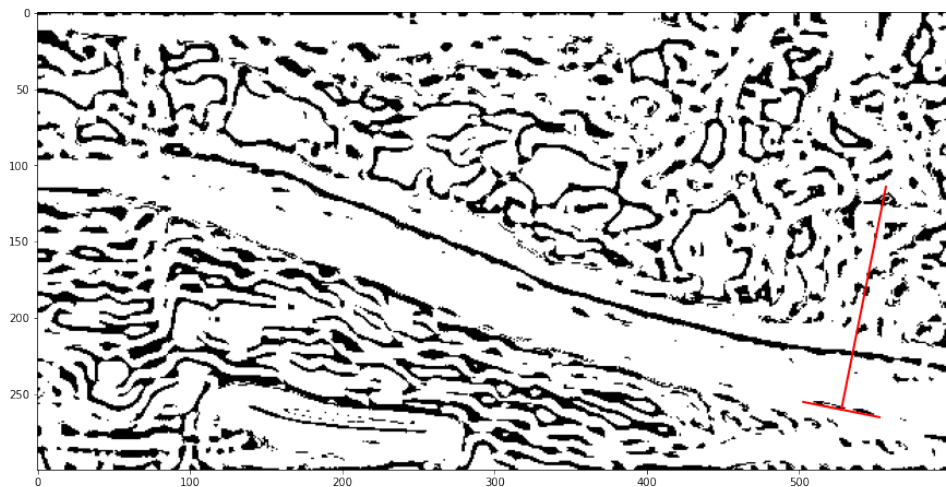
To further improve the system, steps can be taken to check if the bone's lower edge under the MF is contained in the cropped image; otherwise, the algorithm measures something else close to the MF. The best solution would be an automatic cropping procedure guaranteeing the bone's appearance. Another issue to consider is that the initial lines can become stuck in a "pit" in the binary image  $I_b$  (see algorithm 1) if the lower border of the bone is unclear. Further, and most challenging scenario, the binary image  $I_b$  can suffer from artifacts overlapping either the line's pathway when traveling toward the MF or other areas of the image. As a result, the artifacts cause an unclear upper bone



border, terminating the algorithm at an incorrect location, or the line segment suggested in the first place will suffer (see figure 6.11).



(a) Canny edges will be retrieved from the left image and fed to the probabilistic Hough transform to find the best edge candidate. However, an artifact breaks the jawline, and the segment closest to the MF here, will be wrong.



(b) The figure shows a case of a "pit" where the line segment has been initialized on the binary image  $I_b$ , that satisfy the stopping criteria (overlapping black pixels).

**Figure 6.11:** Figure depicts two cases where the measuring algorithm need improvements.

## **Part IV**

# **Conclusion**





## Conclusion and Final Thoughts

In this thesis, we have presented a method for automatically identifying the mental foramen from dental panoramic radiographs, and we have provided a proof of concept for the fully automatic estimation of the mandibular cortical width. This chapter presents a final discussion and summary of the experiments that have been performed. First, the experiments and results are discussed from a holistic perspective based on the objectives of the thesis. Then, some concluding remarks are given, and finally, proposals for future work associated with this thesis's experiments are suggested.

Results from investigating easy and complex scenarios showed agreements between the model and experts when the IoU was less than 0.4, and surprisingly when the IoU was 0. Examining when the IoU was 0 established the immense challenge of determining the mental foramen's location. It is apparent that annotating complex images is exceptionally challenging, and in worst cases, it boils down to a best guess.

When no other landmarks are present when evaluating a prediction of the MF's location, explainable AI (Adadi and Berrada, 2018) is needed to provide insight to the reason behind the predictions. Furthermore, this would allow for an uncertainty measure behind the model, which would highly benefit clinicians. Unfortunately, this could not be resolved within the project's time frame.

Furthermore, we have seen that the patient's position is not static, and artifacts in the dental panoramic radiographs can occur that, in the worst case, lead to many disregarded images. This problem should be kept in mind during x-ray examination and prevented whenever possible. In addition, artifacts constitute a challenge for algorithms that measures the mandibular cortical width. Therefore, we should also consider other possibilities for screening of osteoporosis, in particular, transfer learning to learn attributes of DPRs labeled as affected.

The thesis has studied the detection of mental foramen using deep learning on dental panoramic radiographs. Four models of different complexities have been studied; Faster R-CNN, RetinaNet, CenterNet, and EfficientDet. The first three models demonstrated relatively fair results, whereas EfficientDet-D0 was most beneficial. Different configurations of these architectures have been studied and were used for experimentation. Two experts have examined dental panoramic images from the Tromsø survey 7. Out of 3970 images, 2599 images were suited for deep learning applications. Images were further cropped into a left and right region, resulting in a dataset with 5197 usable regions with ground truth annotations.

The initial claim of the thesis suggested that existing models trained on the COCO dataset could be fine-tuned to detect the mental foramen.

The model EfficientDet indicates sufficient precision and correct predictions considering a threshold of 50% IoU, compared to other well-known models tested in this work. This conclusion is drawn from comparing average precision in tables 6.1 and 6.2. Even though the model does not predict the bounding box perfectly, it gives a helpful suggestion for an expert, which was tested from a subset of 100 easy and complex images. Furthermore, various figures have illustrated the visual and descriptive advantages of using deep learning for detecting the mental foramen. Therefore, our first claim is concluded to be true.

The second claim followed from the first, assuming the first was true. Could an object detector help accomplish an automatic measuring process of cortical width in panoramic radiographs? With previous work and the results from the first claim, it was possible to merge the two to achieve an automatic process. However, the resulting algorithm need improvements, and it is not generalized to handle images regions with high complexity yet, even though the mental foramen was found. Therefore, the algorithm was semi-capable to measure the bone from visual reports, and the second claim cannot be considered concluded.

## 7.1 Future work

This thesis has revealed some exciting aspects. However, further research is needed, and hopefully, this thesis has helped lay a foundation for future studies. Unfortunately, the time constraint has prevented the consideration of certain exciting aspects. Therefore, the following may be considered for future research:

- Expansion of the data set should be considered as it may then be sufficient to train a model from scratch.
- Implementation of a dynamic image cropping procedure based on other landmarks would ensure the bone's lowest edge presence. Therefore, more images can be measured. In addition, it might help with regularization when training a model.
- Experimentation on newer models can take place. In this thesis, EfficientDet D0 was used for inference, while EfficientDet D7 is available with almost twice the mean average precision on the COCO dataset.
- Utilize explainable AI (see Adadi and Berrada (2018)) to improve trustworthiness of the AI-system.
- Robustness of the bone measuring scheme should be improved.
- MCW measurements should be evaluated considering cases where osteopenia is present to establish if the algorithm can differentiate patients based on MCW.

This concludes the thesis.

# Bibliography

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
- Abdi, A. H., Kasaei, S., and Mehdizadeh, M. (2015). Automatic segmentation of mandible in panoramic x-ray. *Journal of Medical Imaging*, 2(4):044003.
- Adadi, A. and Berrada, M. (2018). Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160.
- Aliaga, I., Vera, V., Vera, M., García, E., Pedrera, M., and Pajares, G. (2020). Automatic computation of mandibular indices in dental panoramic radiographs for early osteoporosis detection. *Artificial intelligence in medicine*, 103:101816.
- Alpaydin, E. (2014). *Introduction to machine learning (3rd Edition)*. MIT press.
- Arifin, A. Z., Asano, A., Taguchi, A., Nakamoto, T., Ohtsuka, M., Tsuda, M., Kudo, Y., and Tanimoto, K. (2006). Computer-aided system for measuring the mandibular cortical width on dental panoramic radiographs in identifying postmenopausal women with low bone mineral density. *Osteoporosis international*, 17(5):753–759.
- Benson, B. W., Prihoda, T. J., and Glass, B. J. (1991). Variations in adult cortical bone mass as measured by a panoramic mandibular index. *Oral surgery, oral medicine, oral pathology*, 71(3):349–356.
- Bernal, M., Elenkova, M., Evensky, J., and Stein, S. H. (2018). Periodontal disease and osteoporosis-shared risk factors and potentiation of pathogenic

- mechanisms. *Current Oral Health Reports*, 5(1):26–32.
- Brownlee, J. (2019). *Deep learning for computer vision: image classification, object detection, and face recognition in python*. Machine Learning Mastery.
- Calciolari, E., Donos, N., Park, J., Petrie, A., and Mardas, N. (2015). Panoramic measures for oral bone mass in detecting osteoporosis: a systematic review and meta-analysis. *Journal of dental research*, 94(3\_suppl):17S–27S.
- Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698.
- Chandak, L. G., Lohe, V. K., Bhowate, R. R., Gandhi, K. P., Vyas, N. V., et al. (2017). Correlation of periodontitis with mandibular radiomorphometric indices, serum calcium and serum estradiol in postmenopausal women: A case-control study. *Indian journal of dental research*, 28(4):388.
- Cootes, T. F., Taylor, C. J., Cooper, D. H., and Graham, J. (1995). Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59.
- Devlin, H., Allen, P. D., Graham, J., Jacobs, R., Karayianni, K., Lindh, C., van der Stelt, P. F., Harrison, E., Adams, J., Pavitt, S., et al. (2007). Automated osteoporosis risk assessment by dentists: a new pathway to diagnosis. *Bone*, 40(4):835–842.
- Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., and Tian, Q. (2019). Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Dutta, A. and Zisserman, A. (2019). The VIA annotation software for images, audio and video. In *Proceedings of the 27th ACM International Conference on Multimedia, MM '19*, New York, NY, USA. ACM.
- Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448.
- Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning (1st Edition)*. MIT Press, Cambridge, MA.



- Greenstein, G. and Tarnow, D. (2006). The mental foramen and nerve: clinical and anatomical factors related to dental implant placement: a literature review. *Journal of periodontology*, 77(12):1933–1943.
- Hasan, T. (2012). Morphology of the mental foramen;a must know in clinical dentistry. *journal of Pakistan Dental Association*, 21:167–172.
- Hastar, E., Yilmaz, H. H., and Orhan, H. (2011). Evaluation of mental index, mandibular cortical index and panoramic mandibular index on dental panoramic radiographs in the elderly. *European journal of dentistry*, 5(01):060–067.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- Hinton, G., Srivastava, N., and Swersky, K. (2012a). Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8).
- Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012b). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.
- Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.
- Jacobs, R., Mraiwa, N., van Steenberghe, D., Sanderink, G., and Quirynen, M. (2004). Appearance of the mandibular incisive canal on panoramic radiographs. *Surgical and radiologic anatomy*, 26(4):329–333.
- Kavitha, M. S., Asano, A., Taguchi, A., and Heo, M.-S. (2013). The combination of a histogram-based clustering algorithm and support vector machine for the diagnosis of osteoporosis. *Imaging science in dentistry*, 43(3):153–161.
- Kinalski, M. A., Boscato, N., and Damian, M. F. (2020). The accuracy of panoramic radiography as a screening of bone mineral density in women: a systematic review. *Dentomaxillofacial Radiology*, 49(2):20190149.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kiryati, N., Eldar, Y., and Bruckstein, A. M. (1991). A probabilistic hough transform. *Pattern recognition*, 24(4):303–316.

- Laher, A. E., Wells, M., Motara, F., Kramer, E., Moolla, M., and Mahomed, Z. (2016). Finding the mental foramen. *Surgical and Radiologic Anatomy*, 38(4):469–476.
- Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.
- LeCun, Y. et al. (1989). Generalization and network design strategies. *Connectionism in perspective*, 19:143–155.
- Ledgerton, D., Horner, K., Devlin, H., and Worthington, H. (1999). Radiomorphometric indices of the mandible in a british female population. *Dentomaxillofacial Radiology*, 28(3):173–181.
- Lee, K.-S., Jung, S.-K., Ryu, J.-J., Shin, S.-W., and Choi, J. (2020). Evaluation of transfer learning with deep convolutional neural networks for screening osteoporosis in dental panoramic radiographs. *Journal of clinical medicine*, 9(2):392.
- Levernes, S., M., O. H., Unhjem, J. F., Øvergaard, S., Sekse, T., Wøhni, T., Hannevik, M., Paulsen, G. U., and Saxebøl, G. (2014). Radiation use in norway. useful use and good radiation protection for society, humans and the environment. *StrålevernRapport 2014:2*, 43(2).
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer.
- López López, J., Estrugo Devesa, A., Jané Salas, E., Ayuso Montero, R., and Gómez Vaquero, C. (2011). Early diagnosis of osteoporosis by means of orthopantomograms and oral x-rays: a systematic review.
- Loshchilov, I. and Hutter, F. (2016). Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.
- Naik, A., Tikhe, S., Bhide, S., Kaliyamurthie, K., and Saravanan, T. (2016). Au-

- automatic segmentation of lower jaw and mandibular bone in digital dental panoramic radiographs. *Indian Journal of Science and Technology*, 9(21):1–6.
- Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Icml*.
- Nesterov, Y. E. (1983). A method for solving the convex programming problem with convergence rate  $O(1/k^2)$ . In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547.
- Newell, A., Yang, K., and Deng, J. (2016). Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer.
- on Osteoporosis, N. C. D. P., Prevention, D., et al. (2001). Osteoporosis prevention, diagnosis, and therapy. *JAMA*, 285(6):785–795.
- Paasche Edvardsen, I. (2021). Semi-automatic estimation of mandibular cortical width in dental panoramic radiographs for early osteoporosis detection.
- Parlani, S., Nair, P., Agrawal, S., Chitumalla, R., Beohar, G., and Katar, U. (2014). Role of panoramic radiographs in the detection of osteoporosis. *Journal of Oral Hygiene & Health*, pages 2–4.
- Passos, J. S., Gomes Filho, I. S., Sarmiento, V. A., Sampaio, D. S., Gonçalves, F. P., Coelho, J. M. F., Cruz, S. S., Trindade, S. C., and Cerqueira, E. M. (2012). Women with low bone mineral density and dental panoramic radiography. *Menopause*, 19(6):704–709.
- Patel, V. L., Shortliffe, E. H., Stefanelli, M., Szolovits, P., Berthold, M. R., Bellazzi, R., and Abu-Hanna, A. (2009). The coming of age of artificial intelligence in medicine. *Artificial intelligence in medicine*, 46(1):5–17.
- Qian, N. (1999a). On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151.
- Qian, N. (1999b). On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151.
- Ramachandran, P., Zoph, B., and Le, Q. V. (2017). Searching for activation functions. *arXiv preprint arXiv:1710.05941*.
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on*

*computer vision and pattern recognition*, pages 779–788.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

Rosenhahn, B. and Andres, B. (2016). *Pattern Recognition: 38th German Conference, GCPR 2016, Hannover, Germany, September 12-15, 2016, Proceedings*, volume 9796. Springer.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Rushton, V. and Horner, K. (1996). The use of panoramic radiology in dental practice. *Journal of dentistry*, 24(3):185–201.

Santurkar, S., Tsipras, D., Ilyas, A., and Madry, A. (2018). How does batch normalization help optimization? In *Advances in Neural Information Processing Systems*, pages 2483–2493.

Schwendicke, F. a., Samek, W., and Krois, J. (2020). Artificial intelligence in dentistry: chances and challenges. *Journal of dental research*, 99(7):769–774.

Shi, S., Wang, Q., Xu, P., and Chu, X. (2016). Benchmarking state-of-the-art deep learning software tools. In *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*, pages 99–104. IEEE.

Szklo, M. and Nieto, F. J. (2014). *Epidemiology: beyond the basics*. Jones & Bartlett Publishers.

Taguchi, A., Suei, Y., Ohtsuka, M., Otani, K., Tanimoto, K., and Ohtaki, M. (1996). Usefulness of panoramic radiography in the diagnosis of postmenopausal osteoporosis in women. width and morphology of inferior cortex of the mandible. *Dentomaxillofacial Radiology*, 25(5):263–267.

Taguchi, A., Tsuda, M., Ohtsuka, M., Kodama, I., Sanada, M., Nakamoto, T., Inagaki, K., Noguchi, T., Kudo, Y., Suei, Y., et al. (2006). Use of dental panoramic radiographs in identifying younger postmenopausal women with

- osteoporosis. *Osteoporosis international*, 17(3):387–394.
- Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR.
- Tan, M., Pang, R., and Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790.
- Tofangchiha, M., Khorasani, M., Shokrimozhdehi, M., and Javadi, A. (2017). Diagnosis of osteoporosis using cortex mandibular indices based on cortex thickness and morphology in comparison with visual assessment of the cortex. *Journal of Craniomaxillofacial Research*, pages 345–351.
- Uijlings, J. R., Van De Sande, K. E., Gevers, T., and Smeulders, A. W. (2013). Selective search for object recognition. *International journal of computer vision*, 104(2):154–171.
- Vlasiadis, K. Z., Skouteris, C. A., Velegrakis, G. A., Fragouli, I., Neratzoulakis, J. M., Damilakis, J., and Koumantakis, E. E. (2007). Mandibular radiomorphometric measurements as indicators of possible osteoporosis in postmenopausal women. *Maturitas*, 58(3):226–235.
- Wen, W., Wu, C., Wang, Y., Chen, Y., and Li, H. (2016). Learning structured sparsity in deep neural networks. In *Advances in neural information processing systems*, pages 2074–2082.
- White, S. C. and Pharoah, M. J. (2014). *Oral radiology-E-Book: Principles and interpretation*. Elsevier Health Sciences.

## **Part V**

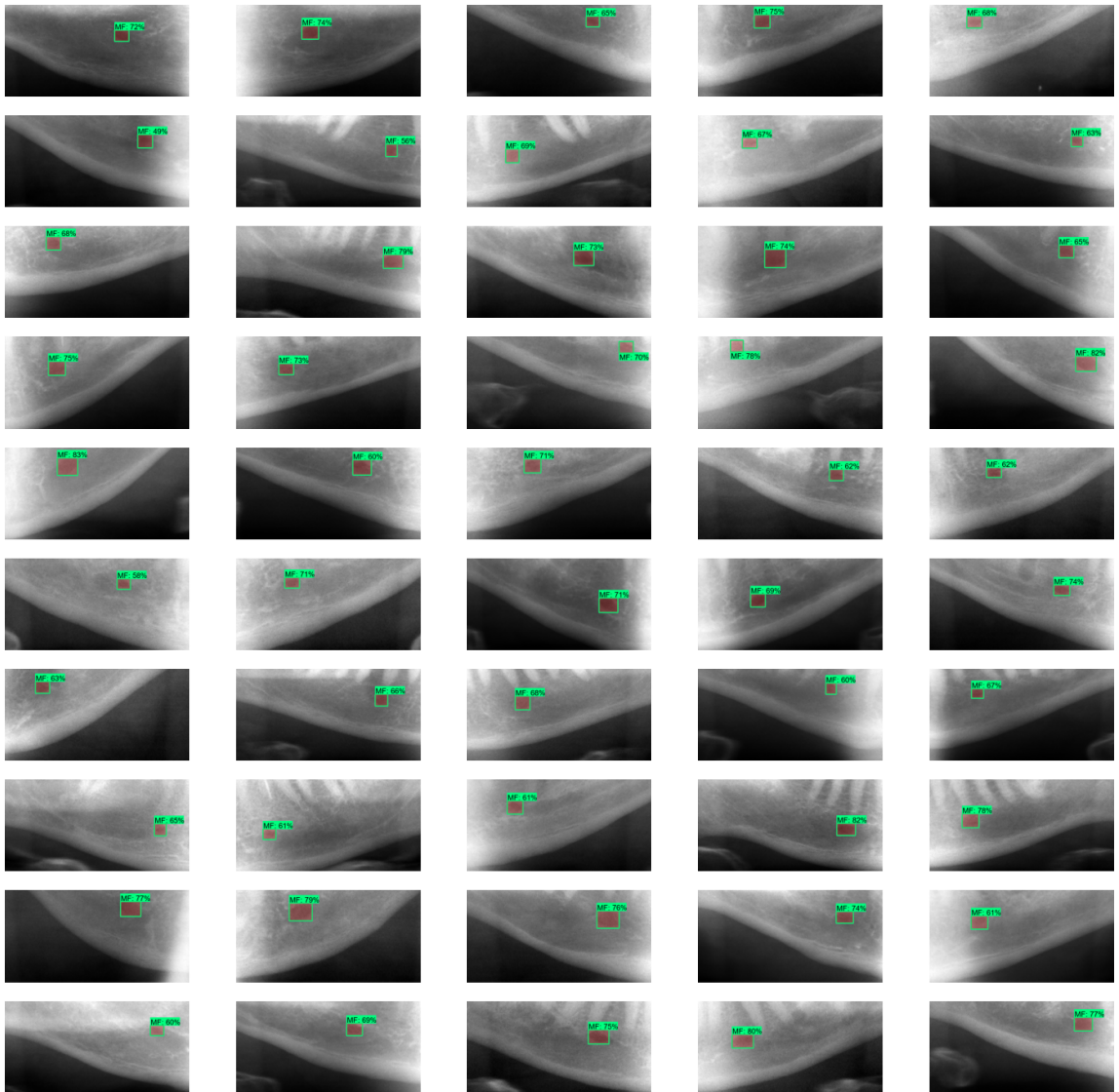
# **Appendix**



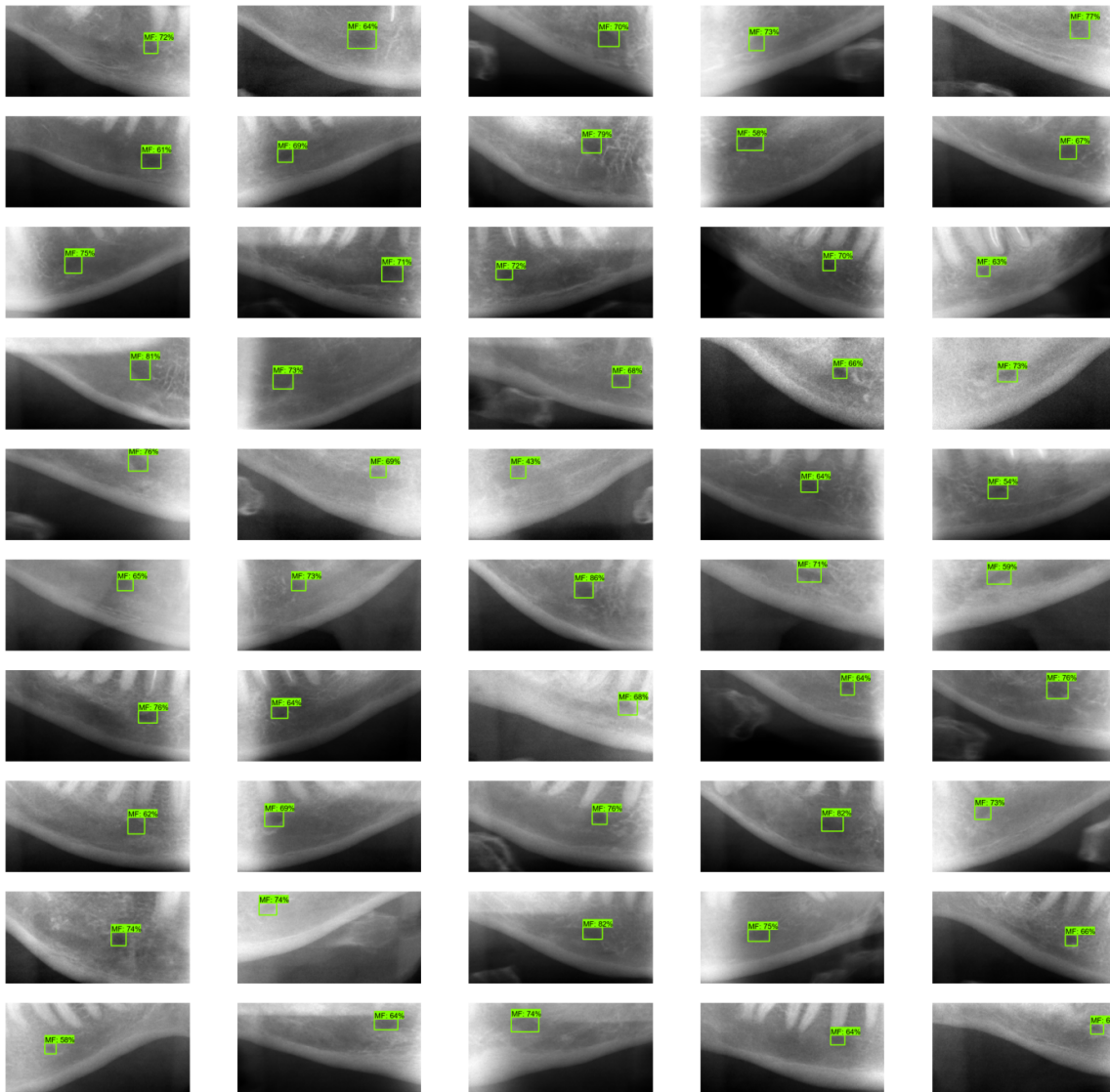
# Appendix



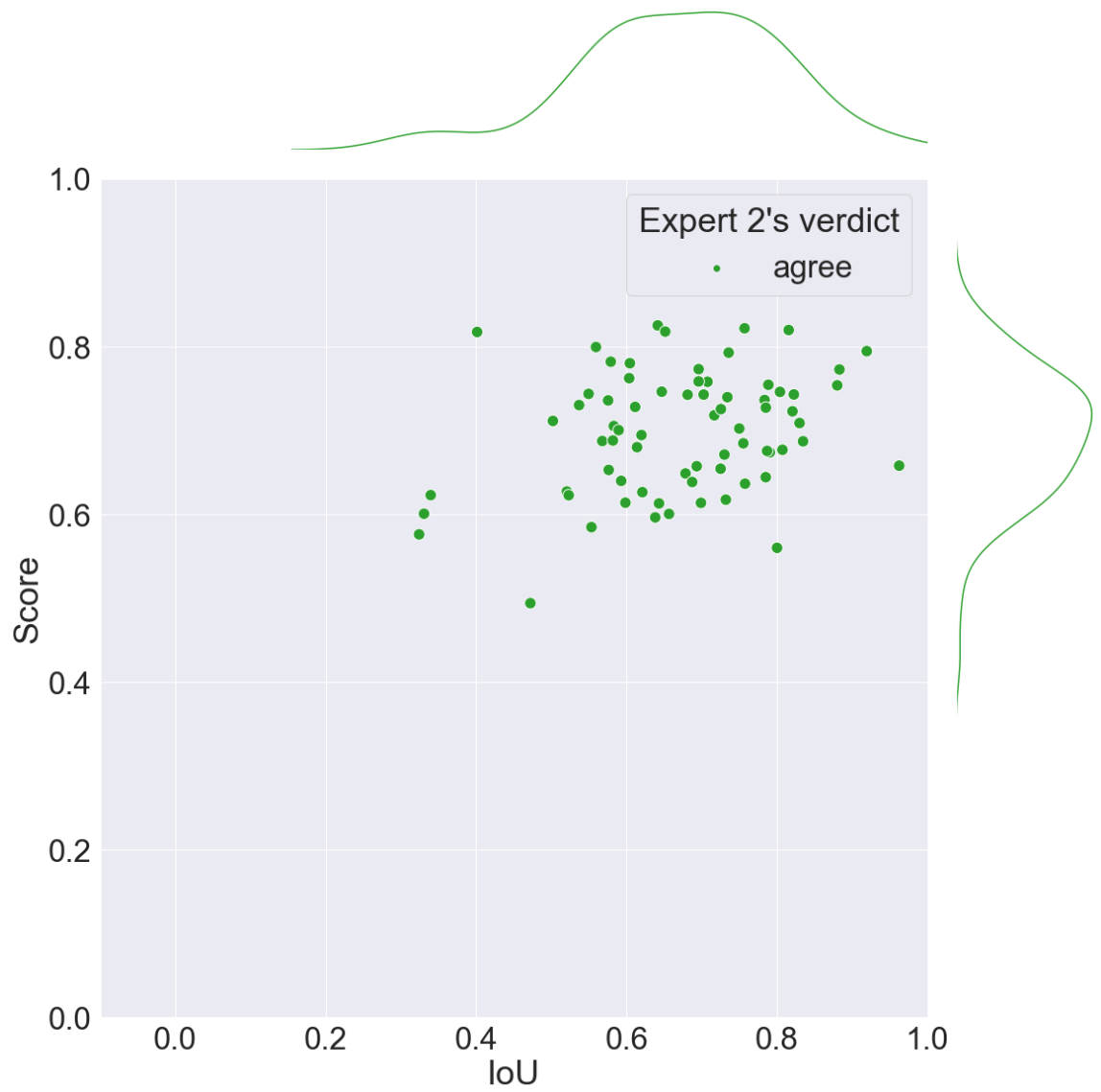
## 7.2 Extra Figures



**Figure 7.1:** Figure visualizing 50 out of 100 predictions from EfficientDet D0 (in green) on the easy images with ground truth filled with red.



**Figure 7.2:** Figure visualizing 50 out of 100 predictions from EfficientDet D0 on easy images (in green) without ground truths filled in red. It is clear that the object detector does a good job in this scenario.



**Figure 7.3:** Figure visualizing prediction score vs. IoU. Expert 2 has manually inspected the results and indicated whether they agree or not with the predicted results.



