University of
Stavanger

FACULTY OF SCIENCE AND TECHNOLOGY

MASTER'S THESIS

Study program/Specialization: Information Technology – Robotics and Signal Processing	Spring semester, 2021 Open
Author: Emil Obrestad	............Emil Obestad.......... (Signature of author)
Programme coordinator: Ketil Oppedal Supervisor(s): Ketil Oppedal and Álvaro Fernández Quílez	
Title of master's thesis: Prostate Lesion Detection on Apparent Diffusion Coefficient MRI based on Convolutional Neural Networks	
Credits: 30	
Keywords: Convolutional Neural Network, Deep Learning, Object Detection, Image Processing, Biomedical Image, Prostate Cancer, Supervised Learning	Number of pages: 73 + supplemental material/other: 30 Stavanger, 29 June 2021 Date/year

Faculty of Science and Technology
Department of Electrical Engineering and Computer Science

Prostate Lesion Detection on Apparent Diffusion Coefficient MRI based on Convolutional Neural Networks

Master's Thesis in Robotics and Signal processing
by

Emil Obrestad

Supervisors
Ketil Oppedal
Álvaro Fernández Quílez

June 29, 2021

Abstract

In 2020 there were 1 414 259 new incidences and 375 304 deaths worldwide caused by prostate cancer. The number of cases could increase even further in the future due to higher life expectancy and population growth. Prostate cancer diagnosis consists of several examination steps that are time-consuming, expensive and can involve risk factors. Magnetic resonance imaging can help locate and classify prostate cancer at an early stage, but it suffers from inter-observer variability. Utilizing a single, relative, automated object detector for prostate diagnosing can produce a more comfortable, efficient and less expensive examination process.

This thesis explores the paradigm of supervised learning, and more specifically supervised object detection, on apparent diffusion coefficient images for prostate lesion, with one-stage and two-stage convolutional neural networks architectures. Image pre-processing techniques to increase bounding box area size and data augmentation to alleviate the shortage of data are investigated to improve network performance. Evaluation of detection performance relative to the prostate anatomical zone is conducted. Different lesion classification approaches were conducted to explore the networks ability to classify lesions. The data set used in this thesis consists of 1109 images with 1281 labelled ground truths that have an uneven distribution of examples between the lesion classes. There are instances of lesion ground truth errors, which could diminish the object detector performance.

An average precision of 0.424 was achieved for clinically significant lesions and 0.156 for insignificant lesions, where the network detector produced the most promising results for lesions located in the prostate transition zone. However, the inefficient data set size and possible lesion ground truth errors limit the network to obtain optimal performance results. Data augmentation improved network performance by artificially increasing the data set size. Experiments conducted showed that convolutional neural network architectures have a problem detecting small objects. Cropping and resizing images increased the bounding box dimensions, which improved detection performance. Object detection shows a great potential to be used in hospitals for prostate cancer diagnosis, which could be an influential tool for reducing over-diagnosing.

Acknowledgements

This thesis marks the end of my Master's Degree in Robotics and Signal Processing at the Department of Electrical Engineering and Computer Science at the University of Stavanger.

I want to thank my supervisors Ketil Oppedal and Álvaro Fernández Quílez, for their advice and strong guidance during my last semester. Providing both insights on machine learning and motivation for this thesis. I also want to thank Rune Wetteland for guidance and an excellent lecture on using the UNIX system. Renato Cuocolo also deserves recognition for providing the prostate lesion mask and valuable feedback. Without his work, this thesis would not have been possible.

I would like to give special thanks to my fellow students for two tremendous and memorable years at the Micro-lab and the ISI room. Finally, I want to thank the student organization ISI and my fellow ISI board members for giving me a fun and meaningful time at the University of Stavanger.

Contents

Abbreviations

ADC Apparent Diffusion Coefficient

AFS Anterior Fibromuscular Stroma

AP Average Precision

AR Average Recall

BB Bounding Box

CNN Convolutional Neural Networks

DRE Digital Rectal Examination

GGG Gleason Grade Group

IoU Intersection over Union

MRI Magnetic Resonance Imaging

mAP Mean Average Precision

mAR Mean Average Recall

NN Neural Networks

NIfTI Neuroimaging Informatics Technology Initiative

PCa Prostate Cancer

PSA Prostate-Specific Antigen

PZ Peripheral Zone

R-FCN Region based Fully Convolutional Networks

SSD Single Shot Detector

TZ Transition Zone

Chapter 1

Introduction

1.1 Motivation

Prostate cancer (PCa) is the second most frequently occurring cancer among men and the fourth most commonly occurring cancer overall [1]. In 2020 alone, there were 1 414 259 new incidences and 375 304 deaths worldwide caused by PCa [1].

PCa diagnosing consists of several examination methods that are both time-consuming and expensive [2]. A general practitioner performs first-stage testing with a constraint accessibility to proper medical tools. These tests are unreliable that can fail to detect PCas or even lead to PCa overdiagnosis, that again can result in an over-treatment. Inaccurate test results can engender the patient for unnecessary apprehension or a false sense of safeness. Determining PCa aggressiveness through biopsy involves risk factors, including hemorrhage and infection [3]. Today, magnetic resonance imaging (MRI) assessment is unreliable because of inter-observer variability, where PCa diagnosing might vary depending on the reader. Moreover, patients with underlying health problems are in some cases not recommended undergoing PCa screening if the risk following examination transcend the benefits [4, 5].

Utilizing computer vision in MRI for PCa diagnosing can help improve lesion localization and classification. The outcome can eliminate further diagnosing for patients with insignificant PCa and allow specialists to mainly focus on patients with clinically significant PCa, reducing PCa overdiagnosis. An advanced object detector can potentially classify the PCa relative to the Gleason Grade Group (GGG), terminate prostate biopsy test and the following risk associated with it, thus making biopsies an unnecessary practice or just transforming them into a support test in cases where it is necessary to confirm the diagnostic. Depending on a single, relative, automated MRI screening test would make the examination process more efficient, less expensive and less stressful for the patient.

1.2 Problem Definition

This thesis aims to detect and predict the clinical significance of prostate lesions found in apparent diffusion coefficient (ADC) MRIs. Today, there exist a variety of object detection Convolutional Neural Network (CNN) architectures, where this thesis will explore some of the meta-architecture and backbone networks. This thesis proposes methods to increase CNN performance for lesion detection in ADC images. The available data material consists of 200 patients containing ground truth data for 299 individual lesion objects, distributed on a total of 1109 two-dimensional ADC images. This approach of using labeled data when training CNN is known as supervised learning, later discussed in Section 3.4.

1.2.1 Proposed Method Overview

There are numerous methods and approaches for lesion detection. Some focus on performance, others focus more on detection speed, but most applications focus on optimizing both. The primary objective of this thesis is to explore different supervised CNN architecture, with one-stage and two-stage object detectors. Architecture, image pre-procession implementation and hyperparameters need to be adjusted based on the data set. This thesis will investigate different image pre-processing techniques and methods to alleviate the shortage of data, such as data augmentation, to incease CNN performance. Evaluation of CNN performance relative to prostate anatomical zone will be conducted since anatomical zone seems to be a relevant parameter for lesion location in PCa [6, 7]. Finally, this thesis will examine model performance utilizing different lesion classification approaches. Figure 1.1 shows an outline of the proposed method for this thesis.

1.2 Problem Definition

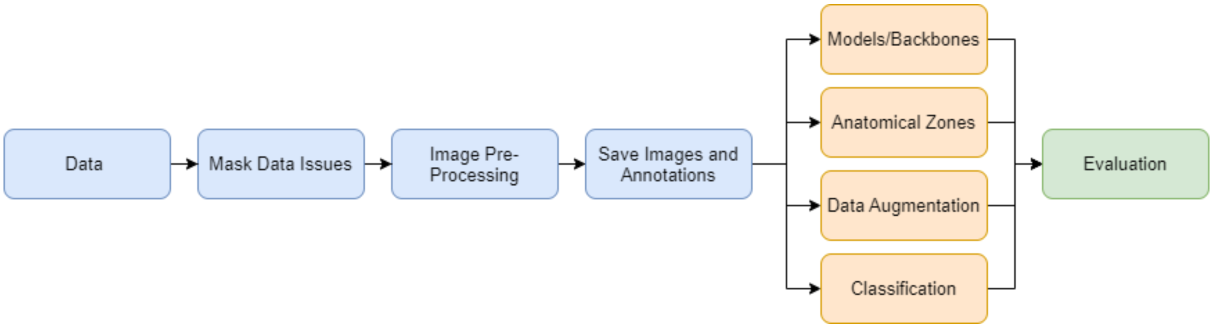


Figure 1.1: Overview of the thesis methodology.

1.3 Thesis Structure

The thesis is structured into nine chapters, where the outline is presented in the following description.

- Chapter 1: Introduction
  - Motivation, problem definition, objectives and previous work.
- Chapter 2: Medical Background Theory
  - Medical background necessary to understand the biological point of view of this thesis.
- Chapter 3: Technical Background Theory
  -Technically background necessary to understand the various methods utilized in this thesis.
- Chapter 4: Material and Methods
  - Presentation of the data set used in this thesis. Also, explaining different methods and image pre-processing used to improve detection performance.
- Chapter 5: Configuration
  - Goes through proposed hyperparameters for improving CNN architecture performance.
- Chapter 6: Experimental Results
  - Presents experiment results for different implementations and methods.
- Chapter 7: Discussion
  - Discuss results from the proposed methods and the constraints for this thesis.
- Chapter 8: Conclusion and Future Directions
  - Presents conclusion and discuss future directions for this thesis.

Chapter 2

Medical Background

2.1 Prostate Cancer

Prostate cancer (PCa) is the second most frequently occurring cancer among men and the fourth most commonly occurring cancer overall [1]. In 2020 alone, there were 1 414 259 new incidence and 375 304 deaths worldwide [1]. That is 7.3% of all incidence and 3.8% of all mortality related to cancer.

Usually, human cells grow and divide, producing new cells to replace dying cells. Cancer results from normal cells that become abnormal, where damaged cells still live even though they should die and new cells form even though they are not needed. This can lead to a growth called cancerous tumor which can expand into different parts of the human body. The most common type of PCa is the adenocacinomas, which develop from the gland cells. Small cell carcinomas, neuroendocrine tumors, transitional cell carcinomas and sarcomas are other, rare, types of cancer that also develop in the prostate [8].

There are four fundamental anatomical zones of the prostate, which are relevant for PCa, see figure 2.1 [9]. These are the peripheral zone (PZ), the transition zone (TZ), the anterior fibromuscular stroma (AFS) and the central zone (CZ). The most common zone in which PCa is commonly developed from is the PZ (70-75%) and the second most common zone is the TZ (25%) [7]. The AFS and the CZ are both unusual, but not improbable, zones for PCa to originate from.

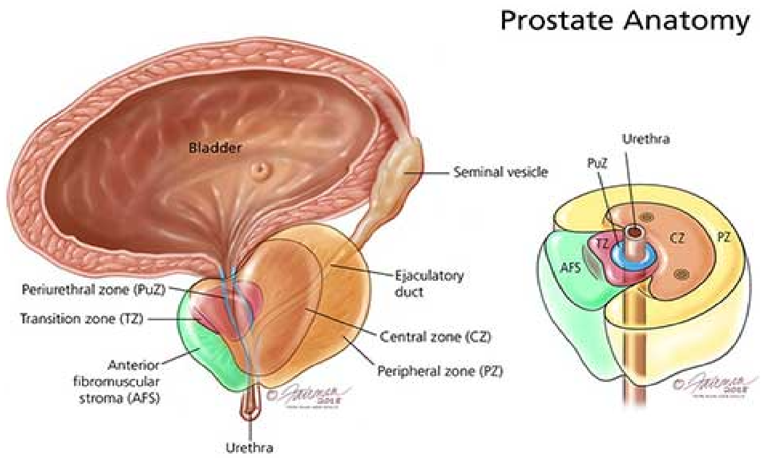


Figure 2.1: Figure shows anatomical zones of the prostate [9].

2.2 Examination Methods

This section discusses existing examination methods used to determine whether a patient have PCa and how the PCa aggressiveness is graded. Benefits and risk associated with PCa screening needs to be discussed between the doctor and the patient before going through possible tests and treatments [5].

2.2.1 Prostate Specific Antigen Test

Prostate specific antigen (PSA) is a substance that is excreted in small amounts from the prostate gland and released into the semen and the bloodstream, that is measured in nanograms per milliliter (ng/mL) units [10]. Higher levels of PSA can indicate PCa, benign prostate enlargement, infections or urinary tract.

The PSA level will normally increase as men get older. A commen PSA cutoff point of 4 ng/mL is often used when deciding if a patient needs further testing. Today, PSA test is an unreliable test for both early detection and for ruling out PCa, where it is one of the main causes of PCa overdiagnosis [11, 12]. However, the PSA test can be practical to monitor PCa development and to follow the effect of a possible treatment [10].

2.2.2 Digital Rectal Exam

Another early stage PCa screening test is the digital rectal exam (DRE), most commonly done after a PSA test. DRE can in some cases detect PCa in men with normal PSA blood level, thus it is worth including in the PCa examination procedure. During a DRE the doctor inserts a gloved lubricated finger into the patient rectum and examine the size, shape and texture of the prostate gland. If there is any sign to abnormalities the patient will be referred to a hospital, or an urologist, for further testing [13].

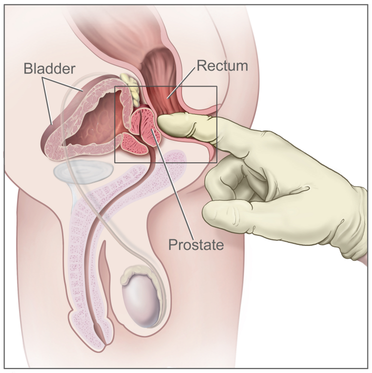


Figure 2.2: Illustration of a digital rectal exam (DRE) performed on a patient [14].

2.2.3 Prostate Biopsy

If the doctor is suspicious from DRE or PSA testing, or the patients have any warning symptoms, the patient is referred to an urologists to take image examination and biopsy from the prostate gland. This procedure is known as the transrectal ultrasound (TRUS) guided biopsy [15]. A thin, hollow needle is inserted into the prostate gland eight to ten times in order to obtain a composite examination [10]. The needle pulls out a small tissue sample which is later examined under a microscope. An ultrasound transducer is inserted together with the needle, to help localize the prostate by sending sound waves and computing the resulting echos into digital images [15]. Preventive antibiotics are given before the examination to prevent serious infection that can occur after a prostate biopsy procedure [3].

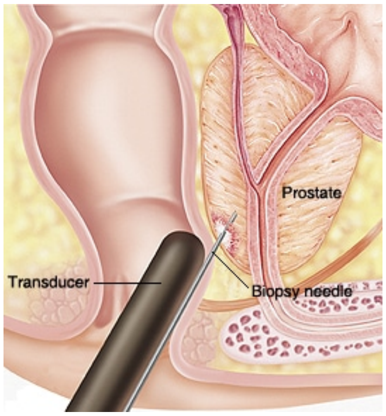


Figure 2.3: Illustration of how a prostate biopsy is performed through the rectum [16].

2.2.4 Magnetic Resonance Imaging

Magnetic resonance imaging (MRI) can help locate abnormal areas in the prostate gland and indicate where tissue samples should be collected from when performing biopsy. From the MRI images, the radiologist can decide whether further examination, such as biopsy, needs to be taken at the given time. The radiologist predicts the probability of a lesion to be clinically significant based on the findings from multiparametric magnetic resonance imaging (mpMRI) [7]. The scoring is based on the T2-weighted (T2W), diffusion weighted imaging/apparent diffusion Coefficient (DWI/ADC), and the dynamic contrast enhancement (DCE) sequences. Each detected lesion is classified using the Prostate Imaging Reporting and Data System (PI-RADS) scoring system. PI-RADS v2.1 (2019) is the latest updated version [17]. The PI-RADS score depends on whether the lesion is located in the peripheral zone (PZ) or transition zone (TZ). Lesions located in the PZ are mainly determined by the DWI/ADC, and lesions located in the TZ are mainly determined by the T2W, to designate PI-RADS category scores [7].

2.2.5 Gleason Grade Group

In 1966 pathologist Donald Floyd Gleason introduced the first pathologically based scoring system for PCa [18], with a score ranging from 1 to 5 deepening on the cell pattern. In 2014 International Society of Urological Pathology (ISUP) introduced an updated grading system [19], with five Gleason grade group (GGG) scores, to simplify the PCa grading prognosis. Table 2.1 shows the Gleason score grading system and which scores are classified as clinically significant.

GGG	Gleason Score	Clinically Significant
Grade Group 1	Gleason Score $\leq$ 6	False
Grade Group 2	Gleason Score 7 (3+4)	True
Grade Group 3	Gleason Score 7 (4+3)	True
Grade Group 4	Gleason Score 8	True
Grade Group 5	Gleason Score 9 and 10	True

Table 2.1: Gleason score and ISUP-Grading [7]

Tissue samples collected from the prostate biopsy are separately studied under a microscope for a deeper understanding of the aggressiveness of the PCa [20]. Cancer cells are assigned a Gleason score depending on its pattern. Figure 2.4 illustrates how grading score are assigned to different pattern appearances. The GGG consist of the two most prevalent patterns, the primary and the secondary pattern, and are summed to yield the GGG class score, see table 2.1 [7]. GGG 1 have features similar to normal tissue samples. Both GGG 4 and 5 have pattern features indicating presence of aggressive cancer cells. Intermediate grading score, GGG 2 and 3, falls in between the highest and the lowest ranking score.

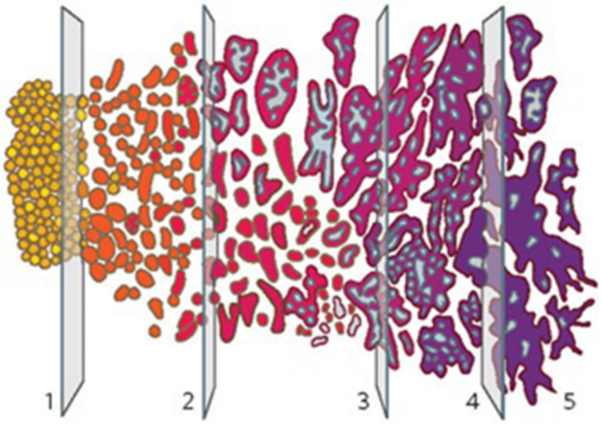


Figure 2.4: Illustration of GGG score for different cancer cell patterns [20]

Chapter 3

Technical Background Theory

This chapter takes a look at the technical background of methods used in this thesis. Object detection architectures and backbones modules are introduced and explained.

3.1 Magnetic Resonance Imaging

Magnetic Resonance Imaging (MRI) is a medical image technology used to form detailed three-dimensional anatomical images [21]. The MRI utilizes a strong magnetic field, magnetic field gradients, and radio waves to capture visualized images of organs in the body. During an MRI, a person lies on a table placed inside the MRI scan machine that generates a strong magnetic field. The magnetic field will align protons inside the body, and when additionally applying a radiofrequency pulse through the body will pull the protons against the magnetic field. However, turning off the radiofrequency currency will realign protons with the magnetic field, which in the process releases energy that the MRI sensor can pick up [21].

3.1.1 Apparent Diffusion Coefficient

Diffusion MR images measure the magnitude of diffusion of water molecules in biological tissues that come in both diffusion-weighted images (DWI) and apparent diffusion coefficient (ADC) forms [22] [23]. These MRI images are often used for acute cerebral stroke and tumours diagnosis [24]. ADC consists of multiple conventional DWI images of different weighted gradient amplitudes, which produce diffusion equivalent to the signal diversity [23]. B-value projects diffusion weighting applied to ADC mapping, thereby indicating the intensity and time of applied gradients. Choice of b-value parameters depends on the organ and its matter structure [23]. Figure 3.1 displays both DWI and ADC images of the prostate gland. The ADC image has a more detailed representation of the prostate gland relative to the DWI image.

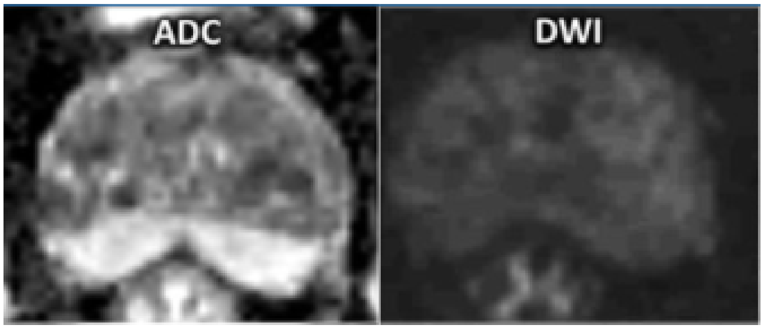


Figure 3.1: Illustration of the ADC and DWI MRI forms that measure diffusion of water molecules in tissues [23]

3.2 Neural Networks

Neural Networks (NN) dates back to 1943, when Warren Sturgis McCulloch and Walter Pitts developed an elementary NN model using electrical circuits [25]. NN is inspired by the human brain nervous system, thus are named neural. A NN contains a large composition of simple neurons, also referred to as nodes or units, that react to an input signal before transmitting an output signal [26]. Figure 3.2 illustrates a feedfoward NN with an input layer, two hidden layers and an output layer.

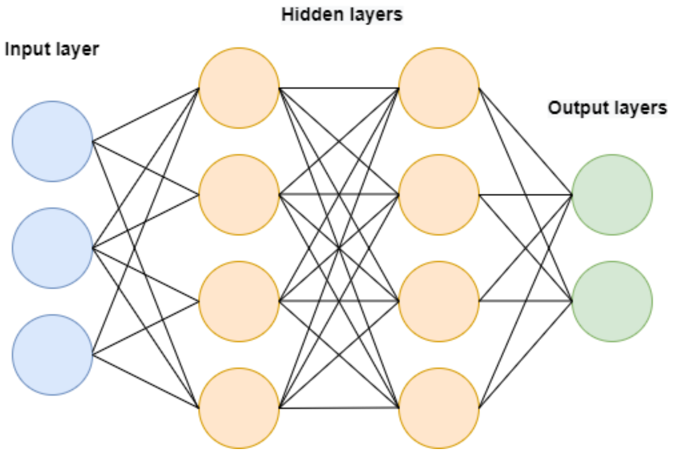


Figure 3.2: Illustration of NN with an input layer, two hidden layers and an output layer.

3.3 Convolutional Neural Networks

Convolutional neural networks (CNN) is a class of deep NN that are popular within the image processing field, with a purpose to derive meaningful patterns from digital images [26].

Convolutional Layer outputs a feature map vector, which is proceeded on to the next layer. The first convolutional layer often detects basic feature shapes, and as the convolutional layer gets deeper in the networks it focuses on extracting more specific complex feature details [26]. Pooling Layer is commonly applied after a convolutional layer to reduce feature maps dimension, thus reducing computational consumption. Fully connected layer connects every activation from the previous layer to produce the final classification output, see Figure 3.2. CNN looks for patterns in regions of the image instead of each pixel, which reduce computational expenses. Also, CNN is translation invariant in such that the object location does not matter [27].

3.4 Supervised, Unsupervised and Semi-supervised Learning

Supervised learning is the process of algorithm learning with labelled data. CNN learns the mapping function from the known input to the known output. The CNN mapping function improves the accuracy by predicting the output of the training data and then learn through backpropagation based on the corrections from labelled data [28]. Supervised learning can be grouped into two types of problems; classification and regression. Classification is a method that assigns data into a category. For example, classifying lesion tumours to be either clinically significant or insignificant. Regression uses an algorithm to predict numerical values by modelling the relationship between dependent and independent variables [29]. CNN performance is measured based on the predicted outcome of a data set that has not been used in training, often termed as test data.

Unlike supervised learning, unsupervised learning trains CNN on unlabeled data, where correct answer are not assigned to the data set. The aspiration for this algorithm learning technique is to discover patterns in the data, and it is associated with tackling two main problems; clustering for grouping data and association to seek relationships between variables in the data [28].

Semi-supervised learning is categorized somewhere between supervised and semi-supervised learning. The CNN is training with both labelled and unlabeled data, often with more unlabeled than labelled data. One of the most critical problems with machine learning is to have enough training data. Labelling data is time-consuming and in many cases impractical, considering the rough rule of thumb in supervised learning is to have more than 10 million labelled examples to exceed human performance [30].

3.5 Object Detection Networks

This section takes a deeper look at three different state-of-art object detection architectures and backbone networks used to detect prostate lesions in this thesis. Before going further into details about the models, this section will explain the essential tasks of an object detector.

3.5.1 Object Detection

Object detection, also referred to as object recognition, describes a collection of computer vision techniques that aims to locate and classify objects in a digital image [31]. Today, object detection is used in a variety of real applications, such as autonomous driving, video surveillance, mobile application, and robot vision [32]. The task of object detection can be divided into two main functions:

- **Object localization** locate objects by drawing a rectangular bounding box (BB) around its predicted boundaries.
- **Image classification** refers to the task of assigning a class label to an object. One of the most common ways to display the classification in object detection is to print the class label together with the BB in the digital image.

Object segmentation is a computer vision extension in object detection, see figure 3.3. Instead of drawing a BB based on the object outer edges, object segmentation points out every pixel in an image that contains an object [33]. Object detection architectures that implement object segmentation, such as MASK-RCNN, has not been utilized in this thesis [34].

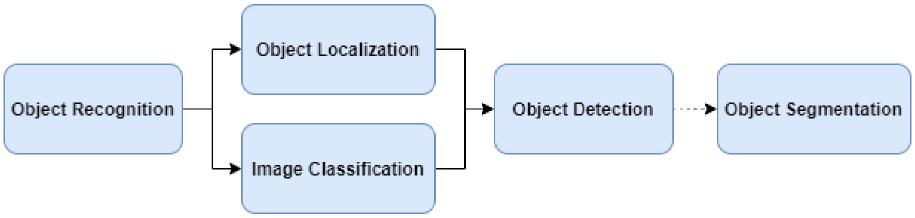


Figure 3.3: Flowchart of computer vision functions associated with object detection [33]

3.5.2 Network Models

This section introduces the meta object detection and backbone networks utilized in this project. The Tensorflow Object Detection API 1x repertory [35] provide an open-source code for the following architectures.

Single Shot Detector

Wei Liu et al. introduced the Single Shot Detector (SSD) architecture in 2016 that contains a single deep sub-network for object localization and classification [36]. SDD increases prediction speed by eliminating a second stage BB proposal and compress all computation into a single network. VGG-16 is initially used as a backbone network in the *SSD: Single Shot Multibox Detector* paper but it is also applicable for other backbones such as the ResNet-50 network [36].

SSD utilizes predefined default boxes for different sizes and aspect ratios for multi-scale feature maps, similar to the anchor boxes utilized in Faster R-CNN. The CNN network applies multiple scales of convolutions feature layers that allow detection at different scales, where initial convolutional layers cover smaller fields that exploit small object areas and the deeper layers cover wider areas that benefit large object areas. Figure 3.4 illustrates the SSD architecture. Feature layers produce a collection of BB predictions, where a Non-Maximum Suppression (NMS) step (Section 3.5.4) filters the overlapping BB to produce the final output. A weighted sum of the smooth localization loss and softmax confidence loss generate the overall model loss [36].

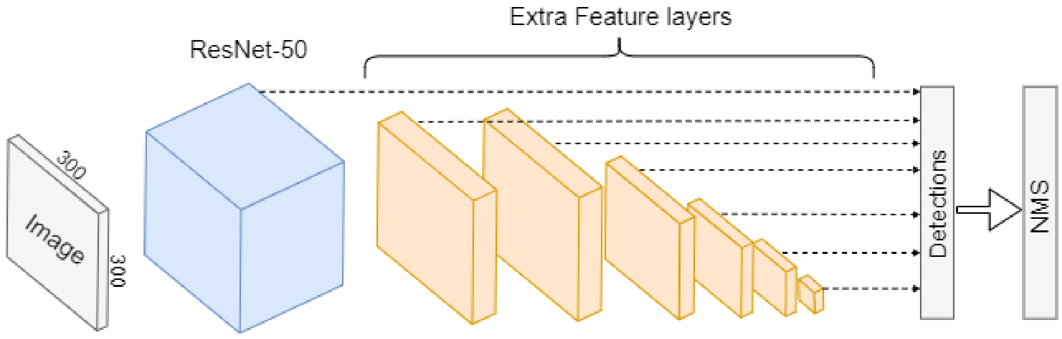


Figure 3.4: Illustration of the SSD architecture [36]

Faster R-CNN

The Faster R-CNN model architecture, developed by Shaoqing Ren et al. is an improved version of the earlier networks R-CNN and Fast R-CNN [37, 38, 39].

Faster R-CNN composes two independent trainable sub-networks; a detection network (Fast R-CNN) and a Region Proposal Network (RPN). A two-stage network obtains higher performance accuracy than a straightforward one-stage network (such as SSD) but has a higher detection latency.

Faster R-CNN produces a convolutional feature map using a backbone network that passes to the RPN, that indicates where the Fast R-CNN network should look for objects in the given image. RPN produce a $n \times n$ spatial window slide over the feature map that predicts region proposal at every spatial location by utilizing predefined anchor boxes of three scales and three ratios. Each sliding window has a total of 9 predefined anchors that is possible region of interests. The NMS step filters the overlapping predictions from the RPN, further explained in Section 3.5.4. The ROI polling layer takes the output from the NMS along with a fully connected layer and extracts a feature vector of length 256 for each of the $n \times n$ proposed region [40]. Two fully connected layers generate an objectness score based on the classifier and a regression score based on the BB coordinates. Figure 3.5 illustrates the Faster R-CNN architecture.

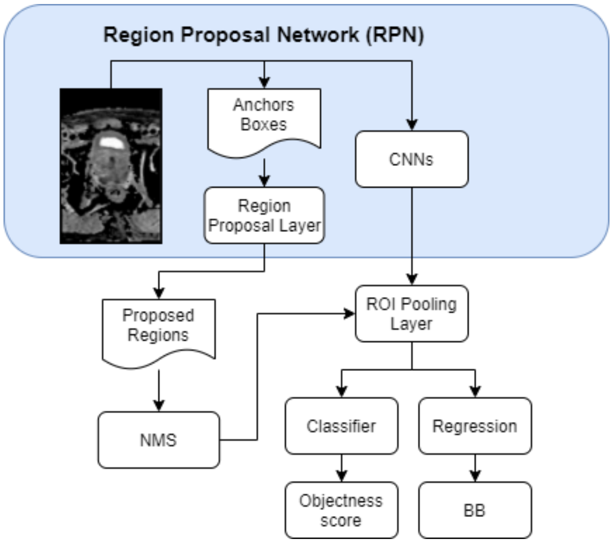


Figure 3.5: Faster R-CNN architecture [41]

Region-based Fully Convolutional Networks

Jifeng Dai et al. introduced the efficient Region-based Fully Convolutional Networks (R-FCN) architecture object detection in 2016 [42]. RFCN consists of region proposal and classification sub-networks, similar to the R-CNN networks, obtaining competitive performance with less latency relative to the Faster R-CNN architecture. The *R-FCN: Object Detection via Region-based Fully Convolutional Networks* paper discuss that object detection networks rely on localization representation that is translation-variant. R-FCN implement a position-sensitive cropping mechanism before the region of interests (ROI) to generate score maps, which decrease per-region computation [42, 35].

R-FCN executes a final ROI layer that uses selective pooling on the score maps to produce a spatial grid score of each ROI. The position-sensitive score maps obtained from the last convolutional layer is expressed as $k^2(C+1)$, see Figure 3.6 [42]. Where k^2 represent the spatial grid relative to the positions, C the object categories and 1 the background category. NMS filters prediction proposals to produce the final output, see Section 3.5.4. The R-FCN process helps tackle the location variance problem in the region proposal, producing faster detection with minor performance reduction.

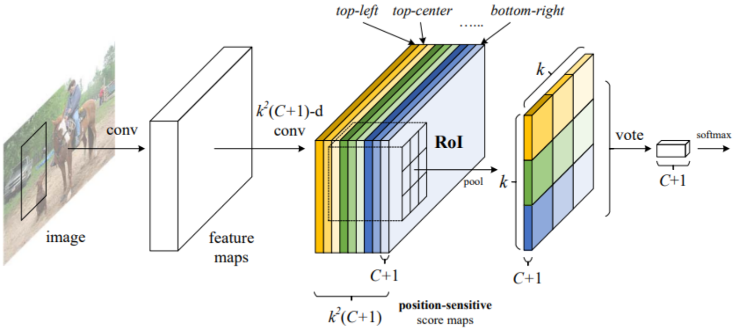


Figure 3.6: R-FCN architecture with a spatial grid ($k \times k$) equal to 3×3 [42]

3.5.3 Backbones

Bacbone network, also know as convolutional feature extractor, is applied to the CNN object detection architecture to obtain highlevel features from the input image [35]. This section present and explain the implemented bacbone networks for this thesis.

MobileNets

MobileNets module were proposed by Andrew Howard et al. and focus on reducing network parameters, computational complexity and achieving high-speed inference suitable for mobile applications [43]. MobilNets achieved VGG-16 lever performance on ImageNet harnessing only 3.33% of the VGG-16 computational and network complexity [44, 35].

MobilNets builds on depthwise separable convolutions that divide filter depth and spatial dimension. The depthwise separable convolution splits a layer into a layer that filter (depthwise convolution) and a layer that combines (pointwise convolution). Depthwise convolution places a single kernel on each of the input channels. A $1x1$ pointwise convolution combines output from depthwise convolutions and generate new features. Depthhwise separable convolutions produce a computational operation cost of:

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot DF \tag{3.1}$$

Function 3.1 expresses the number of input channels (M), number of output channels (N), kernel size (D_K) and feature map size (D_F). Depthwise separable convolutions reduce model size, latency and computational cost. However, the negative repercussion to this application is a minor performance reduction [43]. Figure 3.7 illustrates a five input channel depthwise separable convolution.

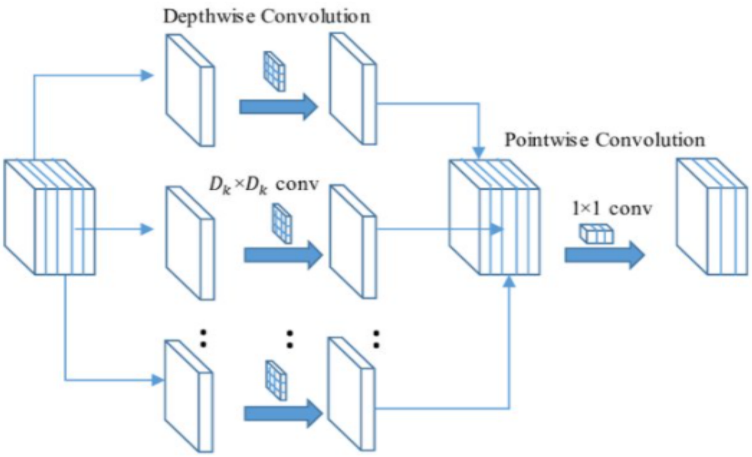


Figure 3.7: Five input channel depthwise separable convolution [45].

Residual Network

The deep residual learning framework Residual Network (ResNet), proposed by Kaming He et al. in 2016, aims to add more network layers to achieve higher performance [46]. ResNet produced a top-5 error rate of 3.57% on the ImageNet test set that gave it first place on the ILSVRC 2015 classification competition [44]. The *Deep Residual Learning for Image Recognition* paper discussed that deeper networks struggles with vanishing gradients, accuracy saturation and degradation because of optimization problems, which culminate in a performance reduction [46]. This led to the introduction of the deep residual learning framework that utilizes feedforward CNN, see Figure 3.8.

ResNet exploits a new mapping function $H(x) = F(x)+x$, where $F(x)$ represents the mapping of non-linear layers and x the identity function. This is different from the direct mapping $F(x)$ formally used in networks modules. ResNet focus on an easier way of realizing identity mapping $H(x) = x$. Pushing the residual $F(x)$ to zero is more accessible than fitting identity mapping. The Residual connection allows for a better optimization process that will reduce the degradation problem, vanishing gradient problem and allow training of really deep networks with high performance results. There are a variety of ResNet framework versions with different depths and procedures. ResNet-18/34 use feedforward for two layers, whereas ResNet-50/101/152 feedforward three layers that allow for an even deeper still trainable CNN [46]. Figure 3.8 illustrates the residual connection for three layers. This thesis utilizes both ResNet-50/101 as backbones when training object detection models.

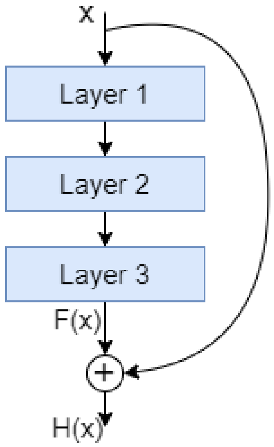


Figure 3.8: Deep residual learning building block for ResNet-50/101/152 [46]

Inception

The Inception network was introduced in 2015 by Christian Szegedy et al. and proposed sparsely connected architecture to reduces the network parameters and computational cost without the expense on the network's performance [47]. Inception module integrating several filter sizes that allows the layer to choose the most relevant filter for optimal learning. This provides an architecture with wider layers, as well as a deeper network without unreasonable computation, illustrated in Figure 3.9. The first Inception network (v1) contained nine Inception units and was 22 layers deep [47]- Figure 3.9 illustrates an Inception unit layer. As many other deep CNN and Inception network experience vanishing gradient in the backpropagation. Two auxiliary classifiers were appended to intermediate layers to provide additional regulation and reduce the vanishing gradient in the network. A weighted combination of auxiliary loss and the real loss construct the total loss during training.

In 2016 Christian Szegedy et al. proposed Inception v2 and v3 to improve network computation from its predecessor [48]. These networks focus on optimizing computation by factorizing convolution and conducting regularization. The motivation behind this updated Inception module versions was that CNN performs worse when convolution alters the input dimension drastically, resulting in information loss known as a representational bottleneck. The solution was to compress input layers dimension to reduce the computational cost and additionally increase the network's accuracy. Filter banks were constructed to be broader instead of deeper to avoid representational bottleneck [48]. Figure 3.9 illustrates factorization implementation in the Inception v2 version.

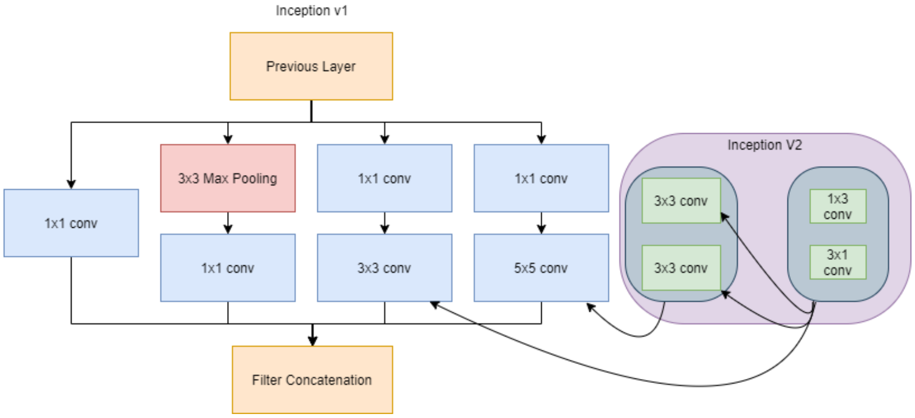


Figure 3.9: Illustration of the Inception module (v1 and v2 versions) [47]

3.5.4 Post-Processing

Non-Maximum Suppression

Non-Maximum Suppression (NMS) is a post-processing technique used in computer vision algorithms to designate a final BB out of multiple overlapping entities. NMS is similar to the mathematical optimization technique; hill-climbing search [49]. Object detection networks often generate multiple proposals of different size and aspect ratios for a single object, creating overlapping BB, where neighbouring proposals often share similar objectness scores. The NMS filter compares BB prediction to their neighbouring proposal to sort out and find the best BB representation, such that there is only one BB representation for each object.

All proposed BBs in a given image are composed in an initial proposal list. The NSM algorithm takes the proposal with the highest objectness score, removes it from the initial list and adds it to a final proposal list. This proposal is compared to the rest of the proposal, calculating the Intersection over Union (IoU) between them. If IoU exceeds a fixed threshold the proposal from the initial list is removed. A typical first stage NMS IoU threshold value of 0.7 has been utilized throughout this thesis. Then again, the proposal with the highest objectness score is removed from the initial list and appended to the final proposal list and compared to the rest of the proposals. This procedure replicates until the initial list is empty.

Repercussion from applying the NMS algorithm is that networks will have a problem detecting multiple similar objects nearby each other, such as a crowd full of people. The network will most likely draw a single BB around the crowd and classify it as a single person.

3.5.5 Anchors

Most state-of-the-art detectors rely on anchors to better locate target objects. Therefore, optimizing anchor parameters can have a significant impact on CNN performance [50].

CNN model configuration files define pre-default anchors for bounding box proposals. Adjusting the anchors will help indicate what size and shape to look for when detecting an object. The anchor aspect ratio derives from the height to width ratio. For example, if the height of the bounding box is two times longer than the width, it would result in an aspect ratio value of 2.0. Opposite, with a width two times longer than the height, the resulting aspect ratio value would be 0.5. The width to height ratio also needs to be considered, especially when applying data augmentation techniques such as rotation.

3.6 Metrics

This section explains the metrics used in this thesis to evaluate network performances.

Intersection over Union

Intersection over Union (IoU), based on Jaccard Index [51], is the most commonly known evaluation metric in object detection [52]. It compares the similarity and diversity between the predicted BB (P_b) and ground truth BB (G_b). Equation 3.2 shows that the area of the intersection divided by the area of union defines the IoU. Figure 3.10 illustrates the overlap between P_b and G_b.

$$\textbf{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}} = \frac{|P_b \cap G_b|}{|P_b \cup G_b|} \tag{3.2}$$

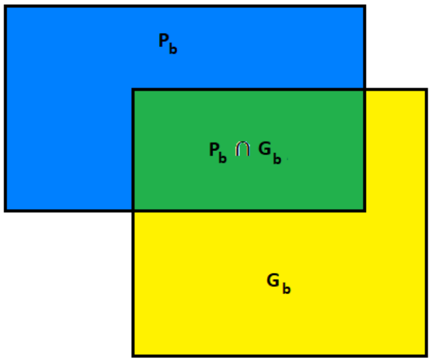


Figure 3.10: Illustrates of how IoU is found from predicted BB (P_b) and ground truth BB (G_b) overlap.

Confusion matrix

The confusion matrix is a popular method used to describe the performance of the localization and classification based on the predicted and the actual values of the data. Table 3.1 represents a confusion matrix with n=2 classes, used in CNN evaluation. A detection is classified as True Positive, correct positive prediction when IoU≥threshold. The threshold is normally set to 50%, but can be adjusted based on the object type, user scenario or other preferences. Detection with IoU<threshold classifies as False Positive, incorrect positive prediction. When the ground truth is not detected, the prediction classifies as False Negative, incorrect negative prediction. True Negative is not applicable in this context due to its representation of all possible detection that is correctly not detected [53]

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive	False Negative
Actual Negative	False Positive	True Negative

Table 3.1: Confusion matrix with n=2 classes.

Precision

Precision estimates the percentage of correct prediction, per class, based on all detection [54]. The average precision (AP) measures the prediction performance for an individual class and is an useful metric to output to measure if the model struggles to detect any of the data classes. The formula for precision is illustrated in Equation 3.3

$$\textbf{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{3.3}$$

Mean Average Precision

Mean average precision (mAP) is the average AP, as shown in Equation 3.4, and measures the model performance for all classes. The AP is usually evenly distributed when calculating the mAP but can also be weighted based on the number of cases in a given class.

$$\mathbf{mAP} = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{3.4}$$

Recall

Recall measure correct prediction based on all ground truth and Equation 3.5 shows the recall formula [55]. This is relevant for cases with an imbalanced data set, where the recall indicates how accurate the model is at correctly classifying relevant predictions.

$$\mathbf{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{3.5}$$

Mean Average Recall

Equation 3.6 shows the mean average recall (mAR) function, which measures the average AR for all classes. Like mAP, classes can be weighted based on the number of examples.

$$\mathbf{mAR} = \frac{1}{N}\sum_{i=1}^{N} AR_i \tag{3.6}$$

Loss

Loss is in most cases printed after every step, or epoch, during model training. There is negligible information about the network performance to gather from the loss value. However, the rate of change can reveal whether the model is learning or not, which again can prevent the CNN from overfitting [30].

3.7 Software

This thesis uses the programming language Python for CNN training, technical implementation, image pre-procession, data collection and data analysis [56]. Repertory structure is illustrated in Appendix F and the code is available at GitHub [1]. The open source Jupyter notebook environment, Google Colab, are used for image-preprocession and data analysis, providing easy access to popular libraries and a free (limited time) GPU. This thesis trains the complex deep CNN experiments on a Tesla V100-PCIE-32GB GPU [57]. This section introduces and explains some of the important python libraries implementation for this thesis.

3.7.1 Tensorflow

TensorFlow is a machine learning library developed by Google Brain Teams that utilize data flow graph [58]. The name originate from the operation that CNN execute on tensors, otherwise known as multidimensional data arrays. A wide variety of deep CNN algorithms, such as training and presumption, can be applied using TensorFlow. TensorFlow also provides an useful toolkit known as Tensorboard that can easily tracks and visualizes metric performance.

This thesis utilize Tensorflow Object Detection API 1x [2] to train, evaluate and deploy object detection models [35].

3.7.2 Numpy

Numpy is a fundamental Python library that provides multidimensional array objects, array matrices, numerical computing and an accumulation of mathematical functions [59]. This library has been used throughout this thesis for data analysis and in the image pre-processing stage for generating arrays, integers and executing mathematical operations.

[1] https://github.com/enliden1/Master_PCa_Detection

[2] https://github.com/tensorflow/models/tree/master/research/object_detection

3.7.3 Pandas

Pandas is a Python programming language library tool built on Numpy packages, practical for data structures and data analysis [60]. This library is used to both read and collect vital data about the patients that is provide in multiple csv files. Pandas is also adopted to construct annotation files for ground truth data, later discussed in Section 4.2.6.

3.7.4 SimpleITK

SimpleITK is an open source simplified programming interface of the Insight Segmentation and Registration Toolkit (ITK), supported by multiple programming languages [61]. This library provides a wide variety of image analysis filter and supports several types of image file formats. This thesis make use of this toolkit in the image pre-processing step for image normalization and image filtering.

Chapter 4

Material and Methods

4.1 Description

The data set used in this thesis consists of 200 individual patients from the PROSTATEx Challenge *SPIE-AAPM-NCI Prostate MR Classification Challenge* associated with the 2017 SPIE Medical Imaging Symposium held in Orlando, USA [62]. The PROSTATEx Challenge focuses on predicting the clinical significance of the lesions found in MRI images. Acquisition of the prostate MR was performed under the supervision of prof. Dr.Jelle Barentsz, at Radboud University Medical Centre (Radboudumc) in the Prostate MR Reference Center [63]. The data set was accumulated and systematized under the supervision of Dr. Huisman, at Radboudumc. Both Siemens 3T MRI scanners MAGNETOM Trio and Skyra gathered the MRI images for the PROSTATEx Challenge dataset [64, 65]. Three b-values of 50 s/mm^2, 400 s/mm^2, and 800 s/mm^2 were procure for the ADC map.

The lesion mask has been reviewed and performed by Renato Cuocolo et al. and is available on GitHub [1][66, 67]. The mask data assemble the first 204 MRI scans from the PROSTATEx challenge data set, which consists of a total of 345 individual prostate MRI scans [62]. There is missing information about lesion mask data from four MRI scans (ProstateX-0052, Prostate-0056, ProstateX-0080, ProstateX-0138), which result in a data set containing 200 individual Patients with lesions findings. MRIs consist of a collection of two-dimensional images that combined represent a three-dimensional image.

The mask data set contains 299 lesion findings distributed on 200 patients. Table 4.1 shows a detailed description of lesion examples, anatomic zone location, and the classification distribution of the data set. Chapter 2.1 explains the anatomic

[1] https://github.com/rcuocolo/PROSTATEx_masks

zone location and PCa classification in greater details. The classification between lesions is unevenly distributed, where insignificant represent 74.58% and clinically significant represent 25.42% of all the lesions. The same applies to the GGG score distribution, see Table 4.1. Lesion findings that did not undergo a prostate biopsy test, see Section 2.2.3, are graded as insignificant along with GGG 1, see Section 2.2.5 and Table 4.1.

	Insignificant	**Clinically Significant**					**Total**
Lesion	223	76					299
Lesion (%)	74.58%	25.42%					100%
TZ	67	9					76
PZ	134	36					170
AFS	22	31					53
Ground truth	914	367					1281
Ground truth (%)	71.35%	28.65%					100%
	No Biopsy	**GGG 1**	**GGG 2**	**GGG 3**	**GGG 4**	**GGG 5**	**Total**
Lesion	187	36	41	20	8	7	299
Lesion (%)	62.54%	12.04%	13.71%	6.69%	2.68%	2.34%	100%
TZ	59	8	5	3	1	0	76
PZ	122	14	20	8	3	3	170
AFS	8	14	15	8	4	4	53
Ground truth	748	166	190	100	42	35	1281
Ground truth (%)	58.39%	12.96%	14.83%	7.81%	3.28%	2.73%	100%

Table 4.1: Number distribution of lesion findings, anatomical zone location, classification with respect to the significant and GGG score, number of images represented in the prostate data set.

Studying the PCa anatomic zone location shows that 56.86% of the lesion findings is allocated in the PZ, 25.42% in the TZ and 17.72% in the AFS zone. Usually, PCa has a higher chance to be located in PZ [7]. Lesions classified as clinically significant have an unequal probability between the three zones for this thesis data set. AFS has the highest possibility (58.49%), PZ has the second highest (21.18%) and TZ has the lowest probability (13.43%) to be classified as clinically significant, see Table 4.1.

One alluring ratio value from the Table 4.1, regarding the GGG classification, is the GGG 2 class which has a higher number representation than the GGG

1 classification. Otherwise the number of examples proportion decrease as the GGG score increase.

1109 unique two-dimensional slices contain at least one lesion finding. Ground truth shows the lesion classification distribution for all the objects obtained in the 1109 ADC images, see Table 4.1. Figure 4.1 shows that some patients have more than one lesion finding obtained, which means some image slices have multiple objects represented in them.

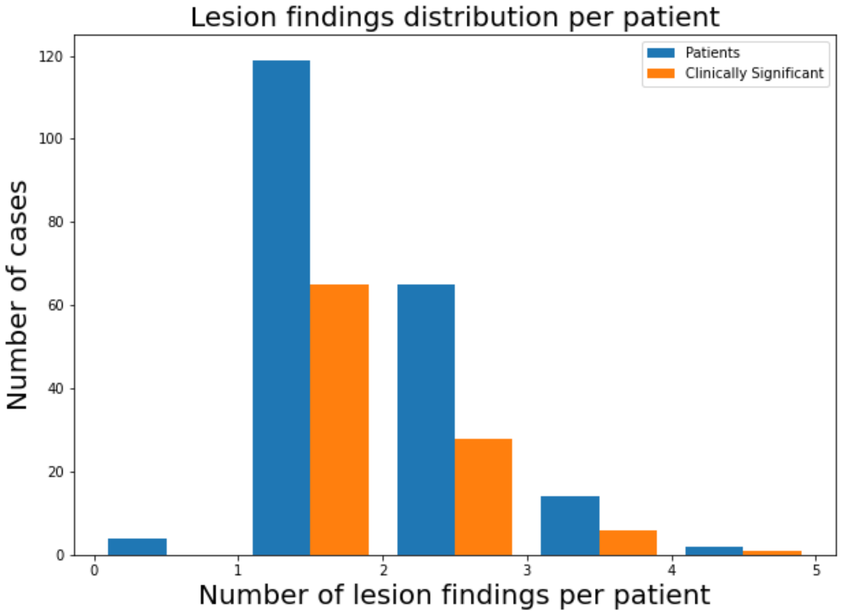


Figure 4.1: Histogram plot shows distribution of lesion findings per patient for the prostate data set, additionally to the distribution of how many of these patients have at least one PCa classified as clinically significant. Number of findings range from 0 to 4.

Figure 4.1 shows the number of cases and the percentage of each number of lesion findings per patient. One to two findings per patient are the most common occurrence, while three to four lesion findings are infrequent and are only present in 8% of the data set. A larger quantity of lesion findings for an individual patient can give the impression of a higher probability that at least one of them is classified as clinically significant, but that is not the case. For one finding, there is 54.6% chance for it to be classified as clinically significant. For two, three and four findings there is 43.1%, 42.9% and 50% chance for at least one of the findings to be classified as clinically significant.

4.2 Image Pre-Processing

This section presents image pre-processing implementation used to prepare images for training and evaluation. Different techniques such as reshaping data, saving organized data and data filtering are inspired by Steinar Valle Larsen work on the PROMISE12 challenge [68].

4.2.1 Organizing Data Sets

The complete data set is divided into training, validation and test set with a distribution of 70%, 10% and 20%, respectively. The training set fits the CNN model for training, whereas the validation data set is held back from training to give an unbiased evaluation of the model's performance while training, to improve hyperparameters tuning. Finally, the final trained CNN utilizes the test data set to provide an unbiased evaluation [69].

The data set is split based on the patient and not the image, which means image slices from one patient only belong to one of the three data sets. Evaluation has to be proceeded with no prior recollection to the patient to avoid biased evaluation. In addition, the data split procedure obtains stratification to distribute the number of classes equally and to mimic the original data distribution to reproduce a real-world scenario case.

4.2.2 Data Filtering

DICOM 16-bit images has pixel values ranging from [-32768,32768]. Normalization is applied with pixel value ranging from [0,255] to obtain 8-bit JPEG images (required by Tensorflow Object Detection API), in which equation 4.1 specifies how to apply linear normalization [70, 35]. The values [Min,Max] represent the input image (I) and the [newMax,newMin] represent the desired pixel value limitations for the output image (I_N). 0 corresponds to black and 255 to white pixel value for a grayscale image

$$I_N = (I - Min)\frac{newMax - newMin}{Max - Min} + newMin \tag{4.1}$$

Outlier removal filter is a data filtering technique used to remove possible image noise. Applying percentile upper-limit and lower-limit, 99^{th} and 1^{th} respectively, proceeds to replace the darkest and the brightest 1% pixels value to its surrounding neighbour values [71].

4.2.3 Data Resizing

The original images are resized to have height and width equal to 512x336. This shape size is found by simply multiplying the most common original shape size (128x84) with three, see Table 4.2. The outcome is to use fixed input image shape and maintain the original aspect ratio for 94.5% of the image representations in the data set.

Height x Width	128 x 120	106 x 128	128 x 84
Number of cases	6	5	189

Table 4.2: Table shows number of cases with different image size from the ADC data set, where each case represent an individual patient

4.2.4 Image Cropping

Bounding box (BB) size is an essential factor for a CNN object detector. Relative to the image size, the lesions are often small. The CNN can have problems detecting them if the area size is limited, especially if the area size is $< 30^2$ or does not correspond to the predefined anchor boxes. COCO metrics provides metric techniques that evaluate CNN performance relative to the object size (small, medium and large), further explained in Section 4.4.1.

Category	Boundaries	Original	Original (%)	Cropped	Cropped (%)
Small BB	$area < 32^2$	835	65.44%	5	0.39%
Medium BB	$32^2 < area < 96^2$	441	34.56%	831	65.13%
Large BB	$area > 96^2$	0	0.00%	440	34.48%

Table 4.3: Number of objects, in the data set, that classifies as either small, medium or large BB area size category, accordingly to the coco metrics explained in section 3.6. Both original images(512x336) and cropped image (600x600) are represented.

	Insignificant	Clinically Significant	Total
Mean BB area (Original)	913	1092	964
Mean BB area (Cropped)	8335	10012	8816

Table 4.4: Table shows the mean BB area size of the original and cropped data sets.

Using the original image with a shape size equal to 512x336 shows that the majority of BB classifies as small BB (65.43%), with a mean BB area value of 964 ($<32^2/1024$). Table 4.3 illustrates distribution of BB size category and Table 4.4 shows the mean BB area size for original and cropped data sets. Note that increasing the image size will correspondingly increase computational cost, which again will increase training time for fixed CNN architecture. Cropping implementation will remove unnecessary image information and increase the BB size without the great cost of the computation. Without losing any lesion objects in the prostate data set, the image is cropped to have an outer edge range of [156:356] for height and [68:268] for width, that produce an image dimension of 200x200. The image shape size is again multiplied by three to increase the overall BB area size, producing a cropped image with a size equal to 600x600, see Figure 4.4. Implementing these image pre-processing techniques produce a total mean BB area value of 8816, close to the COCO metrics defined large BB area value ($<96^2/9216$) [72].

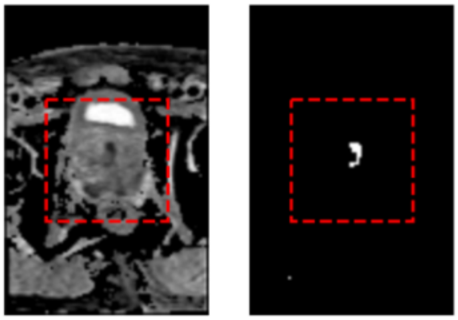


Figure 4.2: Original image (512x336)

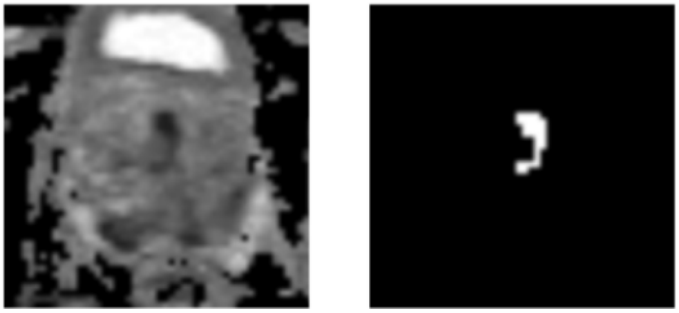


Figure 4.3: Cropped image (600x600)

Figure 4.4: Illustrates of cropping [156:356,68:268] implementation on ADC image with shape size equal to 512x336, before resizing it to 600x600.

4.2.5 Save Data Information

Information from each patient is stored in compressed files (.npz) provided by Numpy, which makes detailed analysis more accessible and makes reproducibility easier [73]. Storing patients information provides easy access to a complete patient overview, where Table 4.5 illustrates the array layout. ADC images are read as a Digital Imaging and Communications in Medicine (DICOM) image format and are stored as a three-dimensional array in the zipped archive files [74]. The two first indexes correspond to the [height,width] and the third index corresponds to the number of two-dimensional images slices. Lesion masks data are provided in Neuroimaging Informatics Technology Initiative (NIfTI) file format. Given that each lesion findings represent a single NIfTI file, the mask array is saved as a four-dimensional array. The first index corresponds to the number of lesion findings, and the other three indexes represent the same as for the ADC MRI array (DICOM). PCa GGG and significant classification are stored in separate arrays.

Array number	Description	Shape
arr[0]	DICOM	(Height,Width,Slices)
arr[1]	NIFIT	(Findings,Height,Width,Slices)
arr[2]	ID	Prostate-XXXX
arr[3]	GGG	1-5
arr[4]	Clinically Significant	True/False

Table 4.5: Npz file construction of the patient information.

4.2.6 Data Labeling

Train, validation and test data sets are prepared by saving two-dimensional slices as individual JPEG images and creating csv file for BB ground truth. Detected lesions in mask data represent the white pixels (255), and the background represents the black pixels (0), as seen in Figure 4.5. Producing PCa ground truth label consists of locating BB coordination in the relevant mask images and then applying it to the MRI images. The mask BB is obtained by simply finding maximum and minimum white pixel values, where maximum and minimum coordinate values are added and subtracted by one to ensure that the whole object is inside the produced BB. However, the approach of finding the minimum and maximum white pixel value works poorly for mask images with errors, such as a random white pixels that appear unrelated to the lesion object. Most of these errors are revealed by seeking out abnormal BB sizes or aspect ratios, which is further explain in Section 5.3. BB errors are corrected by manually adjusting the BB coordinates to ignore the random white pixel occurrences. Each mask image represents a single object, meaning some MRIs slices have multiple mask images.

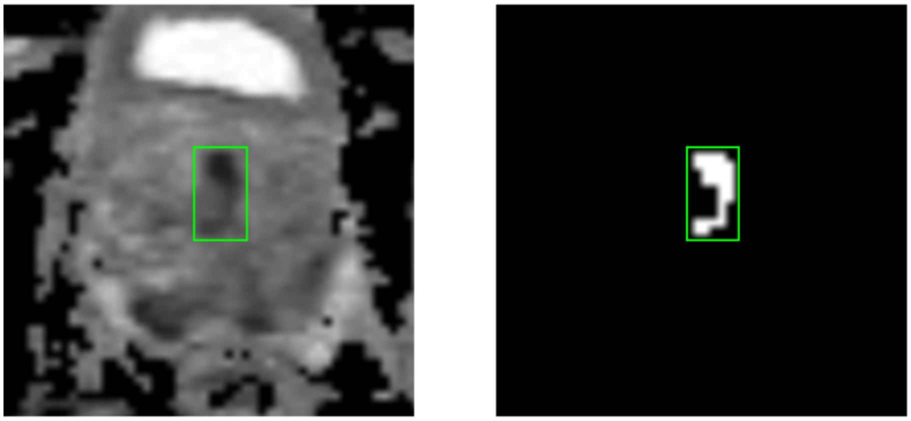


Figure 4.5: Mask image slice align proportional with the respective ADC MRI image slice, with a drawn BB around the corresponding lesion object.

The annotation files contain lesion information about the image filename, image shape size and BB outer edge coordinates, illustrated in Table 4.5. Each annotation line represents a single object, such that the number of annotation lines equals the number of lesions in a given image. Annotation files contain only information about MRI slice with lesion objects, where MRIs without lesions will not contribute during training nor evaluation.

Filename	Width	Height	Class	x_{min}	y_{min}	x_{max}	y_{max}
ProstateX-0002-5.jpg	336	512	Clinically Significant	125	261	166	290

Table 4.6: Structure of how the ground truth is presented in the annotation csv file.

4.2.7 Mask Data Issues

ADC mask data has a representation issue due to format conversion, in such that the mask images are rotated 90 degrees relative to the MRI images. Thus mask images is transposed to get a proportional representation as to the corresponding MRI image. Additionally, patients mask image slice order is inverted (except for PROSTATEX-0199 - PROSTATEX-0203) to get a proper mask-data alignment. Both these interpretations to correctly represent the mask slice relative to the respective ADC slice are only a proposed solution. When gathering the mask data, these issues were unknown to the author or me. To the best of my knowledge, these are the only mask data complications, but it should be considered that there could be other mask data issues.

4.3 Data Augmentation

One of the most crucial aspects to focus on when working with deep CNN is the size of the data set. From Ian Goodfellows book Deep Learning, a general guideline is to use around 5000 labeled examples per class too achieve acceptable performance, and 10 million labeled examples per class to exceed human performance in supervised learning [30]. If the model performance is under-performing it is often because of an insufficient data set. Increasing the data set will improve the CNN ability to generalize and also prevent from overfitting at a relatively early stage [30]. Image data augmentation expands the training data set artificially, by modifying the input images. Methods must be selected according to the data set. It is worth mentioning that some image augmentation methods could deteriorate the models performance, for example by applying vertical flip to images of cars is maybe not the best augmentation implementation, since the probability of the trained CNN to receive an input image of a upside down car is low [33]. This of course depends on the given scenario.

This thesis makes us of the build in data augmentation techniques in Tensorflow Object Detection API [35], where the augmentation techniques will additionally impact bounding box annotation. Implementations used in this thesis is illustrated in Figure 4.6 and listed below:

- Horizontal flip: Reversing each row of the matrix.
- Vertical flip: Reversing each column of the matrix.
- Rotation: Rotates the images in different degrees. Rotation augmentation applied with a, default, fixed number of 90 degrees in this thesis.
- Crop image: Removes parts of the image, by adjusting image outside edge coordinates.
- Brightness: Adjust the overall pixel value in the image.
- Contrast: Adjust the luminance difference between dark and bright pixels values [75].

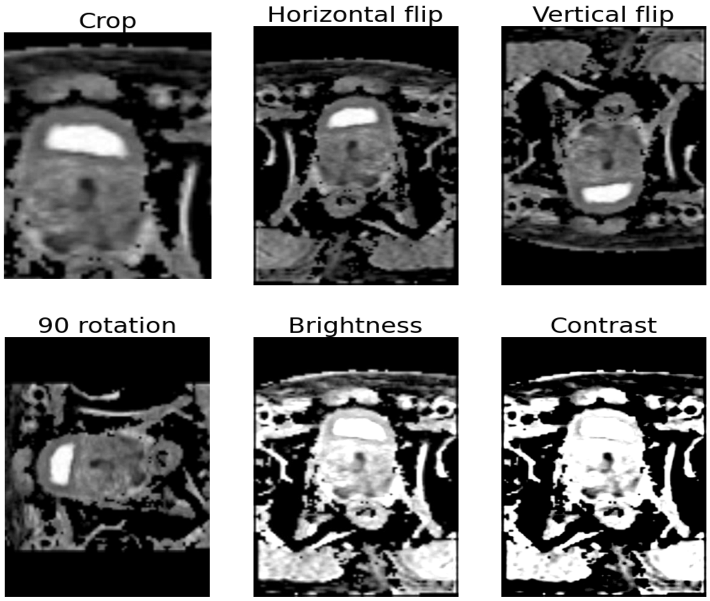


Figure 4.6: Illustration of data augmentation techniques used in this thesis.

4.4 Metric Evaluation

This section introduce the evaluation metric sets used to evaluate networks performance for all the test experiments in Chapter 6. Section 3.6 explain metric background theory.

4.4.1 Common Objects in Context Metrics

Common Objects in Context (COCO) detection evaluation metrics is used to evaluate an object detector performance on Microsoft COCO (MS COCO) data set [72, 76]. The MS COCO data set is a large-scale object detection, annotation and caption data set, consisting of 1,5 million object instances and 80 object categories [77]. Table 4.7 explains the 12 different metrics provided by COCO metrics, used to measure model performance in this thesis.

mAP	
mAP[0.50:0.05:0.95]	IoU=0.50:0.05:0.95
mAP[0.50]	IoU=0.50
mAP[0.75]	IoU=0.75
mAP Across Scales	
mAP_{small}	mAP for small objects ($32^2 > area$)
mAP_{medium}	mAP for medium objects ($32^2 < area < 96^2$)
mAP_{large}	mAP for large objects ($area > 96^2$)
mAR[0.50:0.05:0.95]	
$mAR(max = 1)$	mAR given 1 detection per image
$mAR(max = 10)$	mAR given 10 detection per image
$mAR(max = 100)$	mAR given 100 detection per image
mAR[0.50:0.05:0.95] Across Scales	
$mAP_{small}(max = 100)$	mAR for small objects ($32^2 > area$)
$mAP_{medium}(max = 100)$	mAR for medium objects ($32^2 < area < 96^2$)
$mAP_{large}(max = 100)$	mAR for large objects ($area > 96^2$)

Table 4.7: COCO 12 metrics used to measure object detection model performances [72]

The COCO detection evaluation metrics is one of the methods used to evaluate the models performance in this thesis. Both COCO AP and AR represent the average over all categories (mAP and mAR), not the AP or the AR for a single class. To avoid any deception, COCO AP and AR will be consider as mAP and mAR throughout this thesis.

4.4.2 Pascal Visual Object Classes Metrics

Pascal Visual Object Classes (PASCAL VOC) challenge is another popular standard large-scale data set of images and annotations that also provide an evaluation metric method [78]. The open-source data set consist of 20 class categories. As demonstrate in figure 4.7 the COCO metrics compute $mAP^{IoU=.50}$ utilizing PASCAL VOC metrics. However, for measuring the AP of every individual class an additional evaluation metric method need to applied; the standalone PASCAL VOC metric system. Being able to look at the AP performance for each individually class is a crucial aspect for this thesis.

Chapter 5

Configuration

5.1 Backbone Networks

Table 5.1 compares the number of deep layers and parameters for this thesis implemented backbones networks [48, 43, 46]. Number of parameters range from 3.5M (MobileNet v2) to 44.5M (ResNet-100). CNN latency time is heavily affected by the number of parameters, where a high number of parameters also makes it difficult for performance optimization during training.

Network	**Layers**	**Parameters**
Inception v2	47	11.2M
MobileNet v2	53	3.5M
ResNet-50	50	25.6M
ResNet-100	100	44.5M

Table 5.1: Number of layers and parameters for the respective backbone networks.

5.2 Hyperparameters

Image Size

The Tensorflow Object Detection API resizes images by either using a fixed shape or by allowing padding to keep the original aspect ratio. Padding will add unnecessary information to images, which is undesirable. This project utilizes a fixed image input size of 336x512 for the original data set and 600x600 for the cropped data set.

Learning Rate

The two-stage detectors Faster R-CNN and R-FCN learn much faster than the one-stage detector SSD. Faster R-CNN and R-FCN start with a learning rate of 0.0002, while SSD begins with a learning rate set of 0.002, which gradually decreases while training. These are the default learning rate for each of the architecture. Each experiment learns at different rates, and therefore the learning rate adjustment is set at different time steps, which is illustrated in the Appendix for the respective test results.

Batch Size

Faster R-CNN and R-FCN utilize a batch size equal to 1, while SSD uses a batch size equal to 24. These values are selected from the default values set in the respective model configuration files.

Optimization

Faster R-CNN, R-FCN and SSD utilize momentum optimization (for gradient descent), with a default parameter value of 0.9 [79].

5.3 Anchors

Anchor Aspect Ratio

Taking a deeper look at the distribution for BB aspect ratio of this thesis data set, see Figure 5.1, shows that most height to width ratio is around 1:1. The minimum value is 0.30, the maximum value is 3.68, and the mean value is 0.96. Since data augmentation, such as rotation, is applied to the data set at a certain point, the width to height ratio also needs to be considered. Dividing the minimum and maximum height to width ratio value to one will output minimum and maximum value for the width to height ratio, which produces a new minimum value of 0.27. The highest anchor ratio value is still 3.68 because 3.68>3.33. The model will have problems detecting the objects, with a ratio close to 1:4 and 4:1, if the anchor aspect ratio is not defined based on the minimum and maximum value. Setting the aspect ratio too high or low relative to the data set could negatively impact the CNN performance.

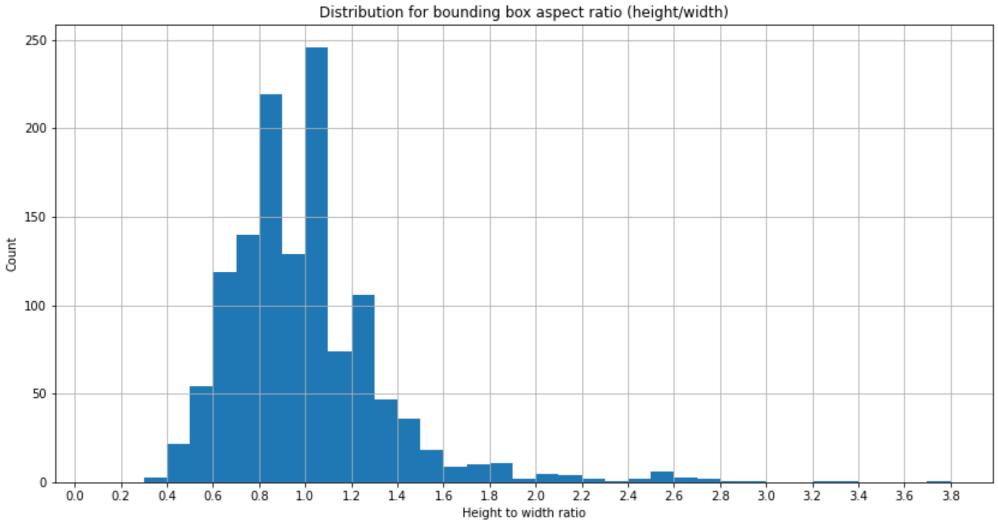


Figure 5.1: Height to width aspect ratio distribution for every ground truth BB annotation in the data set.

Anchor Size

Figure 5.2 shows the overall BB size distribution, relative to the image shape size, both before and after implementing the cropping image pre-processing technique (Section 4.2.4). The original image presentation ranges from [0.013, 0.121] and the cropped image depiction ranges from [0.041,0.350]. Choosing too small anchor sizes could damage the CNN performance because of a possible increase in false predictions. This thesis utilizes a minimum anchor size of 0.075 and a maximum size of 0.9 for the SSD models, which obtain 95.9% of all the ground truth BB sizes. Faster R-CNN utilizes anchors scales of size [0.25, 0.5, 1.0, 2.0]. From trial and error, these parameter values work best for the SSD and Faster R-CNN models. Decreasing these parameter values to exactly fit the minimum and maximum values damage the CNN performances because of the increase in inaccurate predictions.

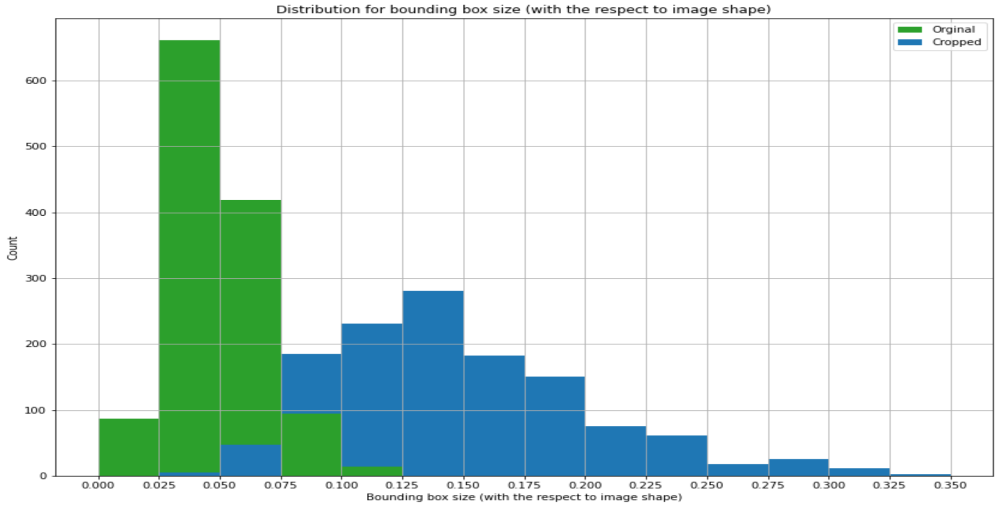


Figure 5.2: BB size distribution, with the respect to image shape size, for both the data set with original representation and the data set utilizing the cropped image preprocessing technique.

5.4 Data Augmentation

All applied data augmentation methods during training utilizes the default parameters value, set by Tensorflow Object Detection API [35].

Horizontal flip, Vertically flip and rotation (90 degrees) has a probability of 50% for being implemented during training. These techniques have a fixed result representation and are either fully applied or not applied to a given image. Crop, contrast and brightness adjustments values randomly range from a given minimum to a given maximum value. These augmentation methods are applied for all given image, with ranging inference impact. Cropping has an aspect ratio ranging from [0.75, 1.33] and an area ranging from [0.1, 1.0], where 1.0 is the original image area and 0.1 is 10% of the original image area. The cropping augmentation always makes sure that there is at least one object present in the cropped image. Contrast adjustment has a delta ranging from [0.8,1.25], and brightness adjustment has a delta value ranging from [0,0.2].

Chapter 6

Experimental Results

In this chapter, the performed experiments and the corresponding result are presented. Figure 6.1 illustrates an overview of the conducted experiments.

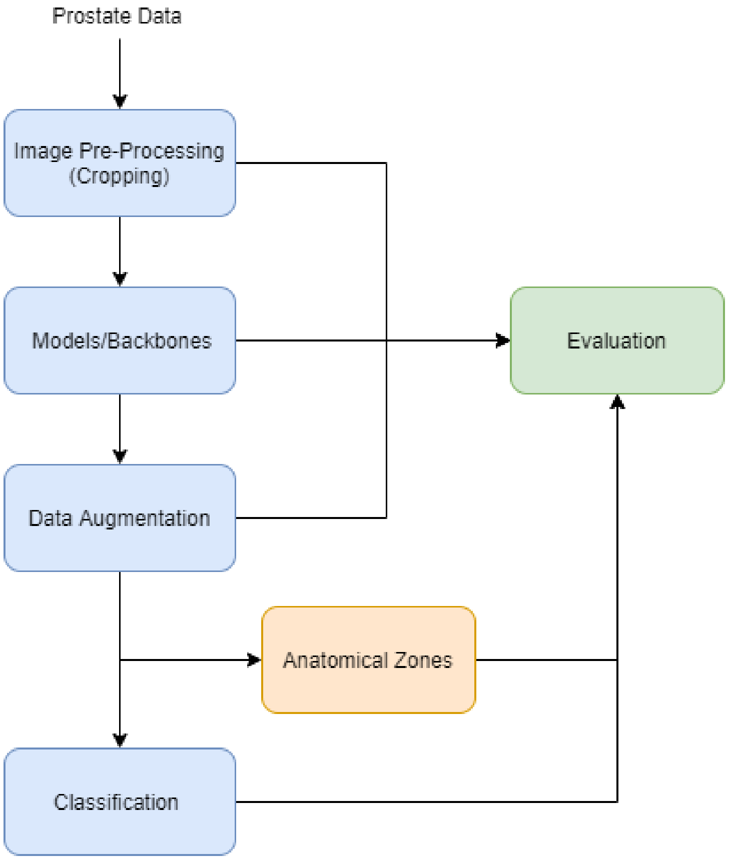


Figure 6.1: Overview of the conducted experiments

6.1 Cropping

Early on, there was an indication that CNN models struggled to detect PCa because the average BB representation was negligible compared to the rest of the image. Therefore the first experiment establishes a better image representation before further testing different models, backbones and training methods.

This section utilizes the Faster R-CNN with the MobileNets v2 backbone to compare test result from training on original images, with shape size 512x336, and cropped images with shape size equal to 600x600. Section 4.2.4 explains the cropping implementation in greater detail. Faster R-CNN Mobilenet v2 is used in this experiment because it was utilized in the early-stage tuning of the hyperparameters. It is worth mention that both the original and cropped data sets consist of the same patient examples for the split training, validation and test data sets.

Table 6.1 shows the best performance result from test data evaluation, and validation computed while training is illustrated in Appendix A.

Test	$\boldsymbol{AP_{Significant}}$**[0.5]**	$\boldsymbol{AP_{Insignificant}}$**[0.5]**	$\boldsymbol{mAP}$**[0.5]**	$\boldsymbol{mAP}$**[0.75]**	$\boldsymbol{mAR}$**[0.5 : 0.95]**
Original	0.182	0.056	0.120	0.004	0.158
Cropped	0.299	0.169	0.237	0.034	0.349
	$\boldsymbol{mAP_S}$**[0.5 : 0.95]**	$\boldsymbol{mAP_M}$**[0.5 : 0.95]**	$\boldsymbol{mAP_L}$**[0.5 : 0.95]**	$\boldsymbol{mAP}$**[0.5 : 0.95]**	
Original	0.013	0.013	NA	0.032	
Cropped	0	0.035	0.197	0.088	

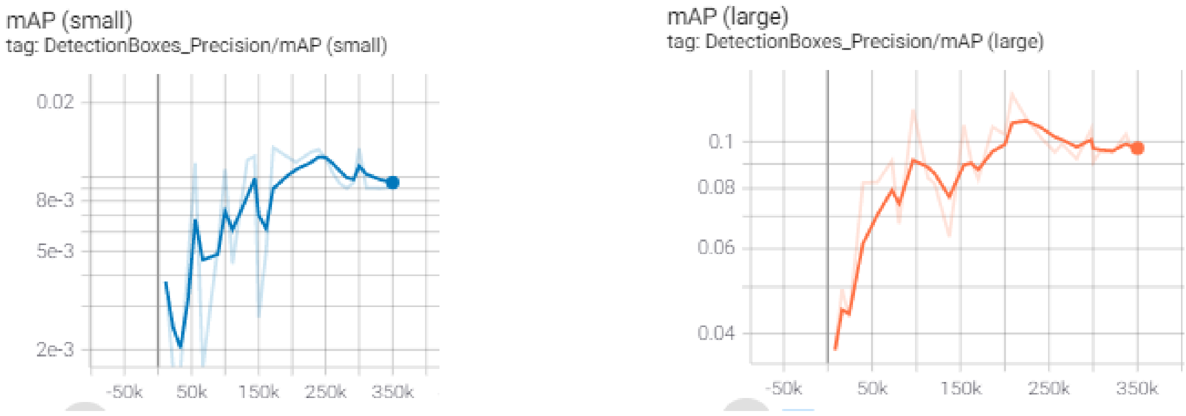
Table 6.1: Final performances evaluation of the original and cropped image data set are given in this table.

First, the original image data set was trained, with an image shape size equal to 512x336, then the cropped image data set was trained, with an image size equal to 600x600, and improvement was observed across all metrics parameters. The most interesting metric parameters comparison between these two data sets is the mAP regarding the BB area size. Table 4.3 reveals the BB area size distribution for both data sets. By cropping images, additionally to increasing the image size, close to all lesion objects shifts to a higher BB size classification (Table 4.3). The cropped mAP_M outperforms the original mAP_S by almost three times the performance value. However, the biggest improvement was seen between BB area size classification mAP_M (original) and mAP_L (cropped), where the performance value increased more than 15 times, see Table 6.1. These improvements are not entirely affected by the average BB expansion since removing unimportant information from images also enhances CNN performances. Original image data set have no representation of large area size objects, thus the mAP_L score is not available (NA). The test data set for the cropped images contain only two small-sized objects. However, the mAP_S outputs value 0 because of no correct predictions (P_b) of the two-small sized objects.

Figure 6.2 and 6.3 shows how the CNN performance evolves while training. Original validation data set has no example of large-sized BB, and cropped validation set has no instance of small-sized BB and is therefore not illustrated in the Figure 6.2.

Color	Blue	Orange
Test	Original	Cropped

Table 6.2: Color representation.



(a) MAP_S.

(b) MAP_L.

Figure 6.2: MAP[0.5:0.95] for small and large sized objects.

Both data sets have images containing medium-sized BB, in which cropped data set has a higher number of instances, see Table 4.3. Figure 6.3 shows the mAP_M performance for both original and cropped data sets while training. The cropped data set also exceeds mAP performance for medium-sized objects. However, the median BB area size for the original validation data set has an area size of 1721 and 4740 for the cropped validation data set, which substantially impacts the performance results. Graphs from Figure 6.2 and 6.3 interpret how objects area size affect the CNN models performance. Cropped mAP_M and mAP_L outperform original mAP_S and mAP_M, respectively.

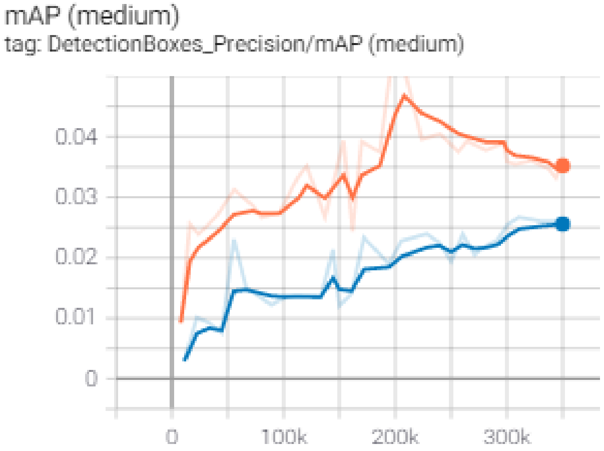
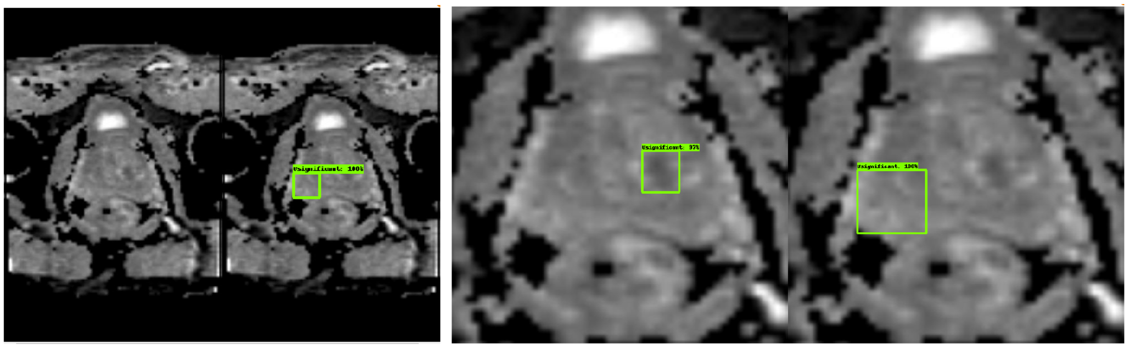


Figure 6.3: MAP[0.5:0.95] of medium (96^2<area<32^2) sized BB, evaluated on validation data set while training. Table 6.2 states color representation

Figure 6.4 and 6.5 illustrates prediction carried out by the trained Faster R-CNN models. Every image showcase the predicted box (P_B) to the left and the ground truth box (G_B) to the right.

The model fails to predict any lesion objects in the original image in Figure 6.4a. However, the cropped image in Figure 6.4b correctly classifies the lesion as insignificant, with an objectness score of 100%, but fails to predict (P_B) the ground truth (G_B). However, the P_B in Figure 6.4b draw a BB around a dark area in the ADC image, which is how most of the lesions is fabricate in ADC images.

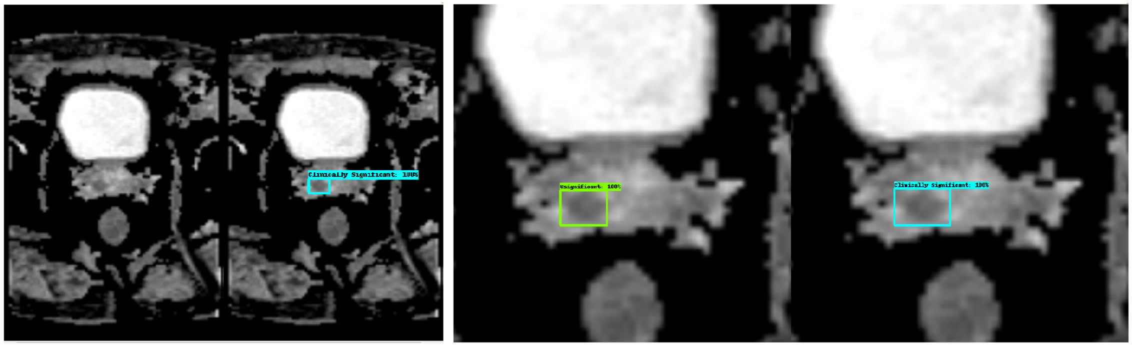


(a) Original images.

(b) Cropped images.

Figure 6.4: Figure displays P_B (left) and G_B (right) for both an original and a cropped image. Both example are of the same patient slice.

For the original image in Figure 6.5a the CNN does not carry out any predictions. However, the cropped image in Figure 6.5b achieve a correct prediction with an objectness score of 97% relative to the ground truth locations but fails to classify the lesion as clinically significant.



(a) Original images

(b) Cropped images.

Figure 6.5: Figure displays P_B (left) and G_B (right) for both an original and a cropped image. Both example are of the same patient slice.

6.2 Models and Backbones

This section explores different object detection architectures and backbone modules. The cropped data set was used because it gave a better performance result than the original image data set, as seen in Section 6.1. Three CNN models and four backbones were explored, where each of the CNN models has a limited number of backbones available in the Tensorflow Object Detection API [35].

Faster R-CNN and R-FCN models were trained for 300 000 steps, with a starting learning rate of 0.0002, that gradually decreased down to 0.00002 (Apendix B), except for the Faster R-CNN Inception v2 model (gather from Section 6.1) that trained for 350 000 step with a learning rate of 0.000002 for the last 50 000 steps. The last 50 000 steps gave no performance improvements and were therefore omitted from the model/backbone experiments. Table 6.3 shows that Faster R-CNN outperforms SSD and R-FCN, with every tested backbone. A limitation to these comparisons is that the ResNet-100 is the only backbone available for all the architectures.

Backbone	$\boldsymbol{AP_{Significant}}$**[0.5]**	$\boldsymbol{AP_{Insignificant}}$**[0.5]**	$\boldsymbol{mAP}$**[0.5]**	$\boldsymbol{mAP}$**[0.75]**	$\boldsymbol{mAR}$**[0.5 : 0.95]**
			Faster R-CNN		
Inception v2	0.299	0.169	0.237	0.034	0.349
ResNet-50	**0.348**	**0.127**	**0.240**	**0.044**	**0.341**
ResNet-100	0.324	0.114	0.221	0.025	0.282
			R-FCN		
ResNet-100	0.199	0.138	0.172	0.034	0.361
			SSD		
Inception v2	0.164	0.090	0.130	0.035	0.163
MobileNet v2	0.177	0.057	0.120	0.045	0.131
ResNet100 v2	0.238	0.137	0.189	0.034	0.298

Table 6.3: Performances parameters from the final evaluation, on different CNN architectures and backbones.

Input image size of 300x300 was used when training SSD Inception v2 and MobileNet v2, since it is the default input size in these models configure files. There were conducted experiment with increasing input image size of the SSD model where performance result did not improve and training time increased dramatically. However, the SSD ResNet-100 model has a default input image size of 640x640, which also was utilized in this experiment. The performance for the SSD ResNet-100 improved compared to the other SSD tests, but the difference between the SSD Inception v2 and ResNet-100 experiments indicates that it is because of the backbone rather than the configured input image size, since the Faster R-CNN model also improved substantially when training with the ResNet-100 compared with the Inception v2 backbone.

6.3 Data Augmentation

The previous experiments have utilized the default data augmentation implementation horizontal flip. This section explores more augmentation technique to help improve CNN performance, using the most promising architecture and backbone from Section 6.2; Faster R-CNN ResNet-50. Table 6.5 shows the performance result and Table 6.4 explains the augmentation technique implementation for its respective test number. Appendix C shows evaluation performance while training.

The most promising augmentation methods for the prostate data set was the combination of horizontal flip, vertical flip, rotation and crop, which produced $AP_{Significant}[0.5]$ of 0.424 and $AP_{Ingnificant}[0.5]$ of 0.156 (Test 5). Test 5 used augmentation techniques from both Test 3 and 4, which improved the network performance. Brightness and contrast adjustment augmentations produced a negative impact on CNN performance, as seen for Test 2 in Table 6.4. These pixel augmentations methods were also applied in Test 6 to see if they could positively impact CNN performance when combined with horizontal flip, vertical flip, rotation, and crop implementations. Comparing Test 5 and 6 reveals that pixel augmentation also decreases performance result combined with other promising augmentation techniques.

Test	**1**	**2**	**3**	**4**	**5**	**6**
Augmentation Implementations	Horizontal flip	Horizontal flip Brightness Contrast	Horizontal flip Vertical flip Rotation	Horizontal flip Crop	Horizontal flip Vertical flip Rotation Crop	Horizontal flip Vertical flip Rotation Crop Brightness Contrast

Table 6.4: Test number explanation of the different augmentation experiments.

	$AP_{Significant}$[0.5]	$AP_{Insignificant}$[0.5]	mAP[0.5]	mAP[0.75]	mAR[0.5 : 0.95]
Test 1	0.348	0.127	0.240	0.044	0.341
Test 2	0.274	0.091	0.184	0.040	0.328
Test 3	0.414	0.138	0.276	0.056	0.424
Test 4	0.396	0.119	0.260	0.084	0.391
Test 5	**0.424**	**0.156**	**0.291**	**0.132**	**0.457**
Test 6	0.371	0.155	0.263	0.031	0.410

Table 6.5: Performances parameters from final evaluation. Table 6.4 explain the augmentation implementation for the experiments.

6.4 Anatomical Zone

This section evaluates the best performing Faster R-CNN ResNet-50 from Table 6.5 (Test 4), relative to the PCa anatomical zone. The cropped test data set (used in Section 6.1, 6.2 and 6.3) is split into three data set for the respective anatomical zone, to evaluate (the already trained) CNN performance of lesions located in different prostate zones.

Anatomical Zone	$AP_{Significant}$[0.5]	$AP_{Insignificant}$[0.5]	mAP[0.5]	mAP[0.75]	mAR[0.5 : 0.95]
PZ	0.096	0.089	0.093	0.007	0.372
TZ	**0.764**	**0.302**	**0.534**	**0.267**	**0.548**
AFS	0.529	0.077	0.303	0.094	0.437

Table 6.6: Final performance evaluation relative to the anatomical zones.

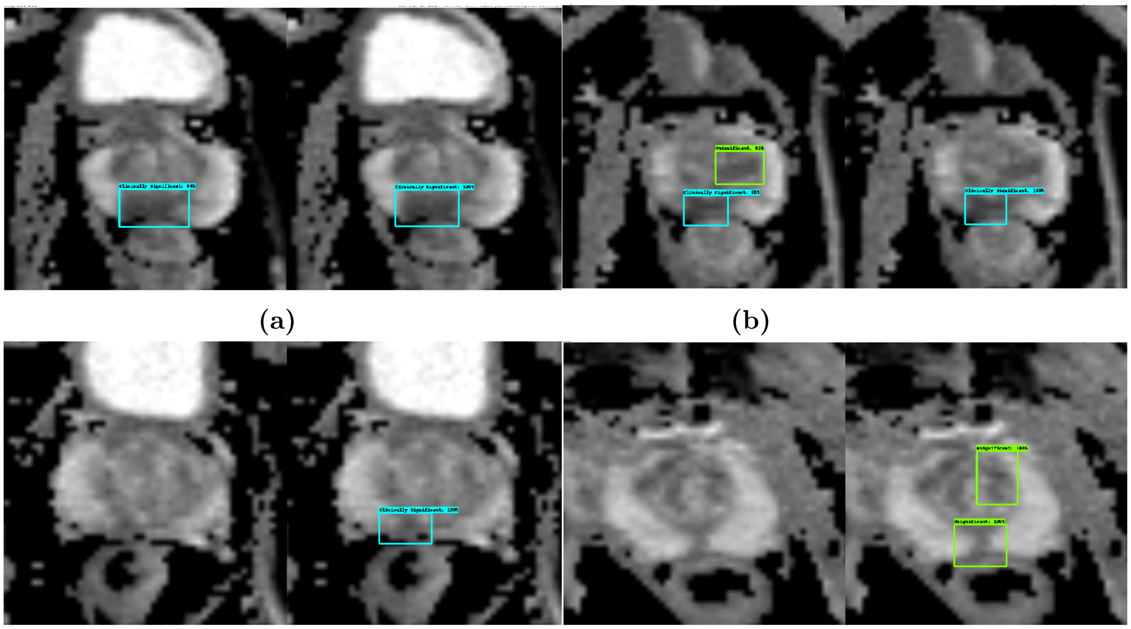
Table 6.6 shows that CNN performs worst on PCa located in PZ, even though ADC is the dominant MRI technique to detect PCa in PZ. Because the performance result is so different from each other, it is essential to look at BB area size of the three (split) test data sets. Table 6.7 reveals that the median area size of the clinically significant lesion is exceptionally low and explains why the CNN model struggles to detect these lesions. TZ provides the most promising performance values, obtaining $AP_{Significant}[0.5]$ of 0.764 and $AP_{Inignificant}[0.5]$ of 0.302. The Median BB area size for clinically significant BB is 15484, which is considerably higher than for the other prostate zones. However, $AP_{Inignificant}[0.5]$ for TZ also outperform $AP_{Inignificant}[0.5]$ for PZ, although both have the same median BB area size. This experiment is an unfair comparison considering the number of labelled data, classes and BB area size is unevenly distributed between the three zones is, as seen in Table 6.7.

Anatomical Zone	Insignificant	Clinically Significant
PZ	90	29
Median BB area	6600	5632
TZ	59	15
Median BB area	6600	15484
AFS	43	25
Median BB area	5632	11563

Table 6.7: Number of ground truth labels for the three prostate anatomical zone and median BB area size, relative to the PCa significance.

Figure 6.6 (PZ), 6.7 (TZ) and 6.8 (AFS) illustrates predictions carried out by the Faster R-CNN ResNet-50 model, relative to the prostate anatomical zones. Each sub-figures shows the prediction BB (P_B) to the left and ground truth BB (G_B) to the right.

PZ


Clinically Significant: 64%
Clinically Significant: 100%
Insignificant: 62%
Clinically Significant: 88%
Clinically Significant: 100%
Clinically Significant: 100%
Insignificant: 100%
Insignificant: 100%

(a)

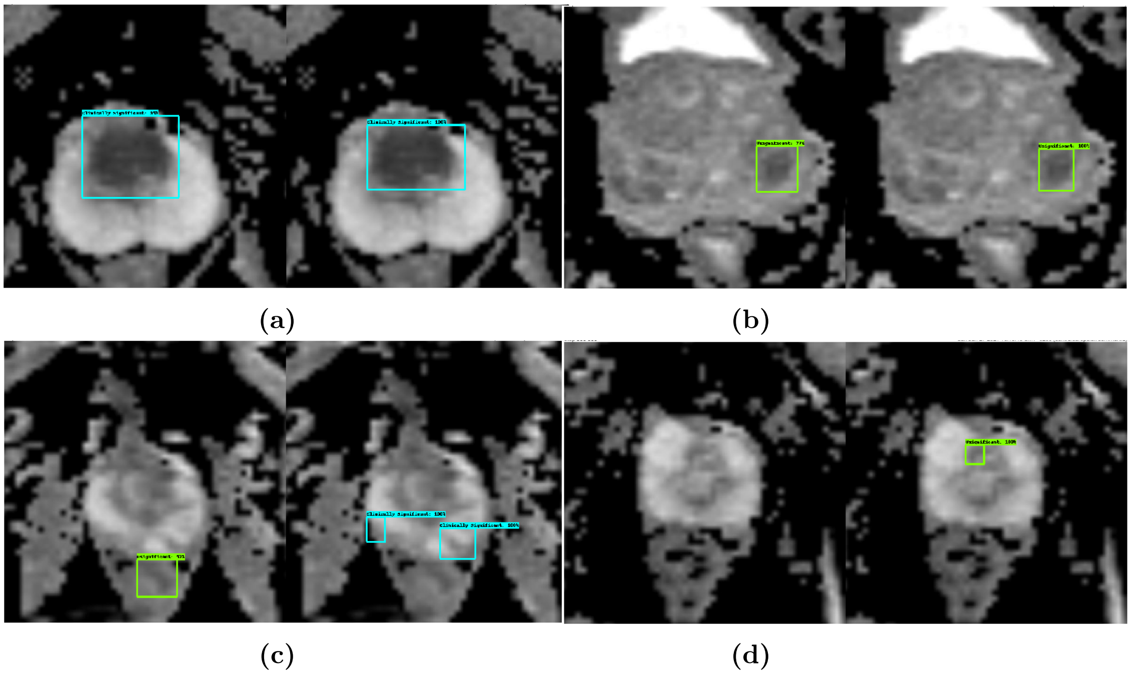
(b)

(c)

(d)

Figure 6.6: Prediction from the prostate PZ. Each sub-figures illustrates prediction to the left and ground truth to the right. Class color: cyan=Clinically Significant, green=Insignificant.

TZ


Clinically Significant: 100%
Insignificant: 100%
Clinically Significant: 100%
Clinically Significant: 100%
Insignificant: 100%

(a)

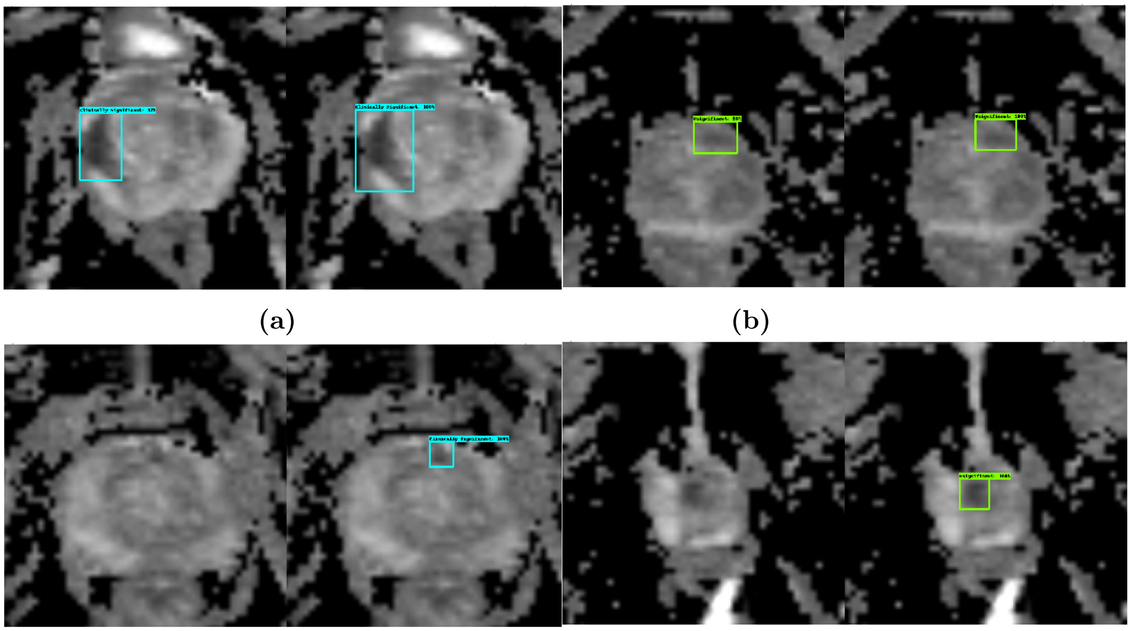
(b)

(c)

(d)

Figure 6.7: Prediction from the prostate TZ. Each sub-figures illustrates prediction to the left and ground truth to the right. Class color: cyan=Clinically Significant, green=Insignificant.

AFS


Clinically Significant: 97%
Clinically Significant: 100%
Insignificant: 59%
Insignificant: 100%
Clinically Significant: 100%
Insignificant: 100%

(a)

(b)

(c)

(d)

Figure 6.8: Prediction from the prostate AFS. Each sub-figures illustrates prediction to the left and ground truth to the right. Class color: cyan=Clinically Significant, green=Insignificant.

6.5 Classification

Previously, the experiment has classified the prostate data relative to the lesions significance, producing a two-class classification. The following experiments look at the CNN performance for data sets that utilize GGG classification, with six classes, and lesion classification, with a single class. The subsequent classification experiments use the Faster R-CNN ResNet-50 model, with the best data augmentation from Table 6.5 (Test 4), on the cropped data set. Table 6.8 shows the performance results for the different classification experiments. Appendix D shows evaluation performance while training and Appendix E illustrates some predictions carried out from the final evaluation, from the different classification experiments.

		Significant Classification		
$AP_{Significant}$[0.5]	**$AP_{Insignificant}$[0.5]**	**mAP[0.5]**	**mAP[0.75]**	**mAR[0.5 : 0.95]**
0.424	0.156	0.291	0.132	0.457

		Lesion Classification		
AP_{Lesion}[0.5]	**mAP[0.5]**	**mAP[0.75]**	**mAR[0.5 : 0.95]**	
0.398	0.398	0.144	0.441	

		GGG Classification		
$AP_{NoBiopsy}$[0.5]	**AP_{GGG1}[0.5]**	**AP_{GGG2}[0.5]**	**AP_{GGG3}[0.5]**	**AP_{GGG4}[0.5]**
0.102	0.064	0.157	0.193	0.013
AP_{GGG5}[0.5]	**mAP[0.5]**	**mAP[0.75]**	**mAR[0.5 : 0.95]**	
0.005	0.090	0.016	0.403	

Table 6.8: Final performance evaluation using PCa Significant, lesion and GGG classification on the Faster R-CNN ResNet-50 model.

Significant Classification

Significant classification has two depending classes, whether the lesion is clinically significant or insignificant, producing mAP[0.5] equal to 0.291. However, if the mean average metrics parameters was weighed relative to the number of examples, the map[0.5] performance value would diminish. Table 6.8 shows that the insignificant class perform worse than the significant class, even though insignificant lesions has far more representations in the prostate data set than clinically significant lesions (Table 4.1).

Lesion Classification

Lesion classification classifies all lesion findings under one single class. CNN model only has to focus on locating the lesion and can ignore the process of classifying the lesions aggressiveness. AP_{Lesion} and mAP[0.5] (both represent the same performance value in this case) achieve a value of 0.398, see Table 6.5. By combining clinically significant and insignificant lesion data, the number of training data for one class increases and the model does not need to assign different classes to the objects, which positively impacts the CNN performance.

GGG Classification

One interesting experiment is to see how the CNN perform detecting PCa relative to different GGG classes. Table 4.1 reveals the GGG class distribution. $AP_{GGG3}[0.5]$ provides the best AP score (0.193) of all the six classes, and $AP_{GGG3}[0.5]$ achieve the second-best AP (0.157). We already knew that the CNN perform better with higher precision on clinically significant lesion than insignificant lesion, thus it was expected one of the GGG 2-5 produced the best AP. However, the most surprising findings from GGG classification experiments is that the $AP_{GGG4}[0.5]$ and $AP_{GGG5}[0.5]$ produce performance value of 0.013 and 0.005. Both $AP_{NoBiopsy}[0.5]$ and $AP_{GGG1}[0.5]$ (Insignificant lesions) outperform the two most aggressive PCa classes (relative to the GGG score). In the total data set of 1281 ground truth labels, GGG 4 and GGG 5 represent 42 and 35 ground truth labels, which is considerably less than any other GGG classes. Lesions that did not undergo a biopsy test have the most ground truth examples (784) and outperform the two highest-ranking GGG classes, even though the lesion is smaller (Table 4.4).

Chapter 7

Discussion

7.1 Image Pre-Processing

Test result from comparing original and cropped images (6.1) highlights the importance of BB area size for a model to detect objects. Other papers also affirm the predicament object detection models has on small-sized objects [80, 32].

The cropping implementation removes as much unnecessary image information as possible and still attaining all lesion in the data set. Removing irrelevant image information allows increasing BB area size by image resizing and procuring a reasonable computational cost. Expanding the BB area size provides better conditions for the CNN to improve detection performance. Cropping and resizing parameters is only a proposed solution in such that CNN performance could increase even further by optimizing these parameters. Especially regarding the data set image size. Because of time limitations, there was not produced experiments using any larger image size than 600x600.

7.2 Models

The most promising object detection architecture to use on the PCa data set is the Fast R-CNN model. This model outperforms both the SSD and R-FCN architectures on the overall performances, especially regarding the clinically significant AP. However, there is some information to take into consideration before comparing these different models. SSD default input image size is default set to 300x300, and training SSD using bigger input image sizes does not affect the model performance, while the training time increases dramatically. However, Tenserflow Object Detection API provide an SSD model using ResNet-100 backbone with a default input size set to 640x640, and produce a higher performance relative to the MobileNet V2 and Inception V2 modules. The result from different backbone networks implementations on the Faster R-CNN model indicates that the SSD ResNet-100 performance improvements are because of the ResNet-100 network rather than the increased input image dimension. The SSD models have less computational cost and latency than Faster R-CNN and R-FCN, but this is somewhat irrelevant relative to the user scenario of PCa detection. Detecting PCa is to be used in hospitals, where computational cost should not be a problem. Also, the data is to be used on images, not on real-time video. Therefore the beneficial aspect of using SSD models is not relevant to PCa detection on MRI images.

Relative to the backbone networks, the ResNet networks produce the best performance. ResNet-50 outperform ResNet-100, which could be because of information loss due to the deeper network ResNet-100 provide. There are limitations when comparing the backbone modules because there is a limited number of available examples for the different CNN architectures in the Tensorflow Object Detection API [35].

7.3 Augmentation

Applying data augmentation with a random probability improves CNN performance by artificially increasing the data set. One of the biggest problem in machine learning, and for this thesis, is the small amount examples data set. As mention in section 4.3, to achieve acceptable performance, there should be around 5000 labelled examples per class. The prostate data set utilizes 1279 labelled object, distributed on 1109 images.

Both contrast and brightness augmentation methods harm CNN performance. Section 5.4 presents the default augmentation parameters used for training, where image brightness continuously decreases, and contrast either decrease or increase when utilized. The results from pixel augmentation on the Prostate data indicate that these hyperparameter values are not the best implementations for a CNN that already has a problem locating the lesion objects because it minimizes its appearance relative to its surroundings. Due to time limitations, brightness and contrast augmentation implementations was discharged after showing inauspicious results, but could positively impact CNN performance utilizing different delta parameter values. Applying pixel augmentation to decreasing the image brightness and contrast is probably better used to optimize performance for an already promising CNN model.

7.4 Anatomical Zones

Study shows DWI/ADC is the optimal technique to detect PCa in PZ, but ADC also shows promising results detecting PCa in TZ [6, 7]. Section 6.4 reproduce the final evaluation (from Section 6.3) of the CNN relative to the anatomical zones, where TZ outperform both PZ and AFS. The classes median BB area size varies among the three zones, which significantly affects the CNN performance. Both TZ and AFS has promising performance result for clinically significant lesions. However, the most promising anatomical zone, PZ, has unfavourable performance results due to small BB area sizes. In addition, experimentation between the zones produces a biased comparison due to the uneven distribution of data examples, class proportion and median BB area size, as seen in Table 6.7.

7.5 Classification

The result from the classification experiments, in Section 6.5, highlights the importance of data examples for each class. It is not easy to pinpoint the optimal data set magnitude to culminate the best CNN performance for PCa detection. Ian Goodwill book *Deep Learning* states that there should be at least 5000 ground truth examples for achieving acceptable performance [30], which significant, lesion and GGG classification fails to reach for any of the classes. Utilizing more classes when training a CNN increases the computational cost, which contributes to reducing CNN performance. The data set is too small to produce any fair comparison or promising results for the carried out experiments, especially regarding the GGG classification.

One future approach could be first to classify lesion based on significance. If the CNN model classifies the lesion as clinically significant, a new detection could be applied for predicting the GGG score.

7.6 Limitation

This section will discuss some of this thesis limitations.

7.6.1 Data Set Size

The data set contains ground truth for 200 individual patients, with 299 lesion objects distributed on 1109 two-dimensional slices. There should be at least 5000 labelled ground truth per class to achieve acceptable CNN performance [30]. However, the data set contains ground truth for 914 insignificant and 367 clinically significant prostate lesions, producing 1281 labelled examples. For supervised learning, this is an insufficient number of instance that limits optimal CNN performance. This is especially the case for GGG classification, with even fewer data examples per class, see Table 4.1. Evaluating CNN on the validation and test data set produces different performance result as a product of a small imbalanced data set. To improve the networks ability to generalize, the number of patient and unique lesion objects needs to increase dramatically. However, collecting a substantial volume of biomedical data and labelling the lesion location and classification is challenging, requiring approval from patients and supervision from specialists.

The data set also have an unequal classification distribution, culminating in an inadequate predictive performance representation, especially for the clinically significant class that represent only 28.65% of all the labelled examples in contrast to the insignificant class that obtains 71.35% (Table 4.1).

7.6.2 Data Set Error

This project started using T2W images with a Yolov3 architecture, where the trained CNN performance result was unsatisfactory, achieving an mAP[0.5] of under 0.001 [81]. With no prior experience with PCa locations or medical MRI images, the first impression was that there was something wrong with the object detection model rather than the data set. Image pre-processing techniques such as histogram equalizer, cropping and resizing was implemented with minor performance improvements.

In search of better detection performance, T2W images were replaced with the ADC images, where the mask image is 90 degrees rotated relative to the ADC MRI image, as mentioned in Section 4.2.7. After a while, I also notice that the slice order was inverted, which was difficult to see since most of the lesion objects appear in the centre of the slice order number. Thus, lesion objects appearing in the middle of the slice number have the same slice number even though the slice order is inverted, and lesion objects located around the centre slice number seem correct to some degree because they often have similar shape and localization as the neighbouring slice numbers. Thus, adjusting the mask data representation improved CNN performance drastically. However, these mask data implementations are only a proposed solution. There could still be other data issues present in the mask data set.

Chapter 8

Conclusion and future work

8.1 Conclusion

This thesis explores the potential to use CNN to detect prostate lesions in ADC images and classify PCa aggressiveness. One-stage and two-stage CNN models and different backbones architecture was trained to examine detection performance on the prostate data set. The two-stage Faster R-CNN architecture, with ResNet-50 backbone, produced the most promising results.

Experiments show that the CNN models have difficulty detecting small objects, where the average prostate lesion constitute a small area size of the image. However, the cropping image pre-processing method and increasing image shape dimension produced a large BB area size that improved model detection performance. The main challenge with this thesis is the insufficient data set size and unbalanced class distribution, which prohibit this thesis to obtain impartial CNN performance results. This is especially the case for GGG classification and anatomical zone experiments, which reduce the number of class examples even further. There is also an instance of mask data error, which stagnated this thesis work progress and possibly diminished CNN performance. On the other hand, applying the optimal data augmentation improved CNN performance by artificially increasing the data set size.

Utilizing the best CNN architectures, image pre-processing techniques, data augmentation methods and hyperparameters achieved an AP[0.5] of 0.424 for significant and 0.156 for insignificant lesions. The model gave the most promising performance on the prostate TZ, producing an AP[0.5] of 0.764 for significant and 0.302 for insignificant lesions. Experiments show that CNN struggles to determine the significance of the lesions to some degree. Eliminating the classification process and mainly focusing on detecting lesion location assembles an AP[0.5] of 0.398.

8.2 Future Directions

The most important future direction is to ensure that the prostate mask data accurately represent the given MRI slice, in which a radiologist verifies lesions localization and classification. One of the recurring problems throughout this thesis has been the insufficient data set size. There are other PCa data set with ground truth labels that can be applied to CNN training, such as the Valencia Oncology Institute Foundation (IVO) data set [82].

This thesis shows the great impact data augmentation has on CNN performance. Applying more augmentation techniques could improve performance even further. Future experiments could test "cut-paste" augmentation, removing objects from images using segmentation and placing them in another image with a different background. [83].

There is an opportunity to draw PCa conclusion combining all ADC images from a given patient. This thesis looks at all MRI slices individually, even though a lesion object usually exists in multiple MRI slices with similar shape and location for a given patient. This information can guide the CNN to detect lesions in the other MRI slices for the given patient. In addition, combining other MRI forms, such as T2W, can improve the CNN ability to conclude PCa findings for a patient even further [6].

Another approach for PC detection, which has been conducted in another paper, is first to detect the prostate gland and then search for the lesion inside the gland, consider a great proportion of lesion findings is located inside the prostate gland [82].

Utilizing Semi-supervised learning, such as the popular machine learning approach Generative Adversarial Network (GAN), could help improve PCa detection by generating more data, especially regarding that gathering labelled medical data is a demanding task [68].

Meta CNN architectures, utilized in this thesis, shows promising detection performance on large-size objects. Future experiments could test other CNN architectures, such as Retina U-Net architecture which focus on detecting small objects in medical images [84]

List of Figures

List of Tables

Bibliography

[1] World Health Organization. Prostate. https://gco.iarc.fr/today/data/factsheets/cancers/27-Prostate-fact-sheet.pdf, December 2020. (Accessed on 02/22/2021).

[2] Bristol-Myers Squibb Oslo Economics. Kostnader for pasientene, helsetjenesten og samfunnet. *Kreft i Norge*, pages 125–134, 2016.

[3] Yu-Peng Wu, Xiao-Dong Li, Zhi-Bin Ke, Shao-Hao Chen, Ping-Zhou Chen, Yong Wei, Jin-Bei Huang, Xiong-Lin Sun, Xue-Yi Xue, Qing-Shui Zheng, et al. Risk factors for infectious complications following transrectal ultrasound-guided prostate biopsy. *Infection and drug resistance*, 11:1491, 2018.

[4] Kimia Kohestani, Jonas Wallström, Niclas Dehlfors, Ole Martin Sponga, Marianne Månsson, Andreas Josefsson, Sigrid Carlsson, Mikael Hellström, and Jonas Hugosson. Performance and inter-observer variability of prostate mri (pi-rads version 2) outside high-volume centres. *Scandinavian journal of urology*, 53(5):304–311, 2019.

[5] Centers for Disease Control and Prevention. What are the benefits and harms of screening for prostate cancer? https://www.cdc.gov/cancer/prostate/basic_info/benefits-harms.htm. (Accessed on 06/03/2021).

[6] Hakmin Lee, Sung Il Hwang, Hak Jong Lee, Seok-Soo Byun, Sang Eun Lee, and Sung Kyu Hong. Diagnostic performance of diffusion-weighted imaging for prostate cancer: Peripheral zone versus transition zone. *PloS one*, 13(6):e0199636, 2018.

[7] Rhiannon van Loenhout, Frank Zijta, Robin Smithuis, and Ivo Schoots. Prostate cancer - pi-rads v2. https://radiologyassistant.nl/abdomen/prostate/prostate-cancer-pi-rads-v2, January 2018. (Accessed on 05/07/2021).

[8] The American Cancer Society medical and editorial content team. What is prostate cancer? https://www.cancer.org/cancer/prostate-cancer/

about/what-is-prostate-cancer.html, August 2019. (Accessed on 03/31/2021).

[9] American Urological Association. Medical student curriculum: Benign prostatic hypertrophy (bph). https://www.auanet.org/education/auauniversity/for-medical-students/medical-students-curriculum/medical-student-curriculum/bph, February 2021. (Accessed on 05/24/2021).

[10] Norsk Helseinformatikk. Psa-skal du teste deg? https://nhi.no/sykdommer/kreft/diverse/psa-skal-du-teste-deg/, February 2021. (Accessed on 04/20/2021).

[11] Scott Gottlieb. Study shows poor reliability of prostate cancer test, 2003.

[12] Stefano Ciatto. Reliability of psa testing remains unclear. *British Medical Journal*, 327(7417):750, 2003.

[13] Norsk Helseinformatikk. Prostatakreft. https://nhi.no/sykdommer/kreft/mannlige-kjonnsorganer-kreft/prostatakreft/, May 2021. (Accessed on 05/25/2021).

[14] Wikipedia. Rectal examination. https://en.wikipedia.org/wiki/Rectal_examination, April 2021. (Accessed on 05/25/2021).

[15] The American Cancer Society medical and editorial content team. Tests for prostate cancer. https://www.cancer.org/cancer/prostate-cancer/detection-diagnosis-staging/how-diagnosed.html, Feburary 2021. (Accessed on 05/25/2021).

[16] Farco. Digital rectal examination (dre). https://www.farco.de/en/fuer-patienten/urologische-untersuchungen/digitale-rektale-untersuchung-dru. (Accessed on 04/19/2021).

[17] Baris Turkbey, Andrew B Rosenkrantz, Masoom A Haider, Anwar R Padhani, Geert Villeirs, Katarzyna J Macura, Clare M Tempany, Peter L Choyke, Francois Cornud, Daniel J Margolis, et al. Prostate imaging reporting and data system version 2.1: 2019 update of prostate imaging reporting and data system version 2. *European urology*, 76(3):340–351, 2019.

[18] Brett Delahunt, Rose J Miller, John R Srigley, Andrew J Evans, and Hemamali Samaratunga. Gleason grading: past, present and future. *Histopathology*, 60(1):75–86, 2012.

[19] Jonathan I Epstein, Lars Egevad, Mahul B Amin, Brett Delahunt, John R Srigley, and Peter A Humphrey. The 2014 international society of urological pathology (isup) consensus conference on gleason grading of prostatic carcinoma. *The American journal of surgical pathology*, 40(2):244–252, 2016.

[20] Dr. C.H. Weaver M.D. Prostate cancer: What you need to know about the gleason score. `https://news.cancerconnect.com/prostate-cancer/prostate-cancer-what-you-need-to-know-about-the-gleason-score`, November 2020. (Accessed on 05/25/2021).

[21] National Institue of Biomedical Imaging and Bioengineering. Magnetic resonance imaging (mri). `https://www.nibib.nih.gov/science-education/science-topics/magnetic-resonance-imaging-mri`. (Accessed on 06/04/2021).

[22] Denis Le Bihan. Apparent diffusion coefficient and beyond: what diffusion mr imaging can tell us about tissue structure, 2013.

[23] Dr Patrick Rock and Dr Mohammad Taghi Niknejad. Apparent diffusion coefficient. `https://radiopaedia.org/articles/apparent-diffusion-coefficient-1`. (Accessed on 06/04/2021).

[24] Akio Ogura, Katsumi Hayakawa, Tosiaki Miyati, and Fumie Maeda. Imaging parameter effects in apparent diffusion coefficient determination of magnetic resonance imaging. *European journal of radiology*, 77(1):185–188, 2011.

[25] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[26] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[27] Osman Semih Kayhan and Jan C van Gemert. On translation invariance in cnns: Convolutional layers can exploit absolute spatial location. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14274–14285, 2020.

[28] Jason Brownlee. Supervised and unsupervised machine learning algorithms. https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/, August 2020. (Accessed on 05/16/2021).

[29] Julianna Delua. Supervised vs. unsupervised learning: What's the difference? https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning, March 2021. (Accessed on 05/16/2021).

[30] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep Learning*, volume 1. MIT press Cambridge, 2016.

[31] Hao Gao. Object localization in overfeat. https://towardsdatascience.com/object-localization-in-overfeat-5bb2f7328b62, August 2017. (Accessed on 02/24/2021).

[32] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*, 2019.

[33] Jason Brownlee. *Deep Learning for Computer Vision - Image Classification, Object Detection and Face Recognition in Python.* 2020. https://machinelearningmastery.com/deep-learning-for-computer-vision/.

[34] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[35] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7310–7311, 2017.

[36] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE*

transactions on pattern analysis and machine intelligence, 39(6):1137–1149, 2016.

[38] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 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, 2014.

[39] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[40] Ahmed Fawzy Gad. Faster r-cnn explained for object detection tasks. `https://blog.paperspace.com/faster-r-cnn-explained-object-detection/`, December 2020. (Accessed on 05/21/2021).

[41] Alegion. Faster r-cnn. `https://www.alegion.com/faster-r-cnn`. (Accessed on 05/26/2021).

[42] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. *arXiv preprint arXiv:1605.06409*, 2016.

[43] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[44] Wei Liu, Olga Russakovsky, Jia Deng, Fei-Fei Li, and Alex Berg. Imagenet. `https://image-net.org/challenges/LSVRC/2015/`, December 2015. (Accessed on 06/07/2021).

[45] Delwar Hossain, Masudul Haider Imtiaz, Tonmoy Ghosh, Viprav Bhaskar, and Edward Sazonov. Real-time food intake monitoring using wearable egocnetric camera. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 4191–4195. IEEE, 2020.

[46] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[47] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[48] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[49] Bart Selman and Carla P Gomes. Hill-climbing search. *Encyclopedia of cognitive science*, 81:82, 2006.

[50] Anton Morgunov. Tensorflow object detection api: Best practices to training, evaluation & deployment. `https://neptune.ai/blog/`, May 2021. (Accessed on 04/20/2021).

[51] Uniqtech. Understand jaccard index, jaccard similarity in minutes. `https://medium.com/data-science-bootcamp/understand-jaccard-index-jaccard-similarity-in-minutes-25a703fbf9d7`, August 2019. (Accessed on 03/22/2021).

[52] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 658–666, 2019.

[53] Rafael Padilla. Most popular metrics used to evaluate object detection algorithms. `https://github.com/rafaelpadilla/Object-Detection-Metrics`. (Accessed on 03/23/2021).

[54] Rafael Padilla, Wesley L Passos, Thadeu LB Dias, Sergio L Netto, and Eduardo AB da Silva. A comparative analysis of object detection metrics with a companion open-source toolkit. *Electronics*, 10(3):279, 2021.

[55] Rafael Padilla, Sergio L Netto, and Eduardo AB da Silva. A survey on performance metrics for object-detection algorithms. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 237–242. IEEE, 2020.

[56] Guido Van Rossum and et al. Python programming language. In *USENIX annual technical conference*, volume 41, page 36, 2007.

[57] NVIDIA. Nvidia tesla v100 gpu accelerator. `https://images.nvidia.com/content/technologies/volta/pdf/tesla-volta-v100-datasheet-letter-fnl-web.pdf`, March 2018. (Accessed on 06/08/2021).

[58] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016.

[59] The NumPy community. What is numpy? — numpy v1.22.dev0 manual. `https://numpy.org/devdocs/user/whatisnumpy.html`. (Accessed on 05/27/2021).

[60] The Pandas Development Team. pandas documentation — pandas 1.2.4 documentation. `https://pandas.pydata.org/docs/index.html`, April 2021. (Accessed on 05/27/2021).

[61] Insight Software Consortium. Simpleitk - about. `https://simpleitk.org/about.html`. (Accessed on 05/27/2021).

[62] Tracy Nolan. Spie-aapm-nci prostatex challenges - the cancer imaging archive (tcia) public access - cancer imaging archive wiki. `https://wiki.cancerimagingarchive.net/display/Public/SPIE-AAPM-NCI+PROSTATEx+Challenges`, January 2021. (Accessed on 03/22/2021).

[63] Jelle Barentsz. About jelle barentsz. `http://www.mri-prostate-barentsz.nl/`. (Accessed on 04/06/2021).

[64] Siemens Healthineers. Magnetom trio, a tim system 3t eco – used mri machine. `https://www.siemens-healthineers.com/en-us/refurbished-systems-medical-imaging-and-therapy/ecoline-refurbished-systems/magnetic-resoncance-imaging-ecoline/magnetom-trio-3t-eco`. (Accessed on 04/09/2021).

[65] Siemens Healthineers. Magnetom skyra. https://www.siemens-healthineers.com/magnetic-resonance-imaging/3t-mri-scanner/magnetom-skyra. (Accessed on 04/09/2021).

[66] Renato Cuocolo, Arnaldo Stanzione, Anna Castaldo, Davide Raffaele De Lucia, and Massimo Imbriaco. Quality control and whole-gland, zonal and lesion annotations for the prostatex challenge public dataset. *European Journal of Radiology*, 138:109647, 2021.

[67] Renato Cuocolo. Lesion and prostate masks for the prostatex training dataset, after a lesion-by-lesion quality check. https://github.com/rcuocolo/PROSTATEx_masks, September 2020. (Accessed on 01/19/2021).

[68] Alvaro Fernandez-Quilez, Steinar Valle Larsen, Morten Goodwin, Thor Ole Gulsrud, Svein Reidar Kjosavik, and Ketil Oppedal. Improving prostate whole gland segmentation in t2-weighted mri with synthetically generated data. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 1915–1919. IEEE, 2021.

[69] Jason Brownlee. What is the difference between test and validation datasets? https://machinelearningmastery.com/difference-test-validation-datasets/, August 2020. (Accessed on 04/09/2021).

[70] Junko Ota, Kensuke Umehara, Naoki Ishimaru, Takayuki Ishida, et al. Application of sparse-coding super-resolution to 16-bit dicom images for improving the image resolution in mri. *Open Journal of Medical Imaging*, 7(04):144, 2017.

[71] Jim Frost. Percentiles: Interpretations and calculations - statistics by jim. https://statisticsbyjim.com/basics/percentiles/. (Accessed on 05/26/2021).

[72] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[73] The SciPy community. numpy.savez — numpy v1.20 manual. https://numpy.org/doc/stable/reference/generated/numpy.savez.html, January 2021. (Accessed on 04/23/2021).

[74] Darcy Mason. pydicom · pypi. `https://pypi.org/project/pydicom/0.9.7/`, March 2012. (Accessed on 06/04/2021).

[75] Archana Singh, Sanjana Yadav, and Neeraj Singh. Contrast enhancement and brightness preservation using global-local image enhancement techniques. In *2016 fourth international conference on parallel, distributed and grid computing (PDGC)*, pages 291–294. IEEE, 2016.

[76] Fang Liu Shuyuan Yang Lingling Li Zhixi Feng Rong Qu Licheng Jiao, Fan Zhang. A survey of deep learning-based object detection. pages 128837–128868, October 2019. `https://ieeexplore.ieee.org/document/8825470`.

[77] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[78] Mark Everingham and John Winn. The pascal visual object classes challenge 2012 (voc2012) development kit. *Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep*, 8, 2011.

[79] Jason Brownlee. Gradient descent with momentum from scratch. `https://machinelearningmastery.com/gradient-descent-with-momentum-from-scratch/`, February 2021. (Accessed on 06/21/2021).

[80] Nhat-Duy Nguyen, Tien Do, Thanh Duc Ngo, and Duy-Dinh Le. An evaluation of deep learning methods for small object detection. *Journal of Electrical and Computer Engineering*, 2020, 2020.

[81] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[82] Oscar J Pellicer-Valero, José L Marenco Jiménez, Victor Gonzalez-Perez, Juan Luis Casanova Ramón-Borja, Isabel Martín García, María Barrios Benito, Paula Pelechano Gómez, José Rubio-Briones, María José Rupérez, and José D Martín-Guerrero. Deep learning for fully automatic detection, segmentation, and gleason grade estimation of prostate cancer in multiparametric magnetic resonance images. *arXiv preprint arXiv:2103.12650*, 2021.

[83] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1301–1310, 2017.

[84] Paul F Jaeger, Simon AA Kohl, Sebastian Bickelhaupt, Fabian Isensee, Tristan Anselm Kuder, Heinz-Peter Schlemmer, and Klaus H Maier-Hein. Retina u-net: Embarrassingly simple exploitation of segmentation supervision for medical object detection. In *Machine Learning for Health Workshop*, pages 171–183. PMLR, 2020.

Appendix A

Results - Cropping

Graphs of metric performance on validation data set while training. All test are trained for 350 000 steps. See table A.1 for color code.
Visualization of the object prediction, when evaluating the model performance, are also included. Left image is the prediction and right image is the ground truth.

Color	Blue	Orange
Test number	Test 1	Test 2

Table A.1: Color code representation of the different test numbers.

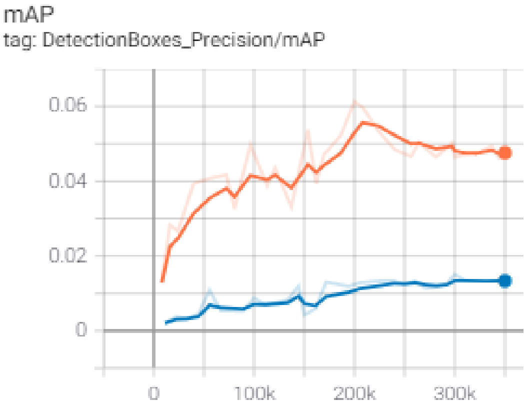
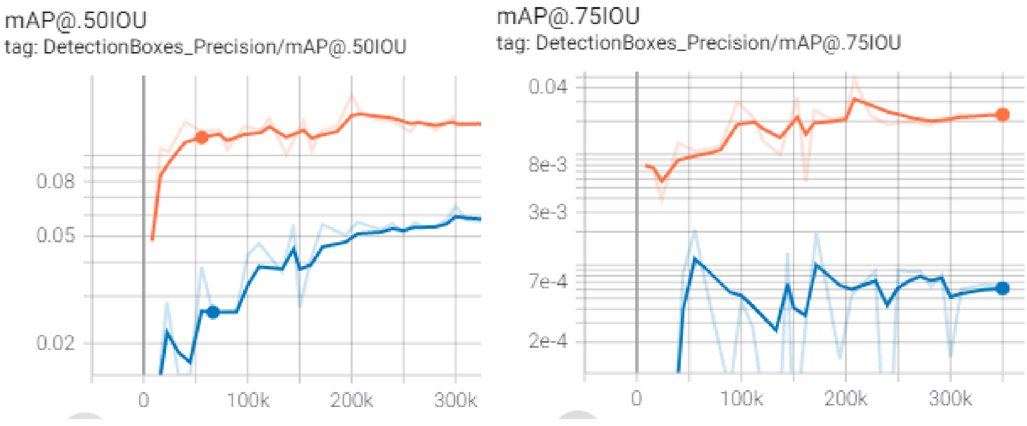


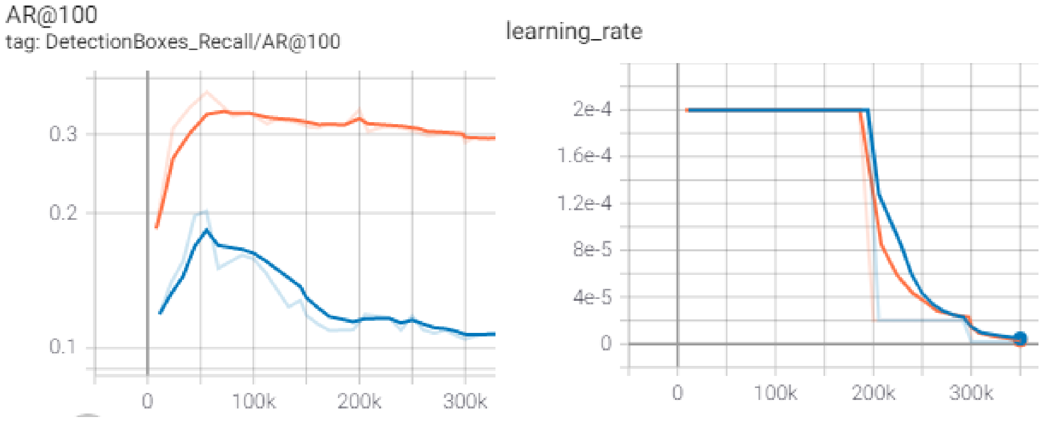
Figure A.1: MAP[0.5:0.95]



(a) MAP[0.5]

(b) MAP[0.75]

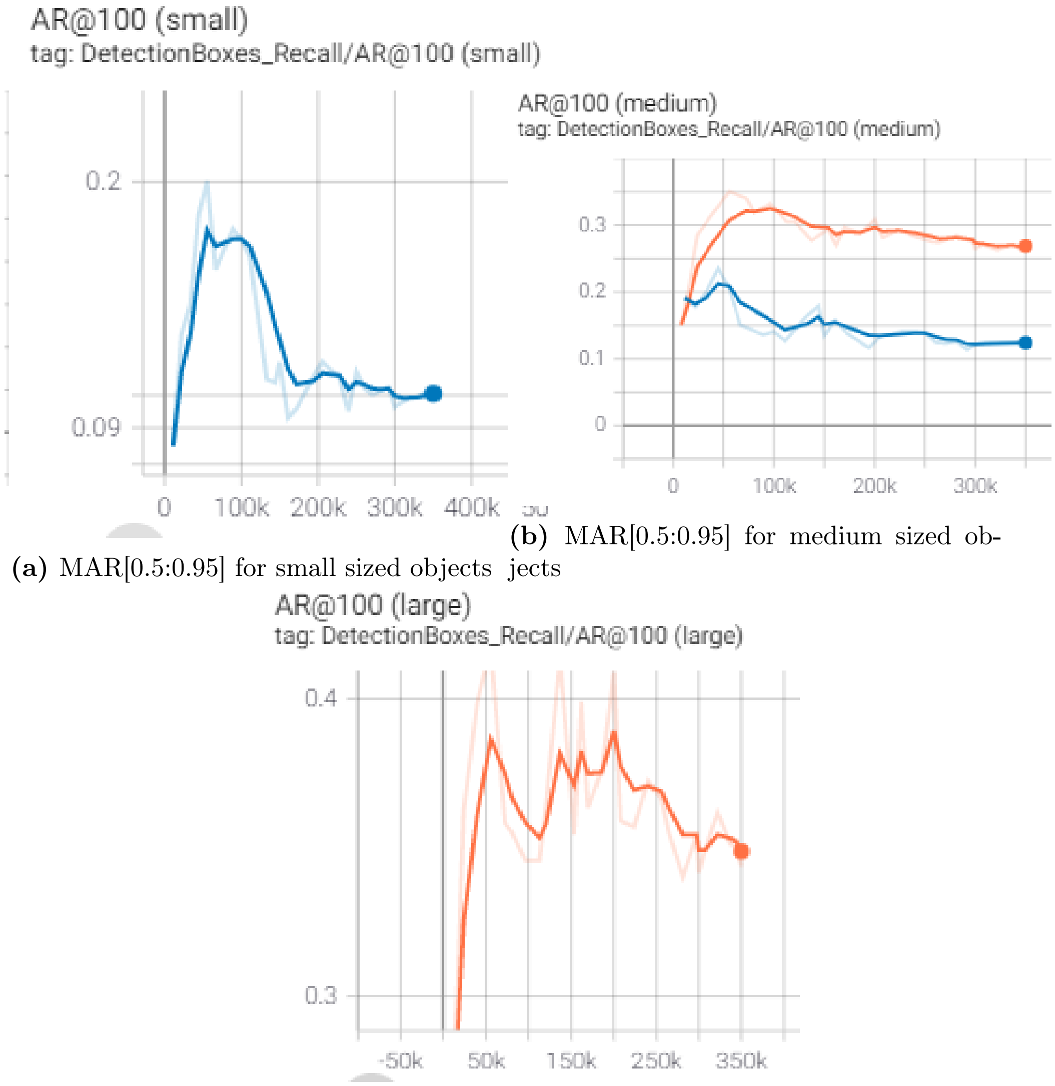
Figure A.2: MAP[0.5] and mAP[0.75] for orginal and cropped data sets.



(a) MAR[0.5:0.95] for all sized objects

(b) Learning rate

Figure A.3: MAR[0.5:0.95] for all sized objects and the learning rate for both tests.



(a) MAR[0.5:0.95] for small sized objects

(b) MAR[0.5:0.95] for medium sized objects

(c) MAR[0.5:0.95] for large sized objects

Figure A.4: MAR[0.5:0.95] for small, medium and large sized objects for original and cropped data sets.

Appendix B

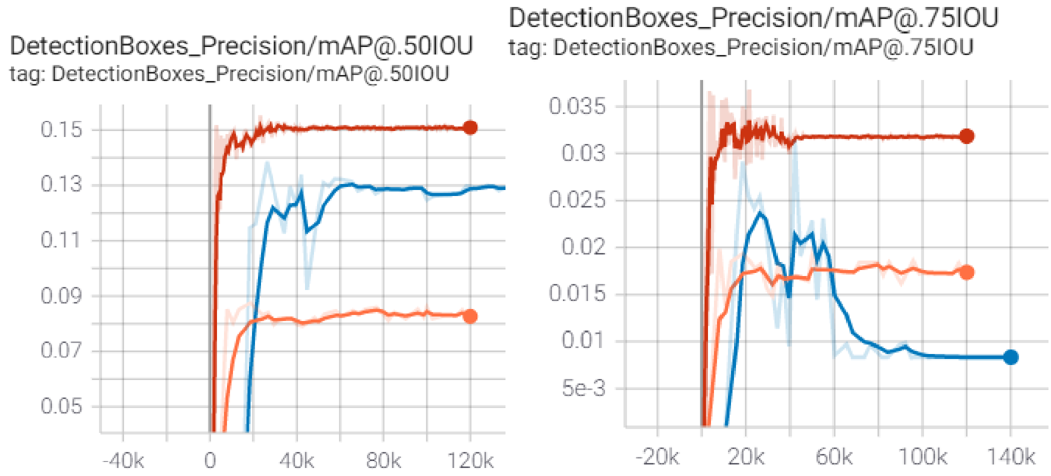
Results - Models/Backbones

Appendix shows the models performance (evaluated on validation data set) while training. The Cropped data set are utilized to test different object detection architectures and module backbones. Appendix is split into two sections, one for SSD networks and the other for Faster R-CNN and R-FCN networks.

SSD

Color	Orange	Blue	Red
Backbone	Inception v2	MobileNet v2	ResNet-50

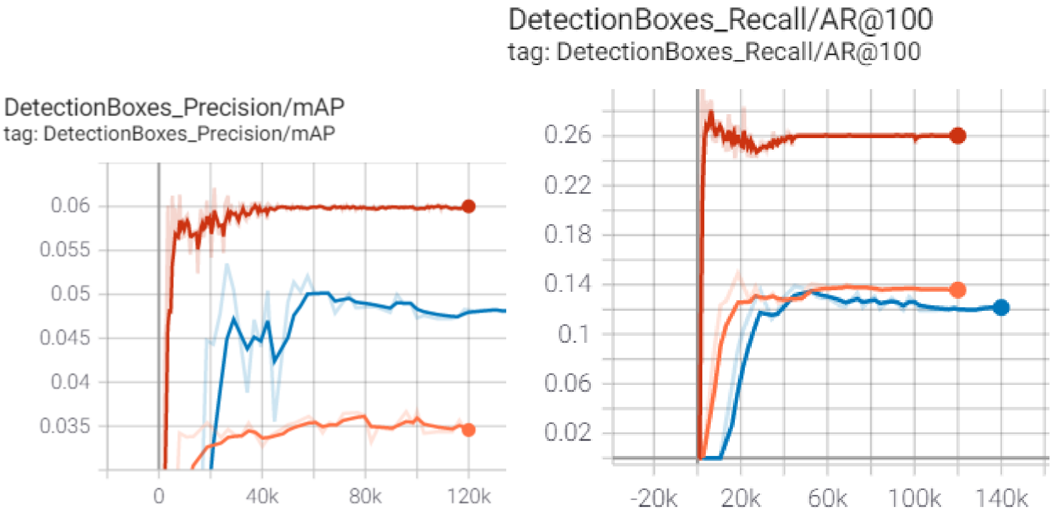
Table B.1: Color representation in the evaluation graphs.



(a) MAP[0.5]

(b) MAP[0.75]

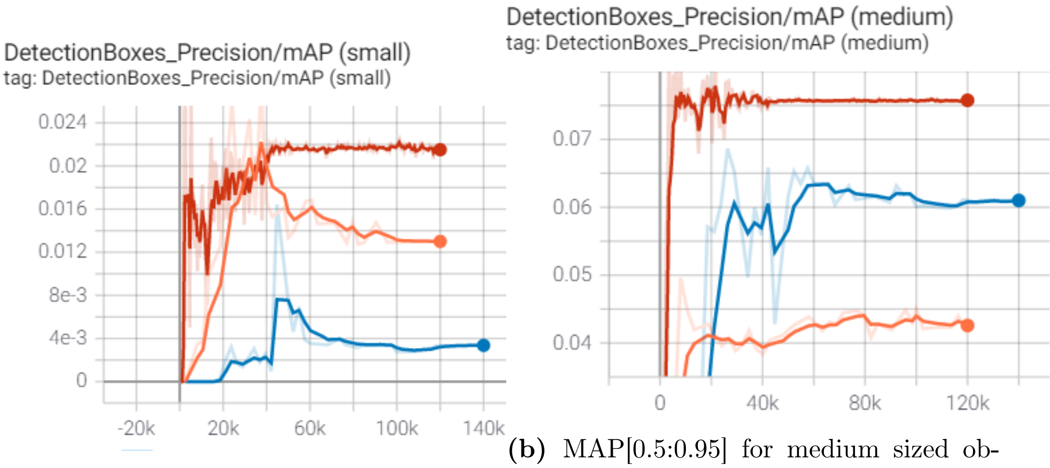
Figure B.1: MAP[0.5] and mAP[0.75] for SSD models.



(a) MAP[0.5:0.95]

(b) MAR[0.5:0.95]

Figure B.2: MAP[0.5:0.95] and mAR[0.5:0.95] for all object sizes.



(a) MAP[0.5:0.95] for small sized objects.

(b) MAP[0.5:0.95] for medium sized objects.

Figure B.3: MAP[0.5:0.95] for small and medium sized objects.

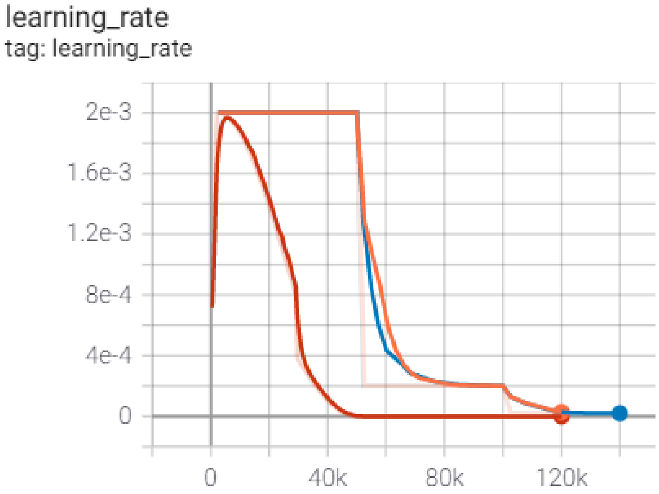
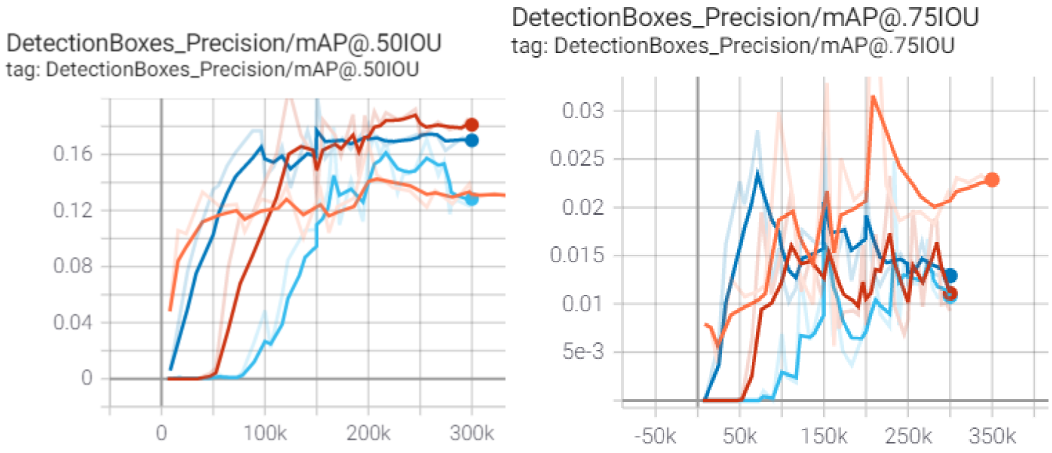


Figure B.4: Learning rate while training.

Faster R-CNN and R-FCN

Color	Orange	Blue	Red	Cyan
Architecture	Faster R-CNN	Faster R-CNN	Faster R-CNN	R-FCN
Backbone	Inception v2	ResNet-50	ResNet-100	ResNet100

Table B.2: Color representation in the evaluation graphs.



(a) MAP[0.5]

(b) MAP[0.75]

Figure B.5: MAP[0.5] and mAP[0.75] for the Faster R-CNN and R-FCN models.

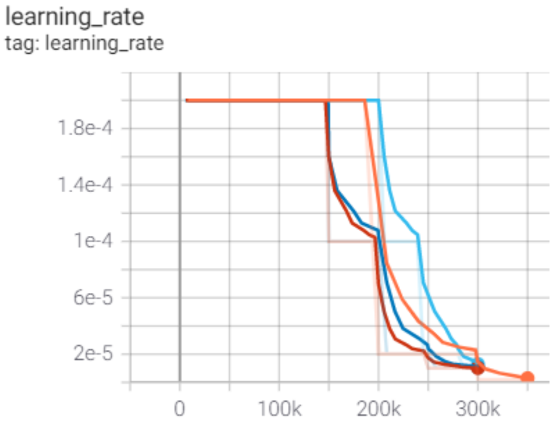
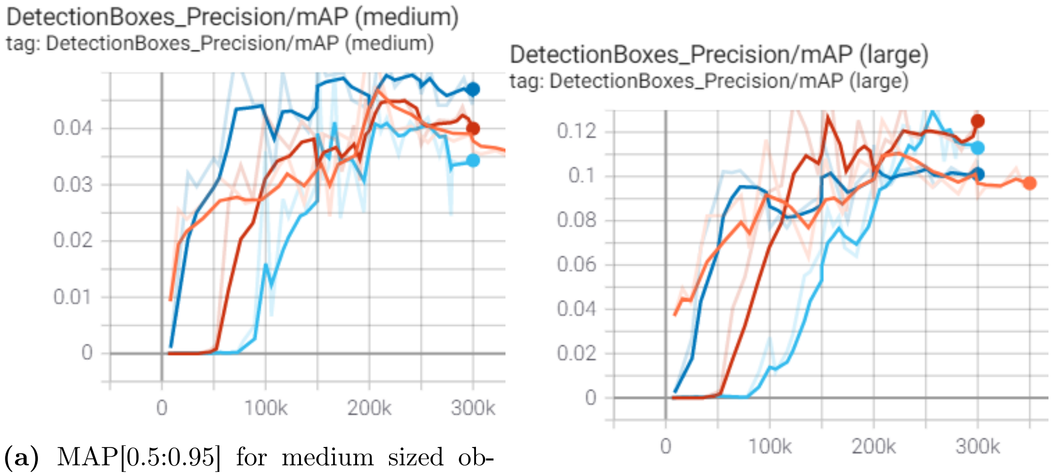


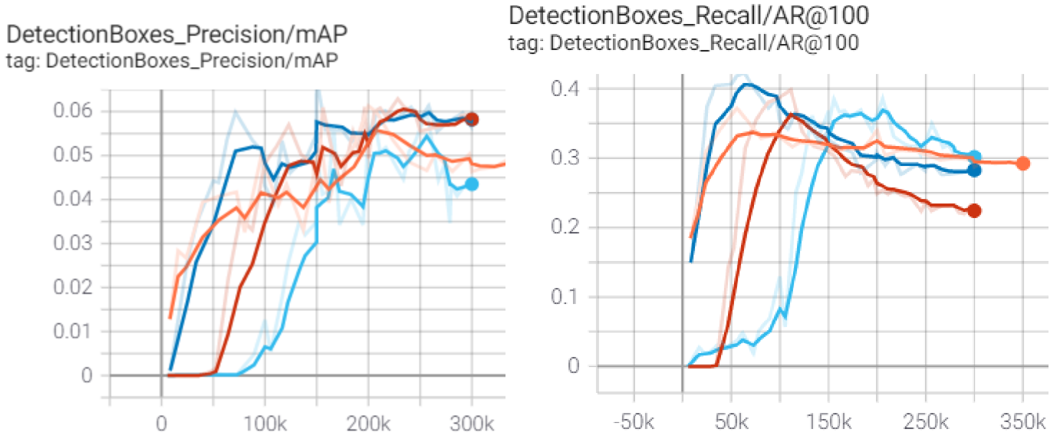
Figure B.6: Learning rate while training.



(a) MAP[0.5:0.95] for medium sized objects.

(b) MAP[0.5:0.95] for large sized objects.

Figure B.8: MAP[0.5:0.95] for medium and large sized objects.



(a) MAP[0.5:0.95]

(b) MAR[0.5:0.95]

Figure B.7: MAP[0.5:0.95] and mAR[0.5:0.95] for all object sizes.

Appendix C

Results - Augmentation

Appendix shows the models performance (evaluated on validation data set) while training. All experiments are trained for 300 000 steps. The Cropped data set are utilized to test the affect on the Faster R-CNN ResNet 50 performance using different augmentation implementation. Table C.1 explains the graphs color code.

Color	Orange	Blue	Red	Cyan	Green	Pink
Augmentation Implementations	Horizontal flip	Horizontal flip Brightness Contrast	Horizontal flip Vertical flip Rotation (90 degrees)	Horizontal flip Crop	Horizontal flip Vertical flip Rotation (90 degrees) Crop	Horizontal flip Vertical flip Rotation (90 degrees) Crop Brightness Contrast

Table C.1: Color code representation of the different augmentation experiments.

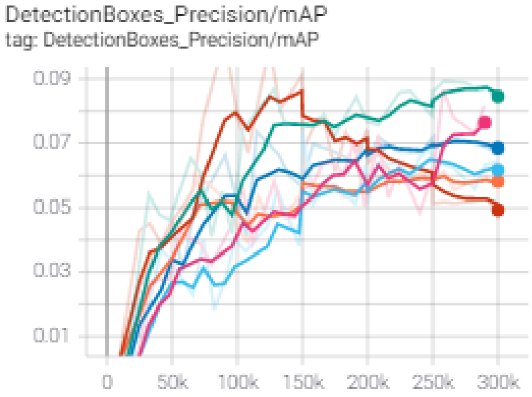
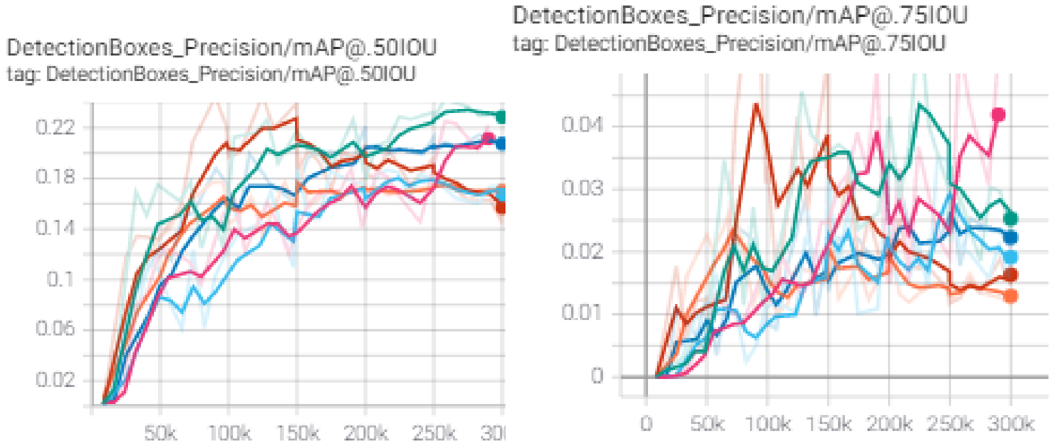


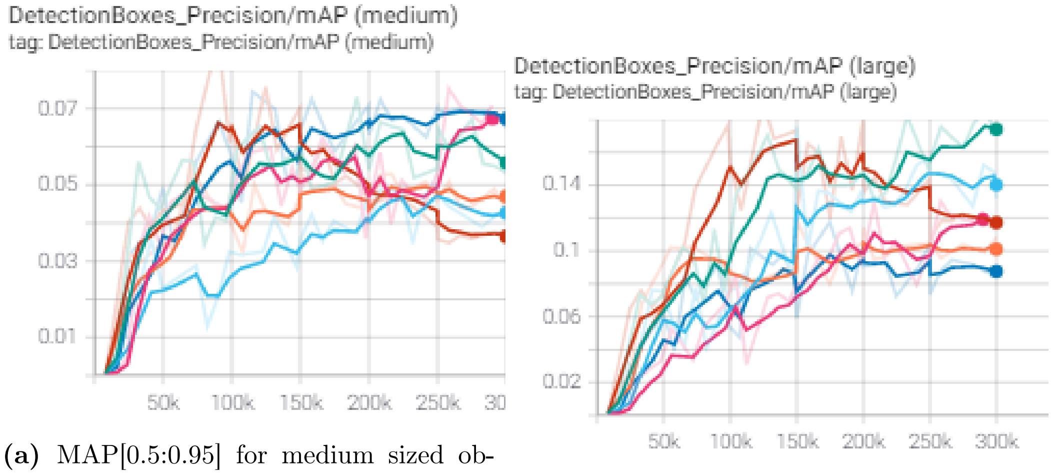
Figure C.1: MAP[0.5:0.95]



(a) MAP[0.5]

(b) MAP[0.75]

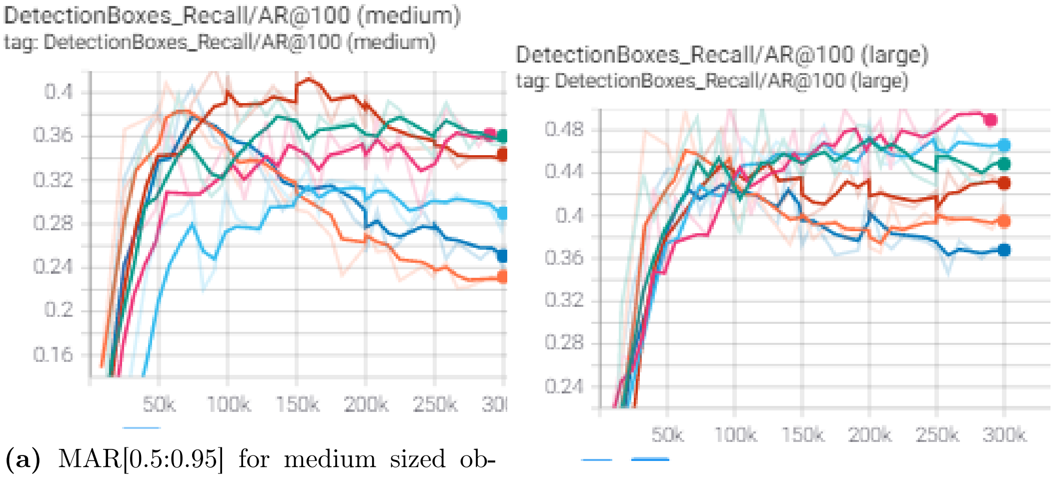
Figure C.2: MAP[0.5] and mAP[0.75] from training the different augmentation methods.



(a) MAP[0.5:0.95] for medium sized objects

(b) MAP[0.5:0.95] for large sized objects

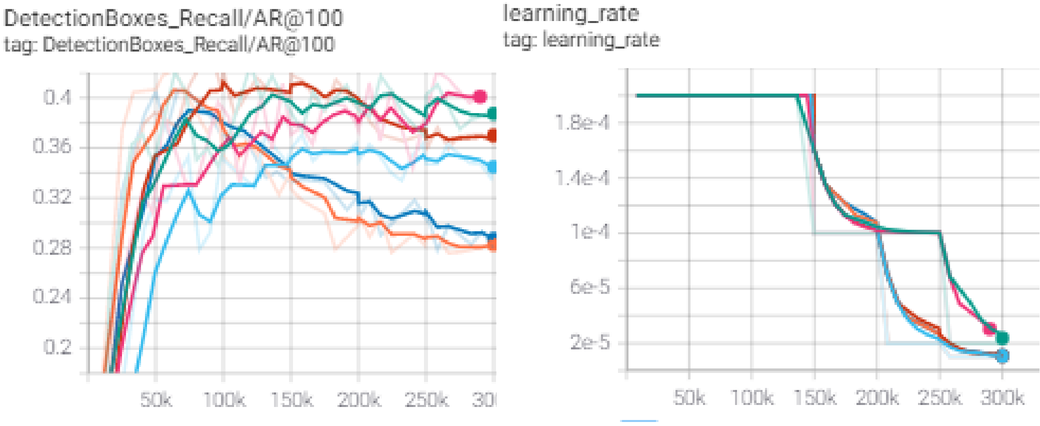
Figure C.3: MAP[0.5:0.95] for medium and large sized objects.



(a) MAR[0.5:0.95] for medium sized objects

(b) MAR[0.5:0.95] for large sized objects

Figure C.4: MAR[0.5:0.95] for medium and large sized objects.



(a) MAR[0.5:0.95] for all sized objects

(b) Learning rate while training

Figure C.5: Overall mAR[0.5:0.95] and learning rate while training.

Appendix D

Results - Classification

Appendix shows the models performance (evaluated on validation data set) while training.The Cropped data set are utilized to test the affect on the Faster R-CNN ResNet 50 performance, with data augmentation, using significant (two classes), GGG (six classes) and lesion classification (one class) . Table C.1 explains the graphs color code.

Color	Orange	Blue	Red
Classification	Significant	GGG	Lesion
Class numbers	2	6	1

Table D.1: Color code representation of the different classification experiments.

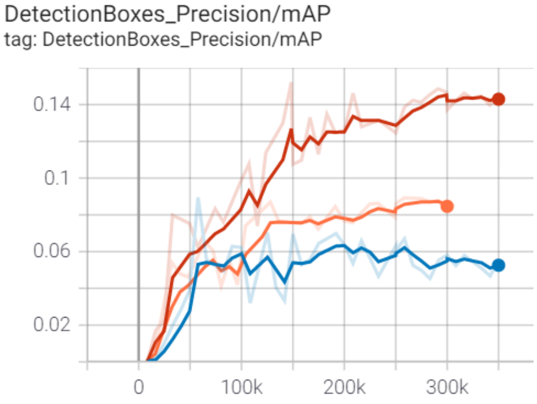
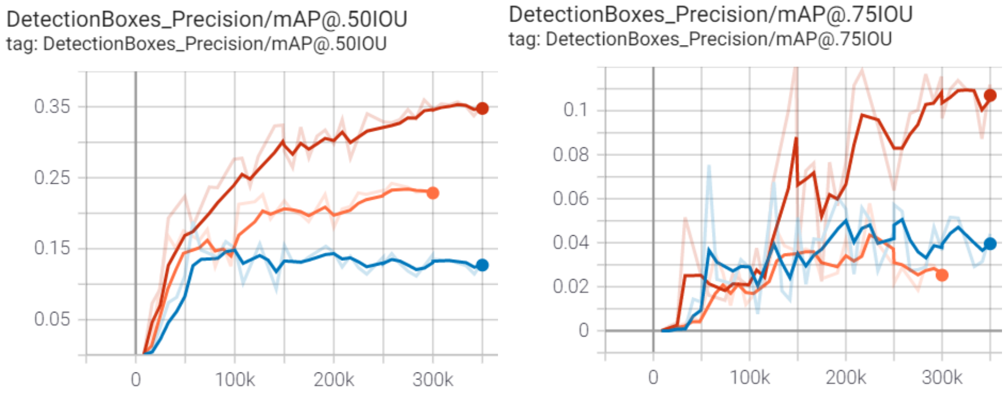


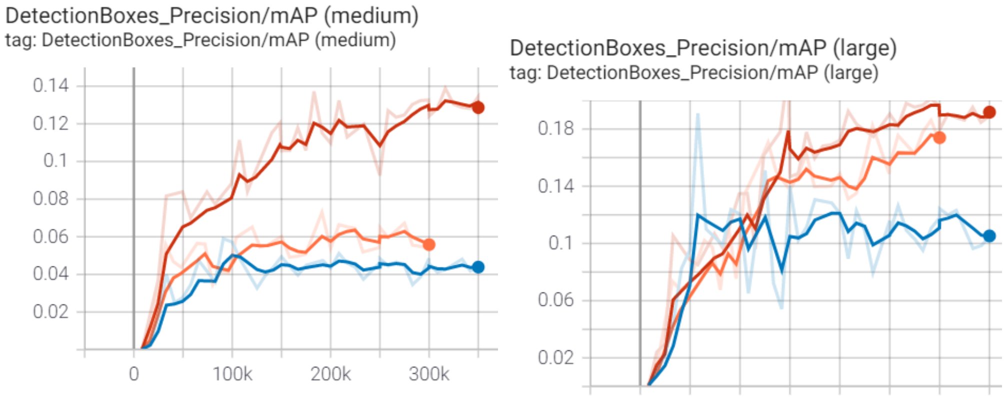
Figure D.1: MAP[0.5:0.95]



(a) MAP[0.5]

(b) MAP[0.75]

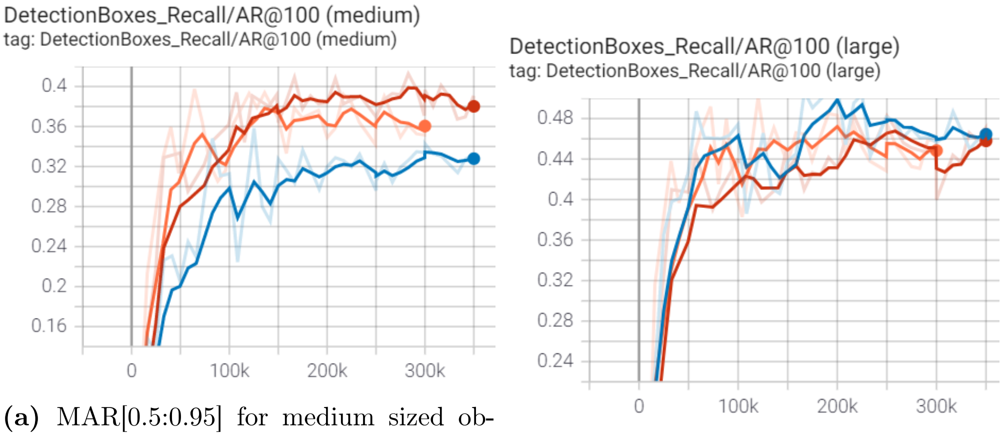
Figure D.2: MAP[0.5] and mAP[0.75] from training utilizing different classification systems.



(a) MAP[0.5:0.95] for medium sized objects

(b) MAP[0.5:0.95] for large sized objects

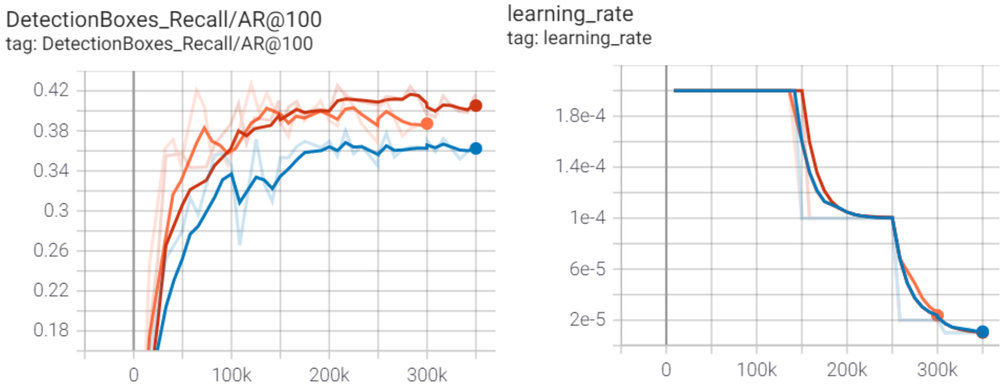
Figure D.3: MAP[0.5:0.95] for medium and large sized objects.



(a) MAR[0.5:0.95] for medium sized objects

(b) MAR[0.5:0.95] for large sized objects

Figure D.4: MAR[0.5:0.95] for medium and large sized objects.



(a) MAR[0.5:0.95] for all sized objects

(b) Learning rate while training

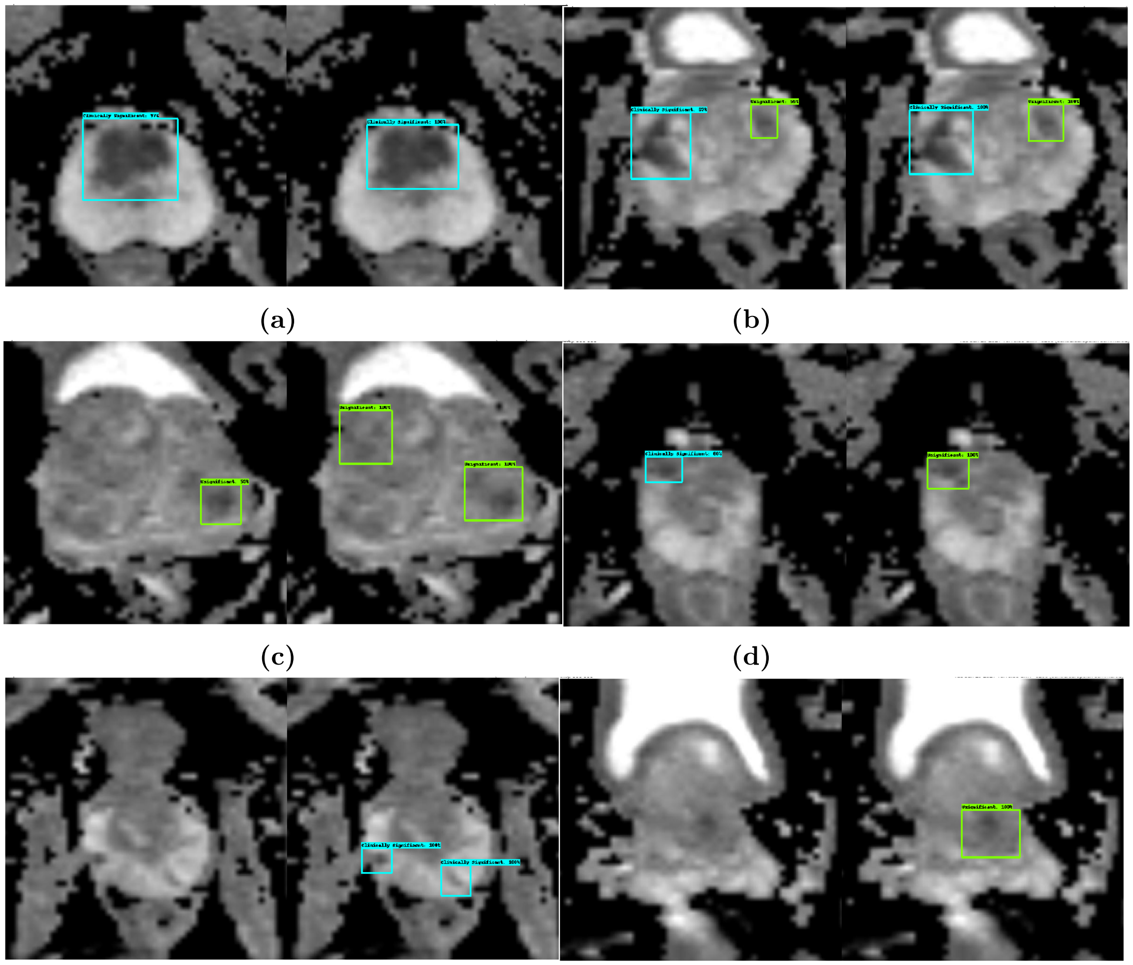
Figure D.5: Overall mAR[0.5:0.95] and learning rate while training.

Appendix E

Prediction Results - Classification

This appendix illustrates predictions carried out by the Faster R-CNN ResNet-50 model, form the classification experiment. Each sub-figures shows the prediction BB (P_B) to the left and ground truth BB (G_B) to the right.

Significant Classification


Clinically Significant: 97%
Clinically Significant: 100%
Clinically Significant: 87%
Insignificant: 98%
Clinically Significant: 100%
Insignificant: 100%
Insignificant: 58%
Insignificant: 100%
Insignificant: 100%
Clinically Significant: 80%
Insignificant: 100%
Clinically Significant: 100%
Clinically Significant: 100%
Insignificant: 100%

(a)

(b)

(c)

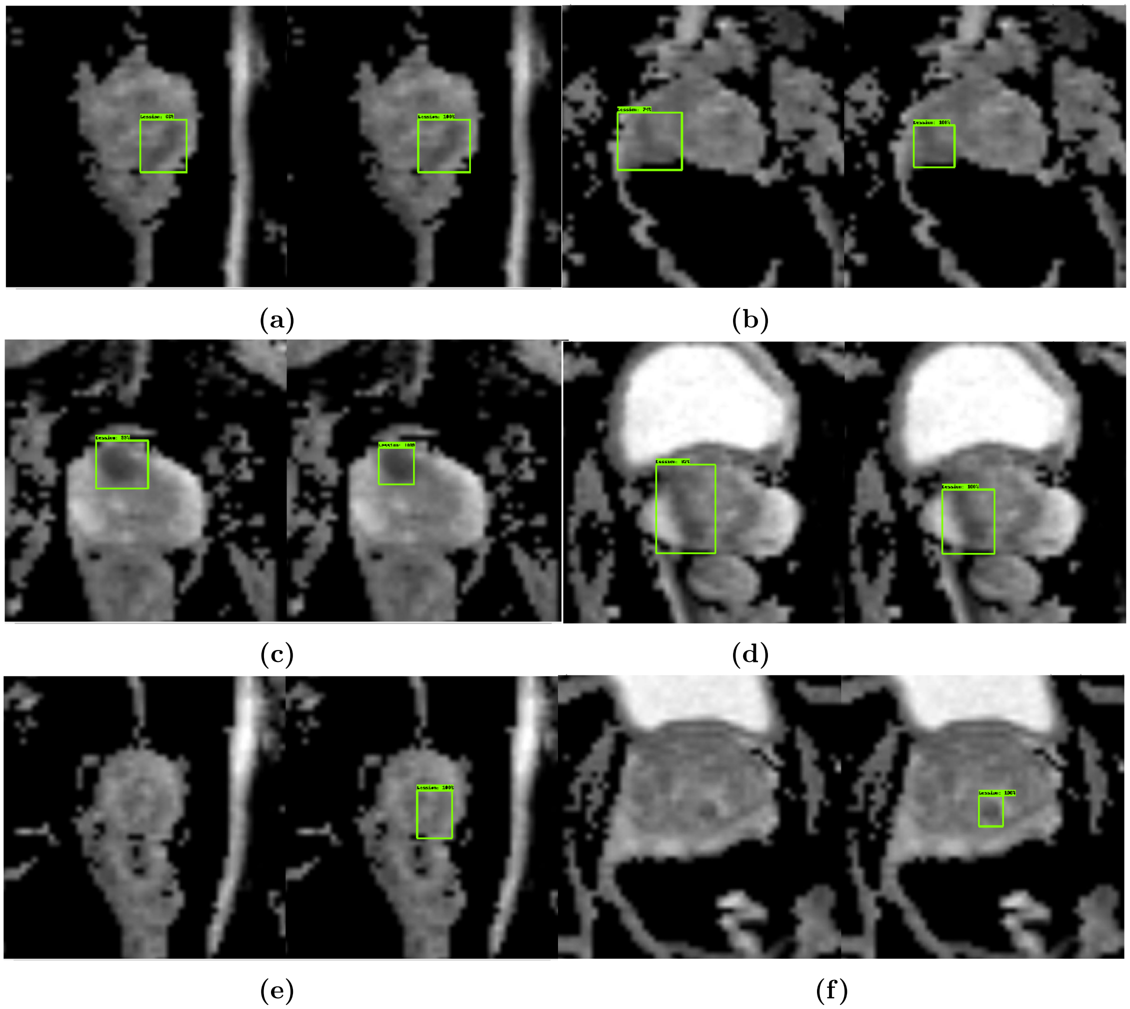
(d)

(e)

(f)

Figure E.1: Some predictions from the Significant classification experiment.

Lesion Classification



(a)

(b)

(c)

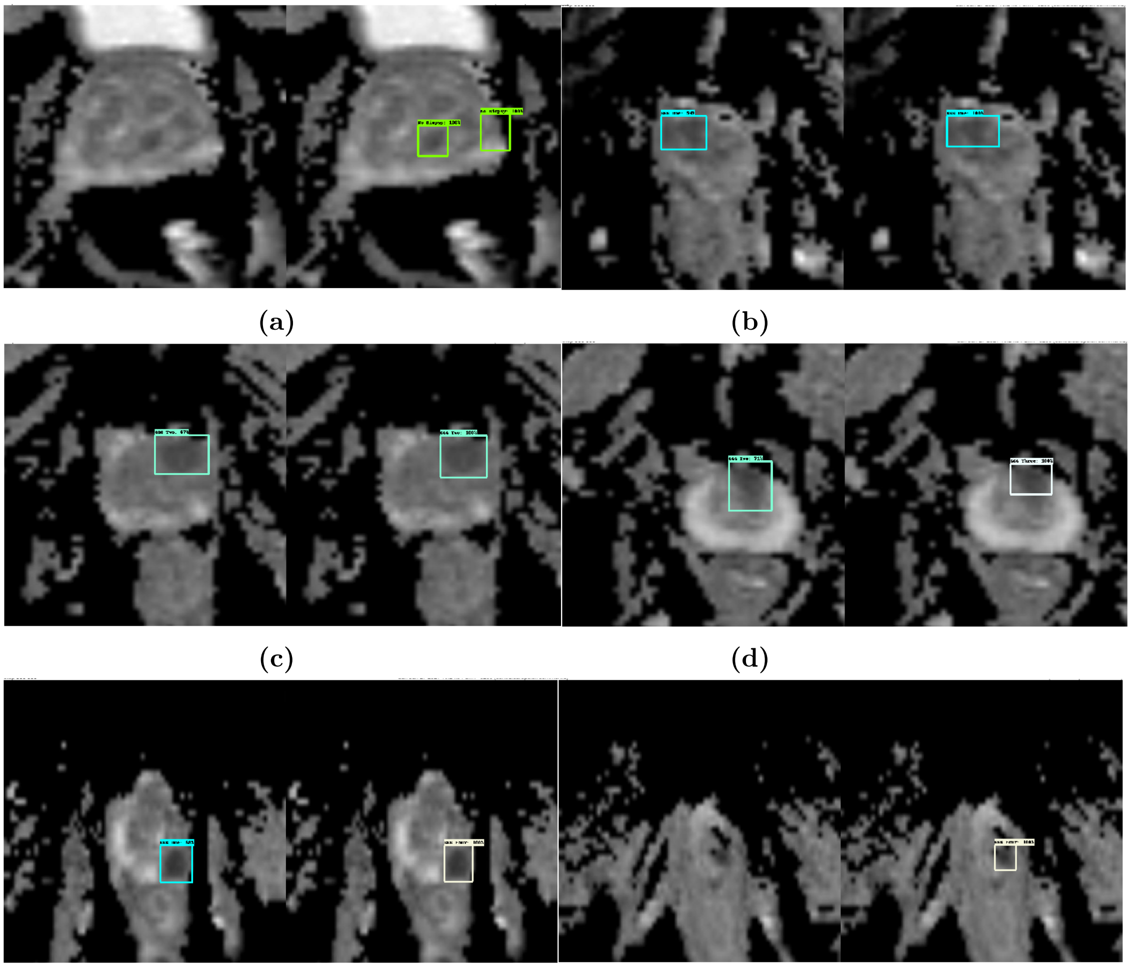
(d)

(e)

(f)

Figure E.2: Some predictions from the Lesion classification experiment.

GGG Classification



(a) (b)

(c) (d)

(e) (f)

Figure E.3: Some predictions from the GGG classification experiment.

Appendix F

Repertory

This appendix illustrates the structure of this thesis repertory, both for image pre-processing and for training deep CNN, utilizing the Tensorflow Object Detection API [35]. Only the main scripts and files relevant for this project are mention as the Tensorflow Object Detection API contains numerous examples. The image pre-processing code is available at GitHub [1].

Image Pre-Procession

```
scripts
    image_preprocessing.ipynb
MRI
    DICOM files
mask
    NIfTI files
    PROSTATEx_Classes.csv
    PROSTATEx_Classes_zones.csv
    Image_list.csv
```

[1] https://github.com/enliden1/Master_PCa_Detection

Repertory

Tensorflow Object Detection API

```
│ scripts
│  ├─ setup.py
│  ├─ generate_tfrecord.py
│  ├─ model_main.py
│  ├─ eval.py
│  └─ export_inference_graph.py
├─ data
│  ├─ images
│  │  ├─ train
│  │  ├─ validation
│  │  └─ test
│  └─ annotation
│     ├─ train.record
│     ├─ validation.record
│     ├─ test.record
│     └─ CustomObject2.pbtxt
└─ configs
   ├─ faster_rcnn_inception_v2_coco.config
   ├─ faster_rcnn_resnet50_coco.config
   ├─ faster_rcnn_resnet101_coco.config
   ├─ rfcn_resnet101_coco.config
   ├─ ssd_inception_v2_coco.config
   ├─ ssd_mobilenet_v2_coco.config
   └─ ssd_resnet50_v1_fpn_shared_box_predictor_640x640_coco14_sync.config
```