IPS Instituto Politécnico de Setúbal
Escola Superior de Tecnologia de Setúbal

João Nuno
Centeno Raimundo

REPRESENTATION LEARNING FOR BREAST CANCER LESION DETECTION

Trabalho de Projeto submetido como requisito parcial para obtenção do grau de **Mestre em Engenharia Informática**

Coordenador de Curso
Professor Doutor Cláudio Miguel Garcia Loureiro dos Santos Sapateiro
IPS-EST

Orientador
Professor Doutor Miguel Angel Guevara Lopez
IPS-EST

Arguente
Professor Doutor Augusto Marques Ferreira da Silva
UA- IEETA

Novembro de 2022

ACKNOWLEDGMENTS

First and foremost, I am extremely grateful to my supervisor Professor Miguel Angel Guevara Lopez for his invaluable dedication, patience, feedback, and continuous assistance.

I would like to express my deepest gratitude to João Pedro Fontes, Ph.D. student, who contributed with experience and guidance in the field of Computer Vision.

I would also like to thank Professor Claudio Miguel Garcia Loureiro dos Santos Sapateiro for his continuous support, teachings, and advice during my undergraduate and master's degrees.

I acknowledge Vista_Lab Research Center and BigData Laboratory, University of Evora, Portugal, for granting access to the GPU cluster *Vision* (DGX-A100 NVIDIA stations), without which it would not have been possible to present well-validated results promptly.

To Polytechnic Institute of Setubal, Portugal, and all faculty members and academic staff to whom I'm grateful for sharing knowledge and providing infrastructure.

To the Centro Hospitalar de Setúbal, especially to the Radiology and Surgery Departments who contributed with anonymized clinical cases and results analysis.

I would also like to warmly thank my family for always believing in me and for their continuous encouragement during this challenging period.

ABSTRACT

Breast Cancer (BC) is the second type of cancer with a higher incidence in women, it is responsible for the death of hundreds of thousands of women every year. However, when detected in the early stages of the disease, treatment methods have proven to be very effective in increasing life expectancy and, in many cases, patients fully recover. Several medical image modalities, such as MG – Mammography (X-Rays), US - Ultrasound, CT - Computer Tomography, MRI - Magnetic Resonance Imaging, and Tomosynthesis have been explored to support radiologists/physicians in clinical decision-making workflows for the detection and diagnosis of BC. MG is the imaging modality more used at the worldwide level, however, recent research results have demonstrated that breast MRI is more sensitive than mammography to find pathological lesions, and it is not limited/affected by breast density issues. Therefore, it is currently a trend to introduce MRI-based breast assessment into clinical workflows (screening and diagnosis), but when compared to MG the workload of radiologists/physicians increases, MRI assessment is a more time-consuming task, and its effectiveness is affected not only by the variety of morphological characteristics of each specific tumor phenotype and its origin but also by human fatigue. Computer-Aided Detection (CADe) methods have been widely explored primarily in mammography screening tasks, but it remains an unsolved problem in breast MRI settings.

This work aims to explore and validate BC detection models using Machine (Deep) Learning algorithms. As the main contribution, we have developed and validated an innovative method that improves the "breast MRI preprocessing phase" to select the patient's image slices and bounding boxes representing pathological lesions. With this, it is possible to build a more robust training dataset to feed the deep learning models, reducing the computation time and the dimension of the dataset, and more importantly, to identify with high accuracy the specific regions (bounding boxes) for each of the patient images, in which a possible pathological lesion (tumor) has been identified. In experimental settings using a fully annotated (released for public domain) dataset comprising a total of 922 MRI-based BC patient cases, we have achieved, as the most accurate trained model, an accuracy rate of 97.83%, and subsequently, applying a ten-fold cross-validation method, a mean accuracy on the trained models of 94.46% and an associated standard deviation of 2.43%.

Keywords: Breast Cancer Detection, Magnetic Resonance Imaging, Computer Vision, Machine Learning, Deep Learning, Convolutional Neural Networks.

RESUMO

O cancro da mama (CdM) é o segundo tipo de cancro com maior incidência nas mulheres. É responsável pela morte de centenas de milhares de mulheres todos os anos. Contudo, quando detetado nas fases iniciais da doença, os métodos de tratamento provaram ser muito eficazes aumentando a esperança de vida e, em muitos casos, os pacientes recuperam totalmente. Têm sido exploradas várias modalidades de imagem médica, tais como MG - Mamografia (Raios-X), US - Ultra-som, CT - Tomografia Computadorizada, MRI - Ressonância Magnética e Tomossíntese, para apoiar radiologistas nos fluxos de trabalho clínico para a deteção e diagnóstico do CdM. A MG é a modalidade de imagem mais utilizada a nível mundial, contudo, resultados de pesquisas recentes demonstraram que o MRI é mais sensível do que a mamografia para encontrar lesões patológicas e, também, não é limitada ou afetada por questões de densidade mamária. Consequentemente, atualmente é uma tendência introduzir a avaliação mamográfica baseada em MRI nos fluxos de trabalho clínico - rastreio e diagnóstico -, mas quando comparada com a MG, a carga de trabalho dos radiologistas aumenta. A avaliação por MRI é uma tarefa mais demorada, e a sua eficácia é afetada não só pela variedade de características morfológicas e origem de cada fenótipo tumoral específico, mas, também pela fadiga humana. Os métodos de deteção assistida por computador (CADe) têm sido amplamente explorados principalmente em tarefas de rastreio mamográfico, mas continua a ser um problema por resolver em ambientes de ressonância magnética mamária.

Este trabalho visa explorar e validar modelos de deteção de CdM usando algoritmos de Machine (Deep) Learning. Como contributo principal, desenvolvemos e validámos um método inovador que melhora a "fase de pré-processamento das imagens de ressonância magnética mamária" para selecionar as fatias de imagem do paciente e as respetivas caixas de contorno que representam as lesões patológicas. Com isto, é possível construir um conjunto de dados de treino mais robusto para alimentar os modelos de deep learning, reduzir o tempo de computação, reduzir a dimensão do conjunto de dados e, mais importante, para identificar com alta precisão as regiões específicas para cada uma das imagens do paciente nas quais foi identificada uma possível lesão patológica (tumor). Os resultados experimentais, num conjunto de imagens de ressonância magnética de domínio público totalmente anotado com 922 casos de doentes com CdM, mostram no melhor modelo uma taxa de exatidão de 97.83%. Foi aplicado um método de validação cruzada de 10 *folds* do qual resultou uma exatidão média de 94,46% com um desvio padrão de 2,43% nos modelos treinados.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

LIST OF CODE BLOCKS

ACRONYMS

AI	Artificial Intelligence
ANN	Artificial Neural Networks
AP	Average Precision
BNN	Biological Neural Networks
BC	Breast Cancer
BI-RADS	Breast Imaging-Reporting and Data System
BPE	Background Parenchymal Enhancement
CADe	Computer-Aided Detection
CADx	Computer-Aided Diagnosis
CNN	Convolutional Neural Networks
CV	Computer Vision
DL	Deep Learning
DICOM	Digital Imaging and COmmunications in Medicine
DM	Digital Mammography
DT	Decision Trees
DTS	Digital Tomosynthesis
FN	False Negative
FP	False Positive
GD	Gradient Descent
HP	Histopathology
IDC	Invasive Ductal Carcinoma
ILC	Invasive Lobular Carcinoma
IW	Imaging Workstations
IoU	Intersection over Union
kNN	k-Nearest Neighbours
mAP	mean Average Precision
MG	Mammogram
ML	Machine Learning
MDL	Machine (Deep) Learning
MRI	Magnetic Resonance Imaging

NB	Naïve Bayes
NN	Neural Networks
OS	Operating System
PACS	Picture Archiving and Communication System
R-CNN	Region-based Convolutional Neural Network
ReLU	Rectified Linear Function
ResNet	Residual Network
SF	Sigmoid Function
SVM	Support Vector Machine
TA	Textural Analysis
TN	True Negative
TP	True Positive
UI	User Interface
V-ANN	Vanilla Artificial Neural Networks
US	Ultrasound

GLOSSARY

Accuracy	The degree of closeness of measurements of a quantity to that quantity's true value. *Accuracy = Correct Classification / All classification.* (Wikipedia, 2022a)
Cancer	A disease in which some of the body's cells grow uncontrollably and spread to other parts of the body (National Cancer Institute, 2021).
Histopathology	Microscopic examination of tissue to study the manifestations of the disease. Specifically, in clinical medicine, histopathology refers to the examination of a biopsy or surgical specimen by a pathologist, after the specimen has been processed and histological sections have been placed onto glass slides (Wikipedia, 2022d).
Magnetic Resonance Imaging	A medical imaging technique used in radiology to form pictures of the anatomy and the physiological processes of the body. MRI scanners use strong magnetic fields, magnetic field gradients, and radio waves to generate images of the organs in the body. MRI does not involve X-rays or the use of ionizing radiation, which distinguishes it from CT and PET scans (Wikipedia, 2022g).
Mammography / Mastography	Is the process of using low-energy X-rays (usually around 30 kVp) to examine the human breast for diagnosis and screening. The goal of mammography is the early detection of breast cancer, typically through the detection of characteristic masses or microcalcifications (Wikipedia, 2022e).
Mitosis	Refers to the cellular process where a single cell division results in two identical cells. Identical means that both cells have the same number of chromosomes and genetic content (BiologyOnline, 2021).
Neoplasm	A type of abnormal and excessive growth of tissue. The abnormal growth usually forms a mass, when it may be called a tumor (Wikipedia, 2022f).
Patient-wise	Operations (arithmetical) on an array/string of patients. When the unit is the patient.
Precision	The fraction of relevant instances among the retrieved instances. *TP=TP/FP* (Wikipedia, 2022b)
Recall	The fraction of relevant instances that were retrieved. *TP=TP/FN* (Wikipedia, 2022b)
Tomography	Imaging by sections or sectioning using any kind of penetrating wave. The method is used in radiology, archaeology, biology, atmospheric science, geophysics, oceanography, plasma physics, materials science, astrophysics, quantum information, and other areas of science (Wikipedia, 2022h).
Tomosynthesis	Tomosynthesis, also digital tomosynthesis (DTS), is a method for performing high-resolution limited-angle tomography at radiation dose levels comparable with projectional radiography. It has been studied for a variety of clinical applications, including vascular imaging, dental imaging, orthopedic imaging, mammographic imaging, musculoskeletal imaging, and chest imaging (Wikipedia, 2021).

Tumor-wise	Operations (arithmetical) on an array/string of tumors. When the unit is the tumor.
Type I Error	In statistical hypothesis testing, a type I error is the mistaken rejection of a true null hypothesis (also known as a "false positive") (Wikipedia, 2022c).
Type II Error	In statistical hypothesis testing, a type II error is the failure to reject a false null hypothesis (also known as a "false negative") (Wikipedia, 2022c).

1. INTRODUCTION

In 2020, approximately a total of 2.26 million women were diagnosed with Breast Cancer (BC) and 685 thousand women diagnosed with breast cancer died (Sung et al., 2021). Also in 2020 breast cancer was the most prevalent cancer and the fifth most common cause of cancer death worldwide. However, early detection and accurate diagnosis are significant to improve the prognosis and increase the survival rate of patients with BC by 30% to 50% (World Health Organization, 2022). The treatment of BC is highly effective when it is detected in the early stages of the disease (World Health Organization, 2021). Therefore, the early detection of BC is a critical issue that represents an urgent global priority.

In the coming years, countries with weak health systems and lower incomes will suffer more severe consequences in terms of both diagnosis and mortality related to breast cancer (Barrios, 2022). The Covid 19 pandemic severely hampered the process of cancer diagnosis and treatment at a worldwide level, e.g., developed countries such as Canada, the Netherlands, Germany, Italy, the United Kingdom, and Australia even suspended their national breast cancer screening programs for periods of between one and six months (Figueroa et al., 2021).

The heterogeneity of breast cancer results from a diversity of factors, in general, dominated by the morphological characteristics of tumors and the origin of the neoplasms (Viale, 2012). The complexity of automated detection and/or classification of breast tumors arises from their variety of types and subtypes. It is currently an unsolved problem with added difficulties for radiologists in terms of Magnetic Resonance Imaging (MRI) image analysis.

BC Computer-Aided Detection (CADe) systems aim to support medical decisions and prescriptions. This work aims to explore and validate BC detection models using Machine (Deep) Learning algorithms for developing a more robust and precise solution capable of assisting radiologists in this process.

Currently, the problem of BC detection has been addressed by applying Artificial Intelligence (AI) techniques, namely Machine (Deep) Learning (MDL) and Computer Vision (CV) algorithms and methods. In this work, we are focused on the development of robust and precise MDL detection models. For this purpose, we have explored already released public domain datasets (MRI-based) and associated metadata (e.g., patients' clinical history, Breast Imaging and Data System (BIRADS), verified biopsies, etc.).

In addition, a collaborative scientific research project is being carried out between IPS and the Public Hospital of the district of Setubal, Portugal. This project will allow the building of a new benchmarking BC digital repository and validate the proposed detection method on national (Portuguese) patients' cases.

The departments of Radiology and Imaging in public and private hospitals are using machines (computers) as medical Imaging Workstations (IW). IWs are interfaces/terminals that are used to ease the work of radiologists to examine medical images. These IWs can communicate with Picture Archiving Communication Systems (PACS), which are servers to store and manage patients' data (images and other

relevant clinical data). This fact is a clear advantage for this project, due to it can be straightforward to deploy/integrate any developed prototype or validated solution as an Application Programming Interface (API) on IWs.

1.1. Document Structure

This document is structured in 9 chapters.

CHAPTER 1, this introduction, to the scope of the problem and the main directions and motivation for this project.

CHAPTER 2, overall planification of this work, setting the objective and schedule for this project as well as the definition of the methodology.

CHAPTER 3, research on the current state of the art in medical imaging, as well as methodologies and applications of machine (deep) learning in the field of breast cancer detection and an overview of convolutional neural networks. Also, it contains a brief description of the popular metrics used for evaluating deep learning models.

CHAPTER 4, presentation, and analysis concerning materials and methods (i.e., details of the dataset selected to explore detection algorithms and methods, as well as, a description of the proposed method, highlighting the contributions to the data preprocessing phase. Description of the chosen deep learning architecture and evaluation methodology.

This section also encompasses the description of the environment, development, and implementation of the proposed method, details, and options of the solutions proposed.

CHAPTER 5, setup, and presentation of the achieved results as well as communication actions, outlining the current state of the work in events related to the domain area of this work.

CHAPTER 6, Conclusions of this thesis and future work proposals considering how the work could be expanded and how the method can be improved.

2. PLANNING

2.1. Scope

- This project arises bound to my software engineering master's degree final project and it was conceived by Professor Miguel Angel Guevara Lopez, my project mentor, and sponsor.
- This project aims to help radiologists (physicians) accelerate the process of detecting pathological breast cancer lesions in MRI-based workflows. The project builds on my supervisor's previous work concerning the development of machine (deep) learning methods for supporting the detection and diagnosis of pathological lesions in breast cancer patients and, it answers to one of the defined tasks inside an ongoing scientific collaboration protocol between the Setúbal Polytechnic Institute and the Centro Hospitalar de Setúbal.
- The main goal of this project is the creation of an AI method capable to improve the early detection of breast cancer pathological lesions on MRI. This is relevant as the process of breast cancer detection in MRI workflows is a high time-consuming process, affected by physicians' fatigue, and requires highly specialized skills.
- Since the early detection and diagnosis of breast cancer is a critical issue, the detection accuracy of the models should be at least 95%, thus ensuring a very small number of False Negatives (FN) (Type II errors). The alternatives are human analysis of hundreds of MRI images per patient for visual detection of breast tumors.

2.2. S.M.A.R.T

The goal of this project is to develop a **computer vision solution for the detection of breast cancer lesions on MRI** with an **accuracy of over 95%** within **eight months**.

2.3. Objective

This project aims to the development of a breast cancer detection model using a machine (deep) learning approach. As mentioned, there are five main medical imaging modalities used for detection, the focus will be on MRI. The selection of this over the remaining four modalities as it is still an unsolved problem and one of the more human-demanding modalities which could benefit from automatization. MRI analysis is an expensive and time-consuming procedure that requires highly specialized skills to perform and is conditioned by human factors such as fatigue.

Also, in terms of communication, this mission intends to produce a scientific contribution in form of a presentation at a scientific conference and a publication in a scientific journal.

2.4. Methodology

This project is highly dependent on external (public) datasets, which will be closely connected to the detection model's success and therefore will require specific metrics and testing procedures to evaluate the accuracy and rethink model parametrization. For this reason, and to allow some responsiveness and strategy reorganization, the project management will be approached from an Agile perspective.

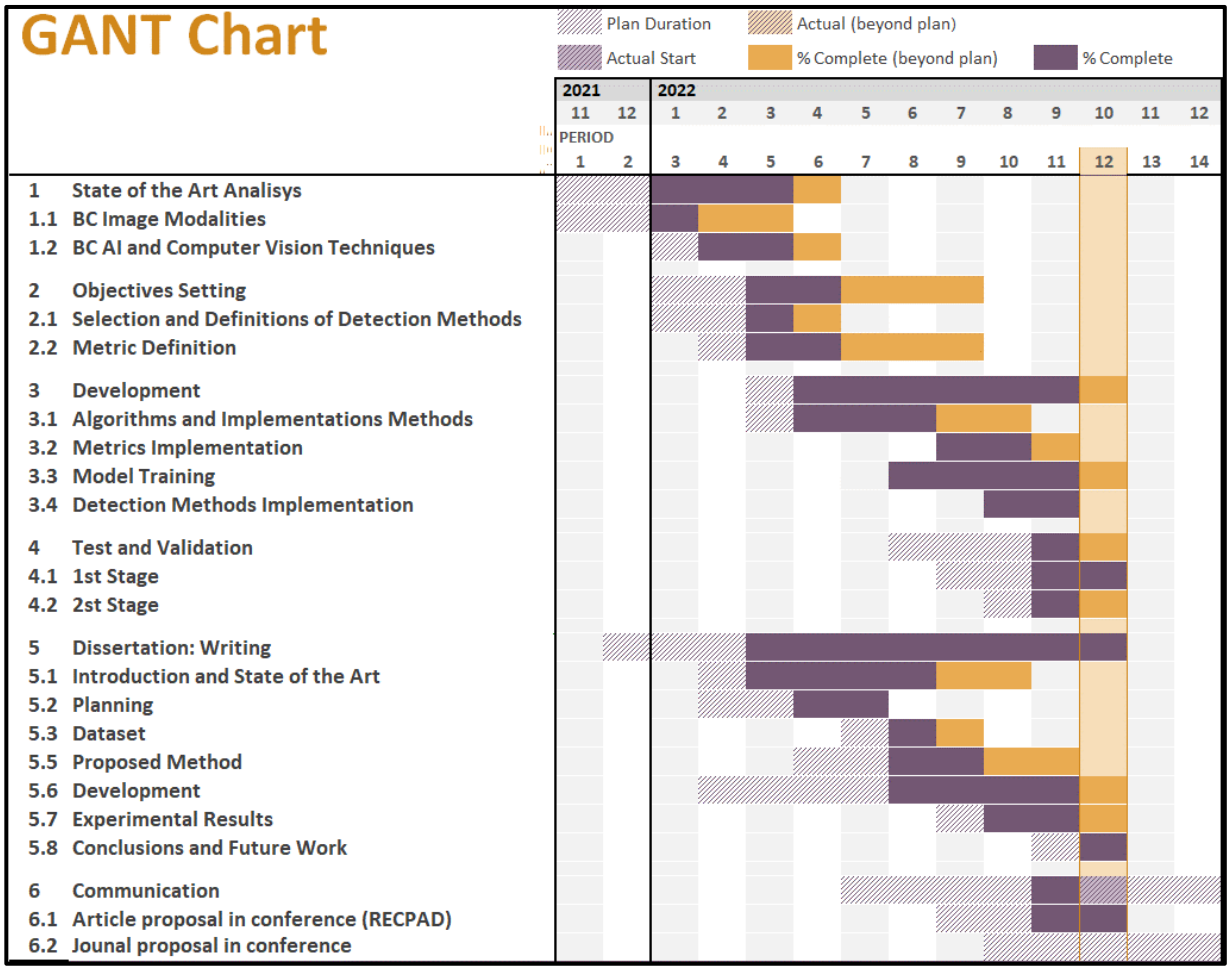
Agile software development assumes that requirements – either functional, quality, or environmental – can change during the project. One of the major advantages of Agile methods is the possibility of splitting usual preliminary software design – as, for instance, good practice in waterfall methodologies - into smaller chunks of functionalities resulting in more flexibility.

The agile methodology follows the Manifesto for Agile Software Development (Beck et al., 2001) as summarized in the following four key principles:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

2.5. Gant Chart

Table 1: Gant Chart



3. STATE OF THE ART

3.1. Medical Imaging Modalities

In general, most of the work developed to date on Computer-Aided Detection / Diagnosis (CADe/CADx) methods/systems in BC pathological lesions (tumors), particularly on classification systems, is based on Histopathology (HP) biopsy images and or 2 Dimensions (2D) X-Ray Mammography (MG). However, other modalities including Ultrasound (US), Digital Tomosynthesis, and Magnetic Resonance Images (MRI) have been explored (Murtaza et al., 2020).

Of all BC tumors, 70% to 80% are related to one of the major histopathological types, Invasive Ductal Carcinoma (IDC) and Invasive Lobular Carcinoma (ILC). These two main classes group most of the diverse spectrum of types (Viale, 2012) (Motlagh et al., 2018). The remaining 20% to 30% of BC tumors not typed as IDC or ILC fit into other categories weakly represented in datasets, a characteristic that traditionally is an impairing factor for DL solutions.

Histopathology biopsy is an invasive and conclusive medical imaging modality. It is considered the which offers rich phenotypic details (Khaliliboroujeni et al., 2022). Although their results in terms of ML algorithms for multi-class classification are not optimal (McCann et al., 2015), it is still widely used for Deep Learning (DL) classification solutions (Gupta et al., 2021).

Mammography is one of the most used modalities for early screening. This modality is not only useful to help determine the tumor mass but also its location (Gupta et al., 2021). Nevertheless, alongside 2D MG additional screening with supporting modalities, such as the US, is required as an MG has low sensitivity, particularly in presence of dense breast tissue images (Aristokli et al., 2022).

The advantage of the US, when compared to other techniques is the fact of not having radiation involved. However, in addition to these advantages, there are other limiting factors. For instance, the US is limited in its ability to distinguish between calcifications and cancerous masses. Their poor image quality is prone to cases of misinterpretation; in part, the US is also used as the ancillary methodology for the screening of BC and to assist in the decision of prescribing further analysis with other modalities such as biopsy (Youk et al., 2017).

Digital tomosynthesis is one of the less studied modalities as it was recently introduced and there less available datasets (Buda et al., 2020). Compared to 2D MG, DTS enables a more effective BC diagnostic capacity and produces more trustful interpretations (Abdel-Nasser et al., 2020). Its 3D views prove to be more powerful for the detection of abnormalities. However, as it is more challenging to handle 3D data for

automated detection, 2D MG classification results were demonstrated to be more efficient (Tariq et al., 2021).

MRI is the most accurate radiological method for accessing tumor size, multifocality, and multicentricity (Durhan et al., 2021), it has better sensitivity and higher diagnostic accuracy (Wu et al., 2022) but it's not used for screening on its own, is considered too expensive and time-consuming (Rezaei, 2021) as for the analysis and assessment of MRI sequences which can easily group several hundreds of images for a single patient.

Until a few years ago, MRI had only been used as an ancillary modality for non-conclusive US and MG increasing their detection rates and it's predominantly used for evaluating other features such as size and for detecting other tumor areas (Aristokli et al., 2022). As aforementioned, MRI is particularly expensive as an activity that requires endeavor, it is highly dependent on human efforts, nevertheless, the same author states that 7 out 8 of his studies reveal that the mean sensitivity of MRI is 95.6%. Therefore, it is crucial to reduce human effort in MRI evaluation and take advantage of its high sensitivity performance/capacity to detect pathological lesions.

The studies reviewed address both detection and classification - as one of the two main types - the binary early detection (including classification), between benign and malignant tumors, or the multiclass classification, which aims not only to distinguish benign and malignant but also different tumor types and subtypes/phenotypes.

Figure 1: Image modalities samples
A: Breast histopathology sample image (Munir Shah et al., 2021).
B: Mammography sample image (Abdelrahman et al., 2021).
C: Breast ultrasound sample (Byra et al., 2022).
D: Breast digital tomosynthesis sample image (Zhang et al., 2020).
E: Breast MRI sample image (Saha et al., 2021).

3.2. Machine (Deep) Learning: Methods

Detection is essentially a discovery process. A process capable of analyzing either the presence or absence of something or at the more generic level of computational language, of objects. Detection is an important process of Computer Vision, and a preliminary process for other processes such as segmentation and classification (Zou et al., 2019).

The Textural Analysis (TA) of MRI is already identified as having the potential to assist in classifying tumors as benign or malign (Brown et al., 2021) (Bębas et al., 2021). There are also studies on DL

algorithms for other modalities such as MRI Background Parenchymal Enhancement (BPE) classification. It is not BC classification but may affect diagnostic accuracy and therefore becomes relevant for this study.

Breast tumors are divided into several categories, 20 major types, and 18 minor subtypes. Apart from the binary classification, over 70% of breast cancers belong to one of two types of BC, Invasive Ductal Carcinoma (IDC) and Invasive Locular Carcinoma (ILC) (Jannesari et al., 2019). This uneven distribution of tumors results in added difficulty in automatically detecting breast cancer lesions using machine (deep) learning techniques. Breast tumor type distribution alone embodies some concerns in terms of dataset representativeness. As an example, in terms of training ML models the standard is to split between training and testing subsets – often randomly – with percentages between 70-30 to 85-15 depending on several aspects. This approach enables testing the model in fresh data that was previously isolated from the training data subset. In terms of DL model training, the dataset is normally split into 90% for the training set and 10% for the testing set.

However, as DL algorithms are used when there are available massive amounts of data, this ratio is used when we have Bigdata. In terms of cancer tumors, with less than 30% representing 18 major types of BC, it means that – assuming a dataset with all tumor types - the training set eventually may not include some of BC types, and, probably, the testing set would not include some of the tumor types.

3.3. Review of Artificial and Convolutional Neural Networks

DL is a subset of ML concerning the set of algorithms that allow computers “to learn from large collections of data” without the need for domain-specific (handcrafted) set of rules. In recent years ANNs have become one of the emergent (trendiest) concepts in terms of AI and DL algorithms.

Warren McCulloch and Walter Pitts began to study this sort of algorithm in the first half of the XX century, proposing in 1943 the first model of an artificial neural network (McCulloch & Pitts, 1943), called perceptron, with the current and increasing computational muscle, the research and improvement of ANNs are becoming ambitious in proportion. Nevertheless, ANNs are still very computationally expensive when used for image pattern recognition for detection and/or classification purposes.

ANNs approach images as sets of pixels, where each pixel is considered a feature, that is, each input feature matches one input node, as well as one output node, matches one class (Isaac Abiodun et al., 2018). Feature extraction aims to transform the input features into a new, smaller, feature set whose informational relevance is as identical as possible to the original input features thus reducing the number of features and therefore the number of input nodes (Ghojogh et al., 2019). This is especially important in terms of image classification as before feature extraction an image has as many features as pixels.

A simple image (Figure 2) as the black and white handwritten smile face in 20x20 pixels has an enormous feature map with a vast number of elements, more precisely 400. In terms of colored pictures, the standard

is to separate three distinct feature maps for Red, Green, and Blue (RGB) channels (Gonzalez, 2018; Jost et al., 2019), one for each color.

The colored picture in Figure 3, has 20x20 pixels, which is equivalent to 1200 features. In Vanilla (default) ANN (V-ANN) each feature is inputted to a separate node, and, as each pixel is a feature, V-ANN would need 1200 input nodes for a simple 20x20 pixels colored image. In Convolutional Neural Networks (CNN) there are a set of operations which may reduce de number of features.

Convolutions are a set of operations executed in a dedicated layer before inputting features to the NN nodes. The convolution layer itself serves as a feature extractor (Rawat & Wang, 2017). In the convolutional layer, each neuron is connected to its neighbor neurons forming feature maps capable of retaining the representation of the image dimensionality (Rawat & Wang, 2017).

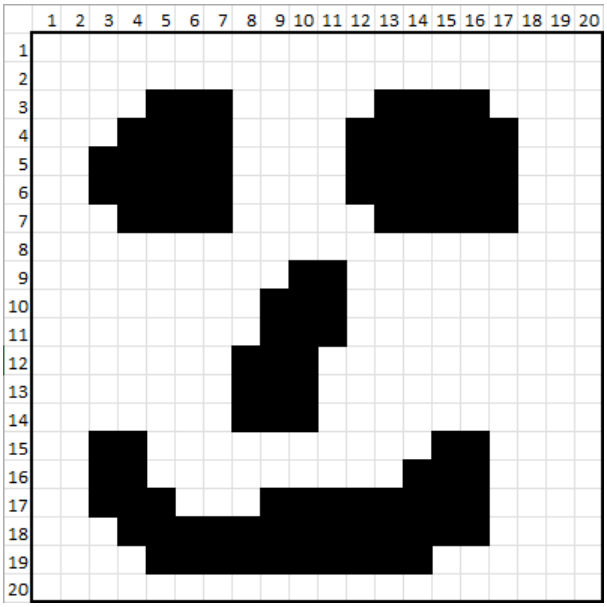


Figure 2: Black and White hand-drawn image of a smile with 20x20 pixels.

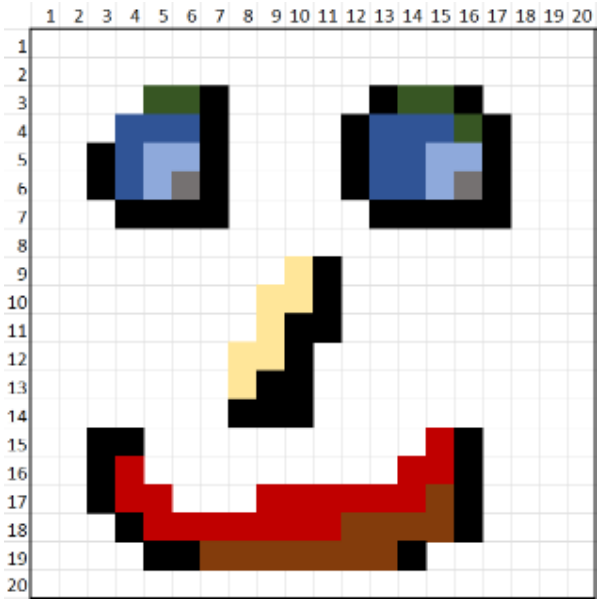


Figure 3: Colored hand-drawn image of a smile with 20x20 pixels.

3.3.1. Artificial Neural Networks

In DL, neurons, or perceptrons – the first artificial version of a biological neuron –, are processing nodes with connections between each other. These neurons are organized into sequential interconnected layers, as described in section 3.3.1.2 “Layers”. Single neurons send and receive signals from the previous and next layers. This is done by processing some input data, along with some weights and biases, and outputting its result value to another node (Han et al., 2018). Each neuron calculates the weighted sum of the input values and connections weights as described in section 3.3.1.2.3 “Hidden Layer”. The weighted sum suffers a transformation process through an Activation Function (AF) so that they become comparable and evaluated to either 0 or 1.

3.3.1.1. Activation Functions

Activation functions, also called Transfer Functions (TF), are used for transforming output values into 0 and 1, transforming values into numbers between 0 and 1, or into values greater than 0. There are plenty of alternatives, depending on the function used. Among the several TF, the major references are the Sigmoid Function (SF) and the Rectified Linear Units (ReLU) (Konde & Thakur, 2021).

The SF is a nonlinear function that turns any value into a value between 0 and 1, illustrated in Figure 4. The ReLU function (Figure 5) is used to transform negative values into zero, that is, select the maximum

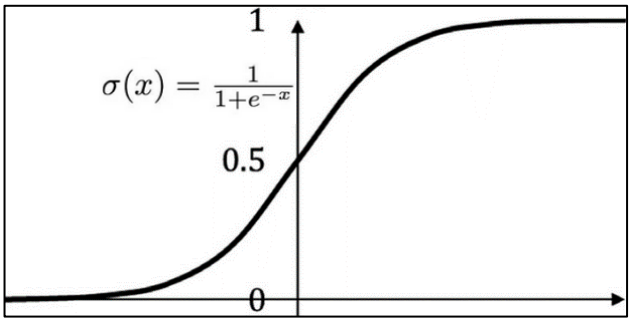


Figure 5: Sigmoid Function

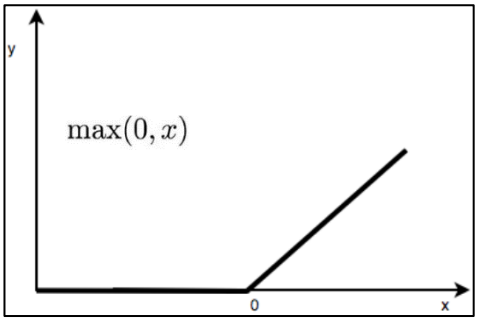


Figure 4: ReLU Function

value between the node output and zero. In part, its popularity is due to its processing cost it is much less expensive (computationally) as the only calculation is to figure out if a value is negative or not. When the input is negative outputs zero otherwise outputs the input value. Also, apart from the computation cost, the underlying meaning of 0 (zero) as an activation function output that will be passed to the neurons on the next layer is the nullifying value of the corresponding parcel of the weighted sum equation (section 3.3.1.2.3 “Hidden Layer”) meaning that it is the value of a deactivated neuron and so not contributing to the prediction result.

The neuron output is a normalized value, and the application of the activation function is done inside the neuron (Konde & Thakur, 2021) as depicted in Figure 6.

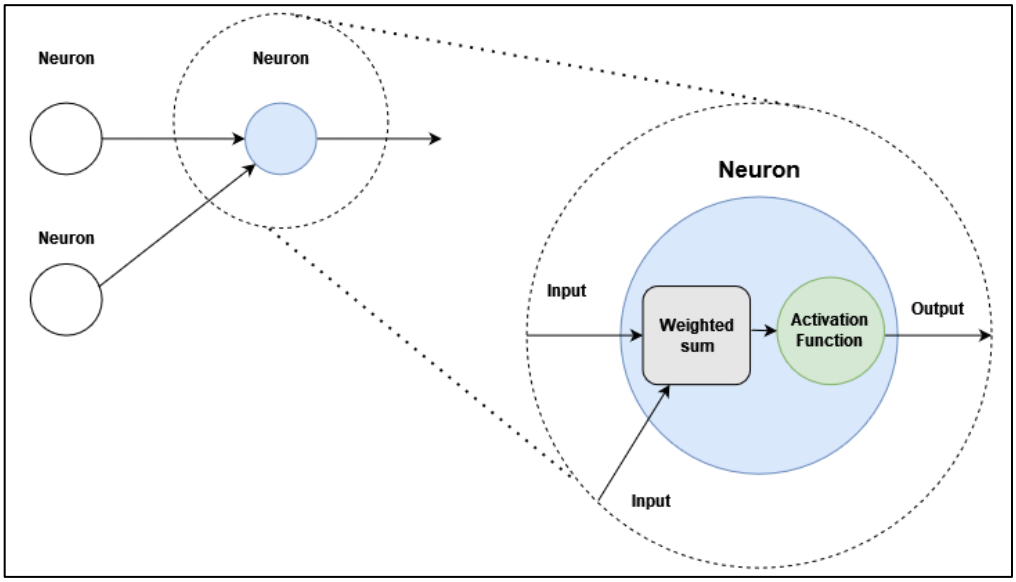


Figure 6::Application of the Activation Function inside the Neuron

3.3.1.2. Layers

Artificial Neural Networks (ANN) is a way to represent groups of neurons. ANNs group neurons on three different types of interconnected layers (Figure 7) that have distinct roles. The input layer, the hidden layers, and the output layer. The input layer – this first layer on an ANN - is responsible for inputting the features into the network and is connected to the first hidden layer. The output layer – the last layer on an ANN - connected to the last hidden layer has as many nodes as classes and receives as input the output of the last hidden layer neurons. The hidden layer, connecting the input layer and the output layer, is the layer responsible for performing calculations and passing normalized values between nodes (neurons).

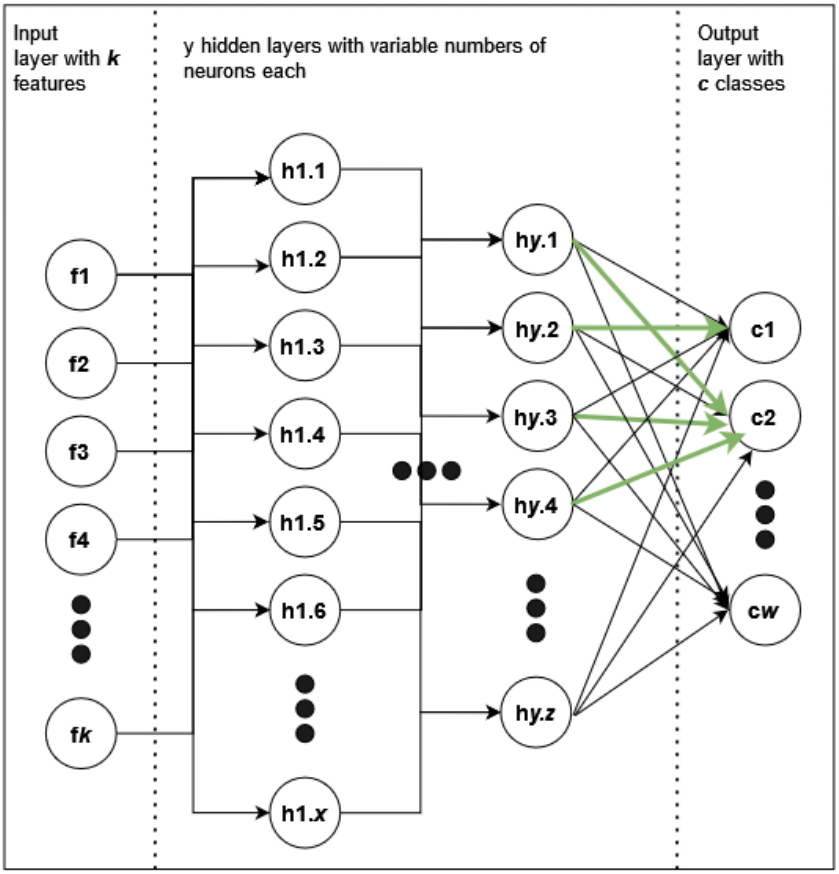


Figure 7: Artificial Neural Network Architecture

3.3.1.2.1. Input Layer

The input layer has as many neurons as input features. This is one of the disadvantages regarding image classification in V-ANN where each image pixel corresponds to a feature in a one-dimensional array. It

loses the notion of the bi-dimensionality of images interpreting them as one-dimension objects, hence not only losing information but also decreasing performance. This characteristic of V-ANN is the brightest side of Convolutional Neural Networks. Another important step contributing to the feature enhancement, although independent from ANN themselves, is the pre-processing of images, which can cover a considerable type of transformations, such as contrast enhancement, noise, and dimensionality reduction (Arya & Shankar Modani, 2019).

3.3.1.2.2. Output Layer and Backpropagation

ANNs can also be pictured as an acyclic-directed graph of artificial neurons (Aloysius & Geetha, 2018). The previous statement is half true as the connections can work in both directions, one at a time. If the connections were to work in both directions arbitrarily the graph would then be cyclic which in turn would negatively influence the pipeline and perhaps trapped on an endless loop. So, information travels from input nodes to output nodes. And afterward from output nodes to input nodes in a process also known as backpropagation which aims to adjust the connection weights.

The output has as many nodes as there are classes in the problem domain to be approached. Finding the error (or loss) between the predicted class and the actual class (Han et al., 2018) is crucial and a precursor of the learning process, and allows to the network building its hypothesis, which is not possible in traditional machine learning classifiers (Dong et al., 2021). This is possible by updating the neural network connection weights. It's an iterative process that aims to minimize the loss and, ideally, reach its end when the loss is minimal. This is achievable by propagating the calculated loss backward and adjusting the connection weights and biases during the training phase.

The perceptrons read input and output values using the back propagated loss value to alter the weights, aiming to decrease the variation between the predicted and actual class (Isaac Abiodun et al., 2018). The loss calculation and correction value weights regulation are not static procedures, distinct methods may well be used. One of the most used combinations is to use the *logarithmic* loss (or *cross-entropy*) (Xie et al., 2021) for error calculation and the gradient descent method to find out the weights and biases regulation values.

3.3.1.2.3. Hidden Layer

Hidden layers are responsible for the core calculations of neural networks. An ANN can have one or more hidden layers. The input layer outputs into the first hidden layer nodes (neurons). The output layer inputs the last hidden layer nodes' output values. When an ANN has more than one hidden layer the same logic applies, i.e., a layer L is fed by its previous L-1 layer nodes and feeds the L+1 layer nodes. Each hidden layer node outputs the normalized weighted sum of its input nodes. Each node on the network has a value and its connection a weight associated.

The weighted sum of the input nodes is the sum of each input node multiplied by its connection weight plus a bias value. Its result value is in turn normalized using the activation function mentioned in the Activation Functions section. After transformation by the activation function, the value is passed to the next layer nodes next to node N (Figure 7).

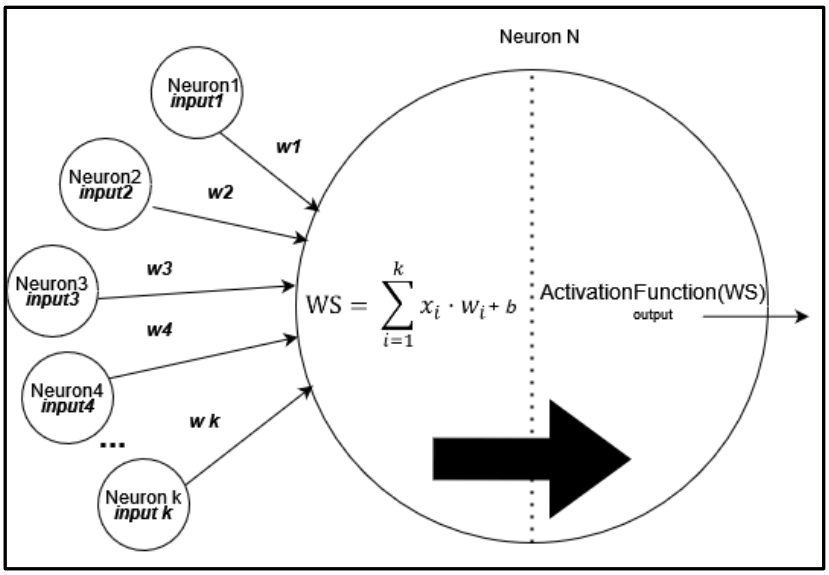


Figure 8: Weighted sum illustration. Input from the previous layer and output to the next layer.

Large output values strengthen the emitting node signal as low values diminish the signals. As for the AF effect, if its output ranges from 0 to 1, a 0 valued weight has the nullifying effect of the corresponding input node (Han et al., 2018), in such case, the nodes with zero value weight connection are a deactivated node. Further on, the node N output of Layer L is in turn the input of the nodes in layers L+1 (Equation 1) depending on the ANN architecture. A connection is a relation between an input and an output.

3.3.1.2.4. Gradient Descent

The gradient descent (GD) is a first-order iterative optimization algorithm for finding a local minimum of a differentiable function. The idea is to take repeated steps in the opposite direction of the gradient (or approximate gradient) of the function at the current point because this is the direction of steepest descent. GD method is an optimization algorithm to guide the learning process (Yamashita et al., 2018) reducing the value of the loss function which is calculated at the end of each epoch (one epoch is running all training set samples one time). Computing GD is a crucial step to subtract or add (feeding a negative *v*) (Equation 1) some value to each weight and bias when backpropagating the error.

This method determines the step size (*v*) for the weights and biases adjustments. The value *v* is influenced by the Learning Rate (LR). LR is a hyper-parameter that controls the weights of ANNs defining

Equation 1: Weight and biases update functions

w : each weight

b : each bias

v : value to be subtracted or added

L : Loss function

W : weights vector

LR: Learning Rate

$$v = calcV(L, W, LR) = LR * \frac{\partial L}{\partial w}$$

$$adjustWeigths(w, v) = w - v$$

$$adjustBiases(b, v) = b - v$$

how quickly an ANN updates the concepts it has learned. LR values ranges between 0 and 1, generally close to use to limit the step size so that it does not become too abrupt. The *v* value is the quotient of the loss function differential and the weight vector differential inhibited by the LR.

LR will also influence the overall training process as a greater LR will produce a greater *v* value which will result in reaching the minimal loss more swiftly. Once reached the minimal loss value, the learning process will end.

On the other hand, a smaller LR will result in reaching the minimal loss more gently, dragging the learning process throughout more time but resulting in a less optimal result as the stop nearest to the lowest loss may have a wider difference resulting in a less suitable process (Cui, 2018) as represented in Figure 9.

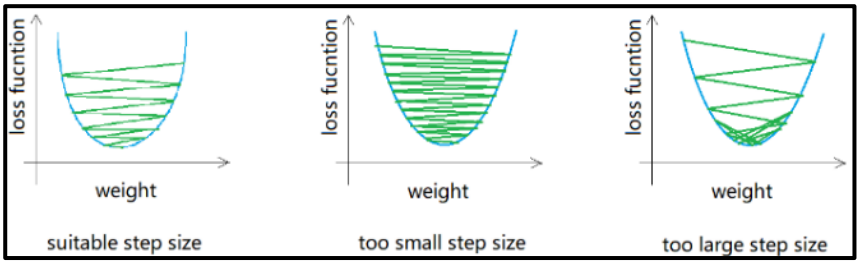


Figure 9: Gradient decent. Illustration for different learning rates (Cui, 2018)

3.3.2. Convolutional Neural Networks

Figure 10 represents the global architecture of CNN with one convolutional layer, one ReLU layer, one pooling layer, and one fully connected layer (FCL).

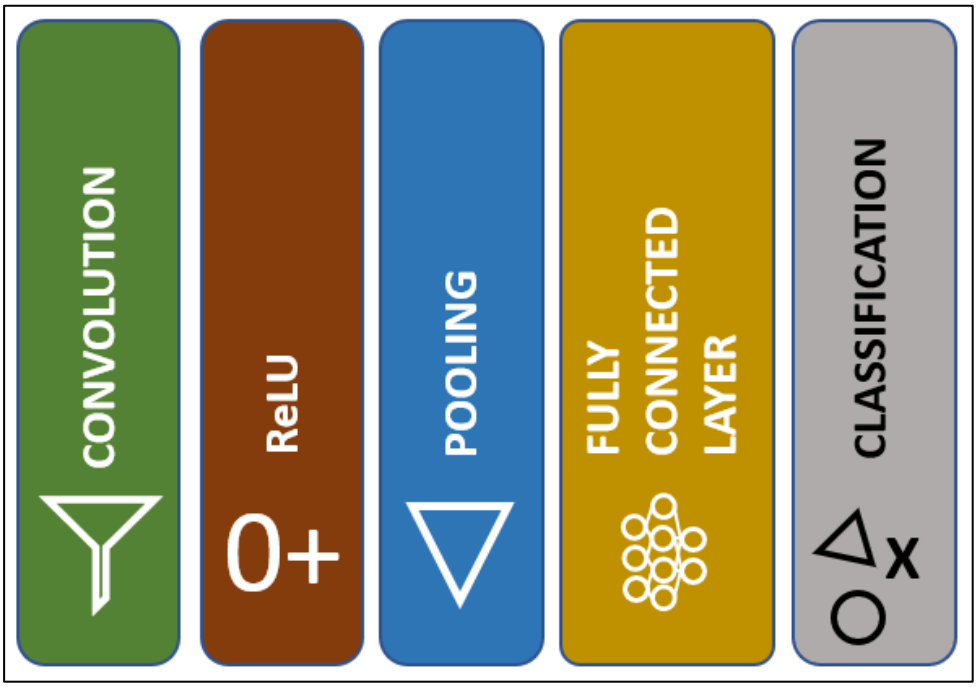


Figure 10: CNN Layers

3.3.2.1. Convolutions Layer

Convolutions have given the name to this architecture, as it is the core operation of a convolutional neural network (CNN). Its purpose is to perform feature extraction. It is executed by applying the same sets of values – kernel or filter - smaller than the input across all possible fitting positions on the input. This is done by calculating the product of each relative position on both the input and the filter (or kernel) (Yamashita et al., 2018). Taking the image in Figure 11 as input of 7 pixels by 7 pixels and calculating the convolutional result for a filter, the result will be a new 2D array with *x* columns and *y* rows.

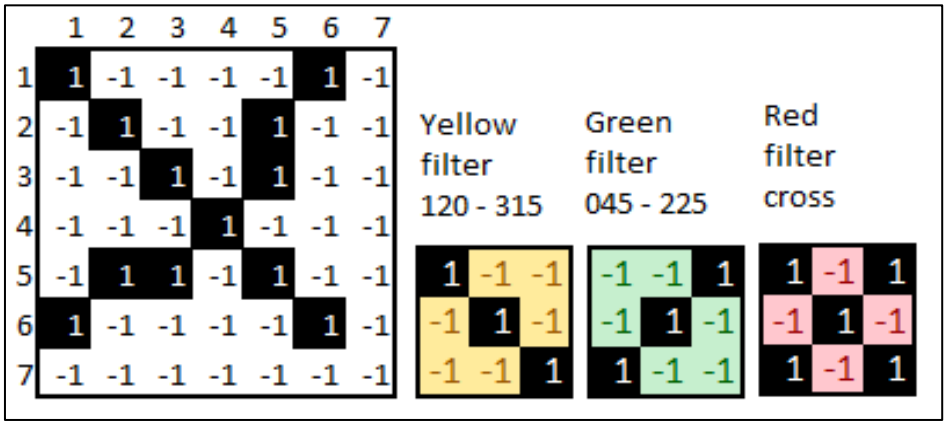


Figure 11: On the left: Representation of an imperfect X on a 7x7 pixels image. On the right: Three filters are identifiable by their background color. The yellow filter represents a diagonal line also referenced as the angles 120 and 315 measured from the center box. The green filter represents a diagonal line and is also referenced with the angles 45 and 225 measured from the center box. The red filter is the representation of a perfect X.

The convolution resulting array size can be determined by applying Equation 2 and Equation 3. It has 5 columns and 5 rows.

In a convolution, the filter slides horizontally and vertically over the input vector on every combination possible. This sliding operation is given the designation of Convolution (Indolia et al., 2018). The convolution result for the *yellow filter* is shown in Figure 12:*E* where it is depicted the higher values forming a diagonal from the convolution result map position 1,1 to position 5,5. The position 1,1 of the convolution result map has the value 1.0 denoting the perfect match of the filter on that set of pixels. The center (position 3,3 of the convolution result map) has the lowest value of the diagonal as the cross of the other diagonal is not represented on the yellow filter and so inhibiting that filter value for that position. The set of feature maps resulting from each application of the filter to the input is called the convolution layer.

Equation 2: Number of rows of the convolution result.

I_R : *Input number of rows*

F_R: *Filter number of rows*

$$y = 1 + I_R - F_R$$

Equation 3: Number of columns of the convolution result.

Ic : *Input number of columns*

Fc: *Filter number of columns*

$$x = 1 + Ic - Fc$$

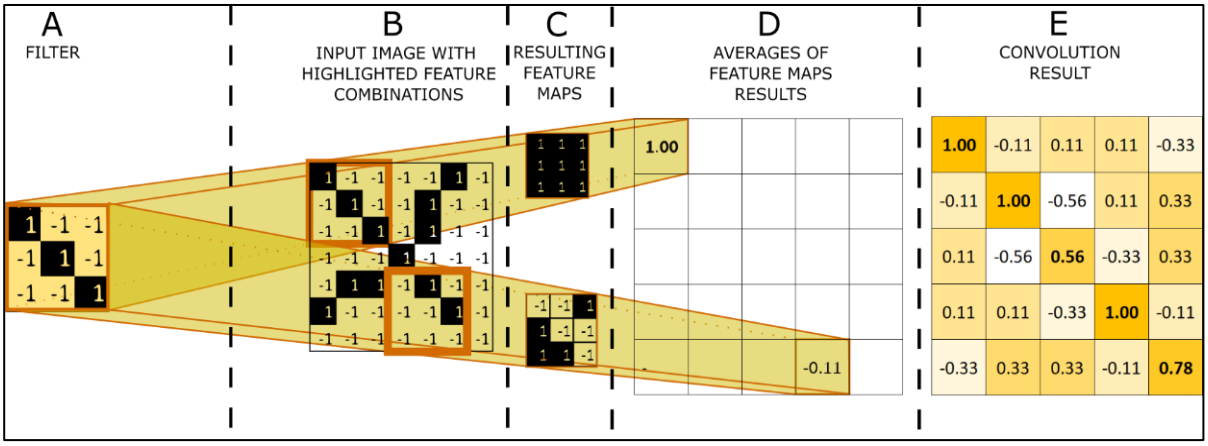


Figure 12: Demonstration of the application of the convolution operation with the yellow filter and the convolution result for the application of the yellow filter on the input image. The filter slides over each 3x3 (filter size) combination of features multiplying its values with the filter values at the same position. The multiplication results in a feature map. The result of the convolution is the resulting averages of each feature map.
A: Filter.
B: Input image with highlighted feature combinations.
C: Resulting feature maps.
D: Averages of feature maps results.
E: Convolution result.

As for a mathematical approach, the convolution can also be calculated by applying Equation 4 on the input array for each filter generating new feature maps.

Equation 4: Formula to calculate each convolution array position result

$$\left(\sum_{i=1}^{F_c}\sum_{j=1}^{F_R} F_{i,j}\cdot I_{x-1+i,y-1+j}\right) / F_c \cdot F_R$$

F_R: *Filter number of rows*

Fc: *Filter number of columns*

x: *Column Index of the first row and column of the overlayed filter on the input array*

y: *Row Index of the first row and column of the overlayed filter on the input array*

3.3.2.1.1. ReLU

As mentioned in section 3.3.1.1, “Rectified Linear Unit” is an activation function to normalize the network values. This is also useful to apply on convolution layers. By setting the minimum value to 0 it can have an inhibitory (deactivating) effect on features improving the network performance.

The application of ReLU on the filtered feature maps has the effect shown in Figure 13 with all negative values being turned to zero. In the same figure, the ReLU result in yellow, denotes a clear diagonal with the same slope as the yellow filter. It is also shown by the green feature map a curvy pattern formed by the highest values (0.56); this green curvy pattern still resembles the shape of the diagonal of the green filter; the arch-shaped arms of the imperfect X reveal the exciting influence of this filter on the feature map. As for the convolutions with the red filter, the image center (with 1.00 value) reveals the core of the cross – the arms intersection.

1.00	0.00	0.11	0.11	0.00	0.11	0.00	0.11	**0.56**	0.11	**0.56**	0.00	0.11	0.11	0.00
0.00	**1.00**	0.00	0.11	0.33	0.00	0.11	0.00	**0.56**	0.00	0.00	**0.56**	0.00	0.11	0.00
0.11	0.00	**0.56**	0.00	0.33	0.11	0.00	**0.56**	0.00	0.33	0.11	0.00	**1.00**	0.00	0.33
0.11	0.11	0.00	**1.00**	0.00	**0.56**	**0.56**	0.00	0.11	0.00	0.11	0.11	0.00	**0.56**	0.33
0.00	0.33	0.33	0.00	**0.78**	0.11	0.00	0.33	0.00	0.33	0.00	0.00	0.33	0.00	0.33

Figure 13: ReLU application on features map for each of the filters, yellow, green, and red.

3.3.2.1.2. Pattern Learning Process

The CNN kernels (filters) are patterns composed of values. The filters mentioned before (Figure 11) are not pretrained or previously conceived. The learning of these filters results from back-propagation just like in the ANN. From the fully connected layer output nodes' loss is backpropagated to the convolution layers (Indolia et al., 2018). These filters shaped as diagonals and crosses are convenient for explaining the process and were completely made up by me just to illustrate the process, a human-friendly tool. The actual patterns may be irrelevant for a human to understand. CNNs also learn the patterns based on weights and the ANN learning process which has been previously described in Section 3.3.1.2.

3.3.2.2. Pooling Layer

The role of the pooling layers is to pick a representative sampling of the feature map to reduce the number of features and so cut the number of parameters fed to the FCL, improving the performance.

Max pooling is one of the most used algorithms used in the pooling layer. It consists of dividing the feature map into non-overlapping patches of the same size and shape, for instance, 2x2 pixels, and selecting the max value of each patch (Ting et al., 2019) which will help to excite the sharpest elements in an image and inhibit the less sharp objects. As shown in Figure 14:*C* the much smaller max pooled feature map still clearly denotes a diagonal shape.

The amount of information retained is indirectly proportional to the size, i.e.., as the size increases the relevance decreases. In edge situations, if we use only one patch with the same size as the whole map, all features are lost and replaced by only one feature with one value. On the opposite side, if the size and shape are 1x1 pixels, no information is lost. But, of course, there is no dimensional reduction which is the purpose of the pooling operation.

The resulting feature maps for all the max pooling operations applied to the convolutions resulting maps for each filter are shown in Figure 15 where is noticeable the similarities with the original image.

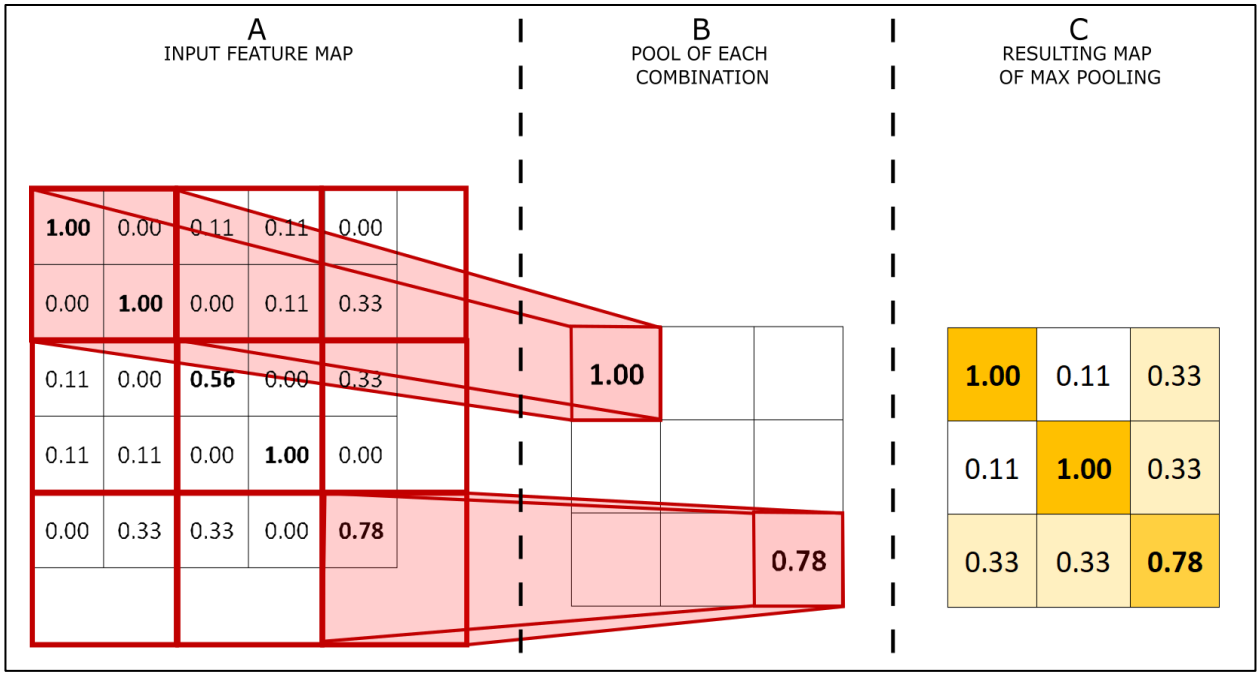


Figure 14: Max pooling result for the feature max resulting from the yellow filter convolution operation.
A: Input feature map.
B: Pool of each combination.
C: Resulting map of max pooling.

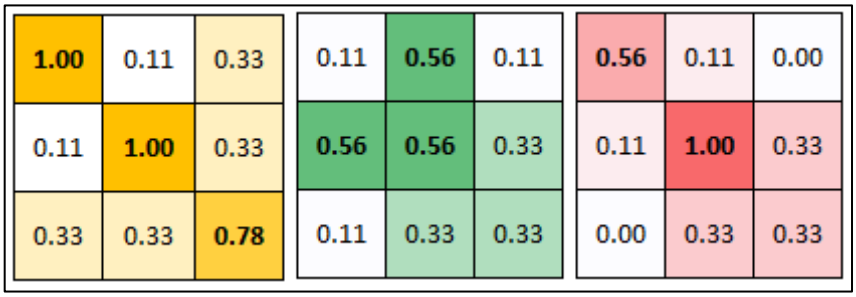


Figure 15: Max pooling resulting feature maps for each of the filters.

3.3.2.3. Feature Extraction Discussion

As is, for this specific case, filter size and pooling strategy, after applying one convolution layer and one pooling layer, across all resulting feature maps from the application of all three filters there is a total of 3 * 3 * 3 features which is 55% of the original features.

More convolutional or pooling layers can be stacked after the pooling to decrease even more the number of features, or other order could have taken place. There could have been applied two convolutional layers and the pooling after, or a convolutional layer then pooling then another convolutional layer, or even a convolution layer followed by two pooling layers. The point is, there is no static solution to all problems.

Each problem can have 1its stacking configuration so besides the FCL both convolutional and pooling layers can be reordered and stacked.

3.3.2.4. Fully Connected Layer

The fully connected layer (FCL) is an ANN and the last layer of CNN. After features are extracted in the convolution layers the feature maps are flattened into a 1-dimension arrays and supplied to the FCL (Yu et al., 2021). It is an Artificial Neural Network to which the flattened features array is fed as the ANN input and in turn, its output layer nodes will vote the class that most likely is represented by the input features.

3.3.3. Summary

One of the greatest advantages of convolutional neural networks compared to other standard ANN is the fact that the feature extractions is carried out by the convolutional layers and pooling layers. Where the V-ANN for image processing would use as input features all pixels flattened out as a 1-dimension array, CNNs have the capability and preserve the 2-dimensional attributes. This allows to preserve significant image features, such as shape and contrasts, and simultaneously significantly reduces the size of the feature maps. As demonstrated on a small image of 7x7 pixels (Figure 11), one single convolutional layer and one pooling layer managed to reduce the number of features by 55%.

3.4. Popular Metrics for Object Detection Networks

As depicted in Table 2, the most popular object detection models use Average Precision (AP), mean Average Precision (mAP), and Intersection over Union (IoU) metrics (Padilla et al., 2021).

AP is calculated for each object class. The mAP is calculated for the entire model, over all classes, and is an extension of AP. Finally, IoU is a metric to evaluate the similarity of areas, in object detection on, bounding boxes. Intersection over Union is calculated to distinct areas (Equation 5).

Equation 5: Intersection over Union

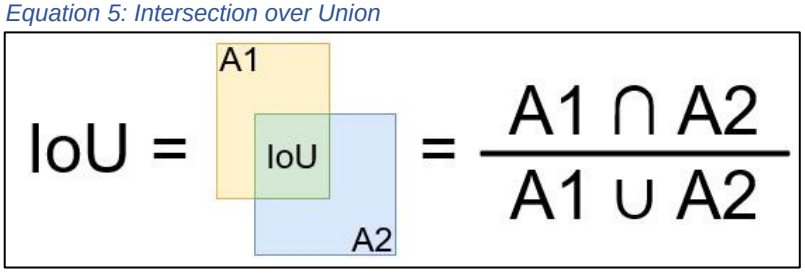


Table 2: Popular object detection methods along with the datasets (common objects dataset, not medical) and metrics used to report their results. This information is retrieved from the actual models' publications (Padilla et al., 2021).

Method	Benchmark Dataset	Metrics
CornerNet	COCO	AP@[.5:.05:.95]; AP@.50; AP@.75; APS; APM; APL
EfficientDet	COCO	AP@[.5:.05:.95]; AP@.50; AP@.75
Fast R-CNN	PASCAL VOC 2007, 2010, 2012	AP; mAP (IOU=.50)
Faster R-CNN	PASCAL VOC 2007, 2012	AP; mAP (IOU=.50)
Faster R-CNN	COCO	AP@[.5:.05:.95]; AP@.50
R-CNN	PASCAL VOC 2007, 2010, 2012	AP; mAP (IOU=.50)
RFB Net	PASCAL VOC 2007	mAP (IOU=.50)
RFB Net	COCO	AP@[.5:.05:.95]; AP@.50; AP@.75; APS; APM; APL
RefineDet	PASCAL VOC 2007, 2012	mAP (IOU=.50)
RefineDet	COCO	AP@[.5:.05:.95]; AP@.50; AP@.75; APS; APM; APL
RetinaNet	COCO	AP@[.5:.05:.95]; AP@.50; AP@.75; APS; APM; APL
R-FCN	PASCAL VOC 2007, 2012	mAP (IOU=.50)
R-FCN	COCO	AP@[.5:.05:.95];AP@.50; APS; APM; APL
SSD	PASCAL VOC 2007, 2012	mAP (IOU=.50)
SSD	COCO	AP@[.5:.05:.95]; AP@.50; AP@.75; APS; APM; APL; AR1; AR10; AR100;ARS; ARM; ARL
SSD	ImageNet	mAP (IOU=.50)
Yolo v1	PASCAL VOC 2007, 2012; Picasso;People-Art	AP; mAP (IOU=.50)
Yolo v2	PASCAL VOC 2007, 2012	AP; mAP (IOU=.50)
Yolo v2	COCO	AP@[.5:.05:.95]; AP@.50; AP@.75; APS; APM; APL; AR1; AR10; AR100;ARS; ARM; ARL
Yolo v3	COCO	AP@[.5:.05:.95]; AP@.50; AP@.75; APS; APM; APL; AR1; AR10; AR100;ARS; ARM; ARL
Yolo v4	COCO	AP@[.5:.05:.95]; AP@.50; AP@.75; APS; APM; APL
Yolo v5	COCO	AP@[.5:.05:.95]; AP@.50

4. MATERIAL AND METHODS

4.1. Dataset

As mentioned in the introduction section, the objective of this thesis is to explore and validate algorithms and methods to support the detection of BC pathological lesions in MRI. Therefore, after reviewing/visiting several datasets released for the public domain, it was decided to use the Duke Breast-Cancer-MRI dataset (a compilation of dynamic contrast-enhanced magnetic resonance images of breast cancer patients with tumor locations), available at "The Cancer Imaging Archive" (TCIA) (Clark et al., 2013; Saha et al., 2018) . This dataset comprises a collection of 922 positively diagnosed BC patients' (biopsy proven) cases fully annotated and anonymized developed by the Duke Hospital, Durham, North Carolina, USA. The patient's age range is 21 to 89 years old, and the average age is 52 years old. Annotations were performed by 8 radiologists to whom the cases were randomly assigned. The pathological lesions (tumors) annotations are identified using 3D bounding boxes delimited by 2D coordinates, plus a set of slices where the tumors were found, as depicted in Figure 16. For each patient are available 5 or 6 image sequences (series). A sequence is a series of radiofrequency pulses, each one with its specific settings, resulting in

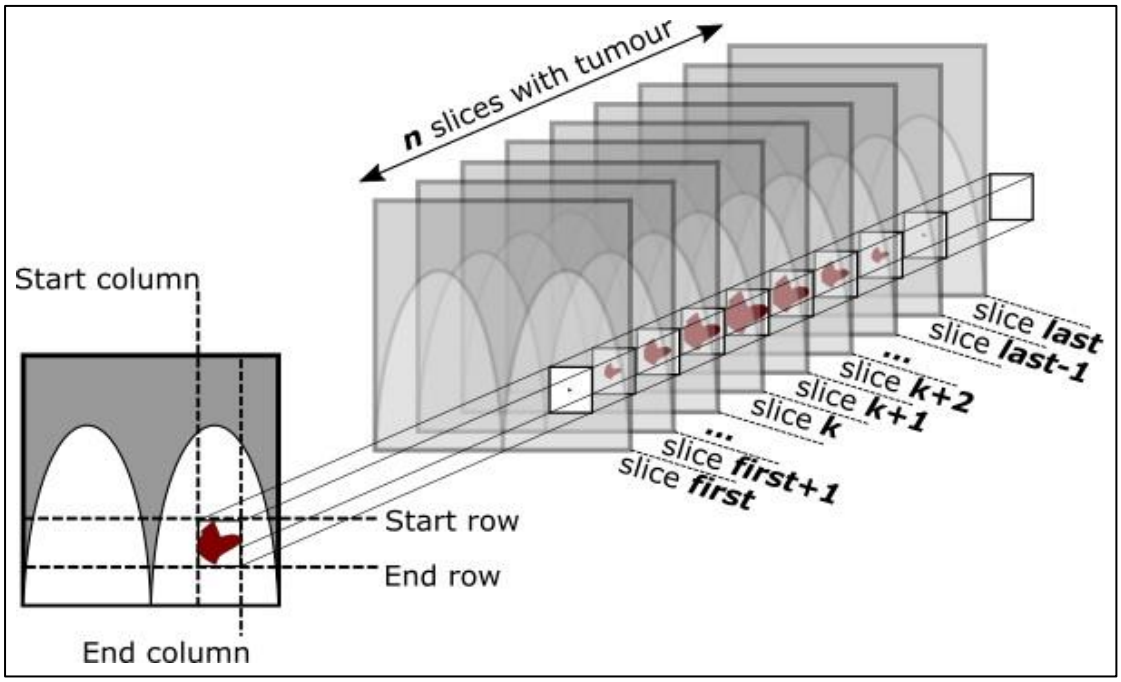


Figure 16: Illustration of the annotation schema for the bounding boxes delimiting the tumor. The first and last few slices may have loose bounding boxes. Effect of lesions edges on the 3D boxes.

a set of images (slices). This dataset includes five to six sequences of pre-operative dynamic contrast enhanced MRI. A non-fat saturated T1-weighted sequence, a fat-saturated gradient echo T1-weighted pre-contrast sequence, and mostly three to four post-contrast sequences. The non-fat saturated MRI sequences are not to be used (Saha et al., 2021) as they do not match the annotation.

The MRI scans were performed using equipment from 2 manufacturers, GE Medical Systems and Siemens, 8 distinct models were used, as illustrated in Table 3.

Overall, the dataset is composed of a total of 772439 images (slices) distributed by 922 patients, an average of 838 images per patient. The complete dataset occupies a disk storage space of approximately 342 Gigabytes, each DICOM image has an average of 443KB. All images are square in shape and the size varies between 320x320 pixels (33 cases), 448x448 pixels (261 cases), and 512x512 pixels (628 cases). However, the size of the images remains the same in all images from the same patient (patient case).

It can be observed that tumor features, shape, and size, differ highly depending on the type and subtype. The minimum number of slices where a tumor is identified/detected in a patient case is 2 slices and the maximum number of slices is 131. A negative feature of this dataset is that each patient only had one tumor annotated. The radiologists only annotated the largest biopsied tumor of each patient. Meaning that in cases where patients have more than one tumor, the smaller tumors (even if confirmed by biopsy) were not annotated at all (Saha et al., 2021).

Therefore, this can cause undesired False Positives (FP) occurrences (Type I Error) as result. It's expected to have mismatched detections. Without ground truth annotations it's impossible to validate these FPs. There is no means to assess whether false positives are real or just a result of the lack of annotation.

This inability to evaluate FP and False Negatives (FN) tumor-wise detections leads to the incapacity to calculate precision and recall tumor-wise. As the dataset doesn't include patients' cases without BC pathological lesions, it is only possible to measure True Positives (TP) and FN patient-wise, in which we can compute patient-wise metrics.

Table 3: Case count by MRI equipment model and manufacturer.

Manufacturer	Model	# Cases
GE Medical Systems	Optima MR450w	98
	Signa Excite	10
	Signa HDx	272
	Signa HDxt	248
Siemens	Avanto	179
	Optima MR450w	98
	Trio	1
	Trio-trim	57

4.2. Proposed Method

The proposed method aims to train CNN models for learning breast cancer pathological lesions (tumor patterns) using (golden standard) annotated image sets. The main goal is to improve the detection of pathological lesions in MRI, i.e., new (unannotated) images. The method includes three main steps (Figure 17): preprocessing, training, and evaluation (metrics assessment). All three steps can be configured according to several settings by editing specific configuration files described throughout this section. The prototypes for these configurations can be observed in section 8 “Appendix”. The main contributions are in the preprocessing step, but also the setting up of an optimal evaluation metrics approach.

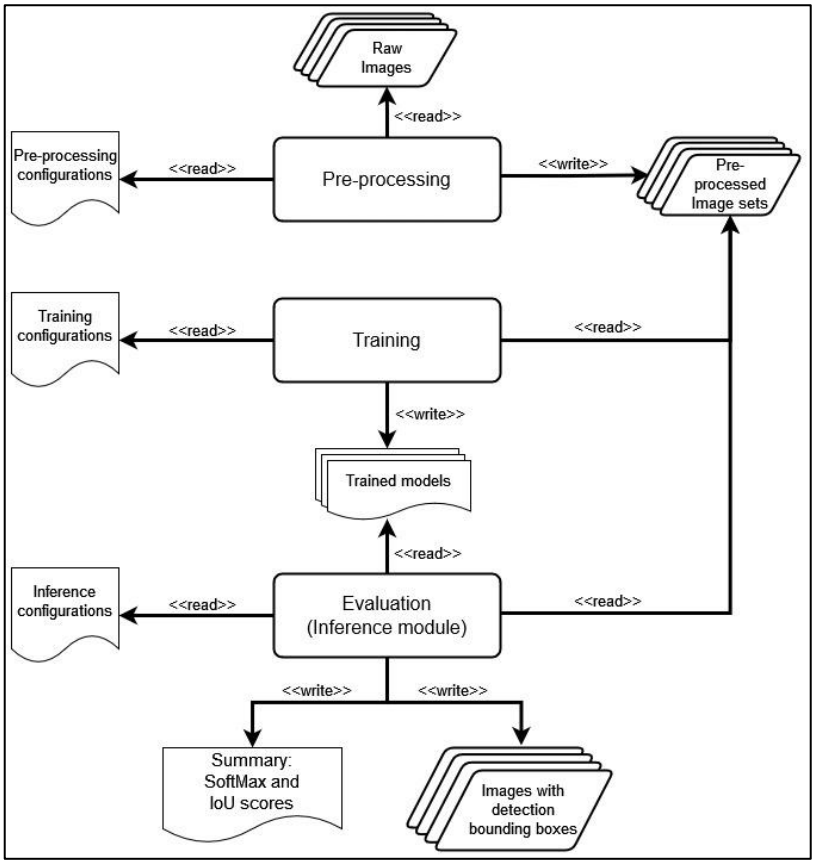


Figure 17: High-level workflow of the proposed method.

The focal point of this project is not to evaluate the network performance - which is already thoroughly tested for common objects datasets and very well documented and rated (Faster R-CNN Inception ResNet V2 (Huang et al., 2017; Lin et al., 2017; Szegedy et al., 2016)) - but rather help physicians detect BC. Specifically, point out where - which sequence and slice - the tumor is most noticeable and its classification score. The most negative weakness – or damaging aspect - of the medical assisting AI model are the false negatives (Type II errors) which can produce misleading evaluations, i.e., not identifying a tumor when it exists. The regions detected by the model - possible breast cancer lesions - should be pointed out to radiologists, with indication of the specific sequences and slices, so that further analysis can be performed.

4.2.1. Pre-processing

Pre-processing implementation options are further described in section 4.3.2.

This step aims to prepare the dataset before starting the training process. It includes loading, selecting, extracting, and preprocessing the necessary information from the DICOM files' digital content (MRI images) and associated image metadata. This step, divided into several stages (Figure 18), aims to prepare the dataset before for training process. It includes loading and interpretation of DICOM files and their image-associated metadata. This step is crucial (phase) as DICOM image metadata includes specific features about the scanning options used.

The images are all resized to 448x448 pixels, nevertheless, the original size is not discarded given that it is necessary to calculate the coordinates of the bounding boxes for the new image size. With all the data/information needed, well-identified and organized, we build an annotation XML file that is generated for each image indicating the bounding box coordinate

To increase image diversification, several operations of data augmentation are performed. This process typically covers transformations on images by inverting and rotating images and adjusting brightness and contrast. This process is further described in the next section as it will be processed in the ANN model.

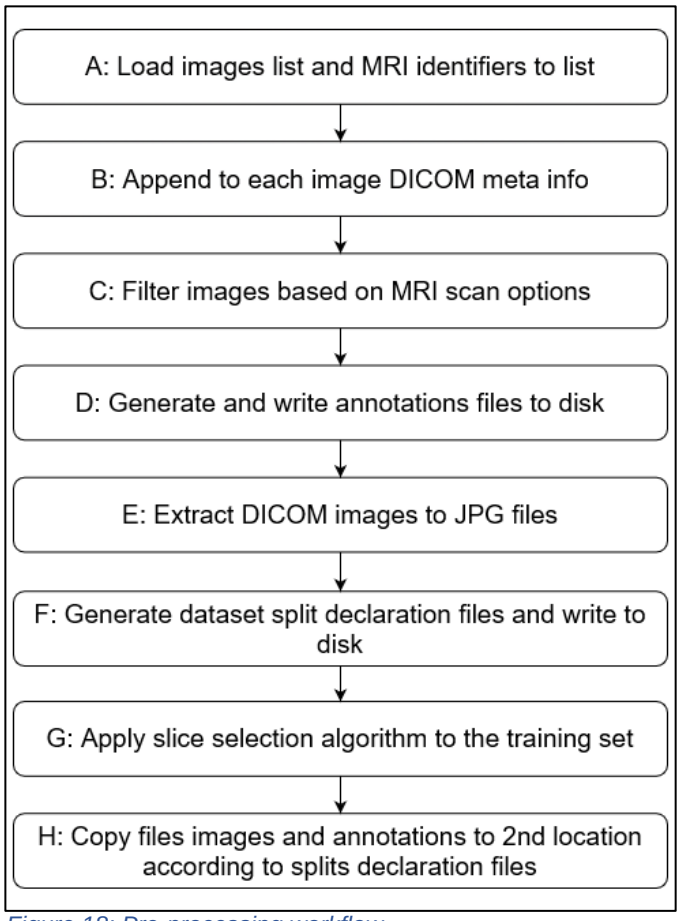


Figure 18: Pre-processing workflow

To be ready for the training process, patients' cases are separated into training and test sets. This process must be done patient-wise, rather than image-wise, to ensure that there are no images of the same patients in both training and test sets. Image selection is also part of pre-processing. The purpose of choosing only a few images from each patient is to exclude images that have too little information and too much noise. This process is only performed after the dataset is split into training and test subsets (patient-wise). This is an important aspect to highlight, the fact that we only want to filter out the slices with a tumor to include into the training set, to avoid bounding box looseness issues. The proposed slice selection method is not applied to the test subset because it should take all slices where a tumor is identified into consideration, i.e., the inference process doesn't take into account the bounding box annotations so there is no looseness, the model should be able to detect all tumors. Again, the loose bounding box is only a problem for the training process since it is desirable not to introduce noise into the model.

4.2.2. Training: Faster R-CNN Approach

Deep Neural Networks (DNN) have proven their efficiency in object detection; therefore, it was a decision to approach the problem by making use of already existing networks. In specific, it is used the Faster R-CNN model implementation, composed by two modules; the first module implies a CNN for region proposal and the second module for object detection on the proposed region (Ren et al., 2015). Its architecture is illustrated in Figure 19.

The chosen model is the Faster R-CNN Inception ResNet V2 which combines Residual Network (ResNet) architecture, which addresses the exploding/vanishing gradient problems by introducing normalization layers (Residual Blocks) allowing to build of deeper neural networks (He et al., 2015) and, on the other side, this model also uses Inception V3 modules. Inception networks stack multiple inception modules that have in their architecture extra convolution layers of 1x1 filters (Szegedy et al., 2015) allowing

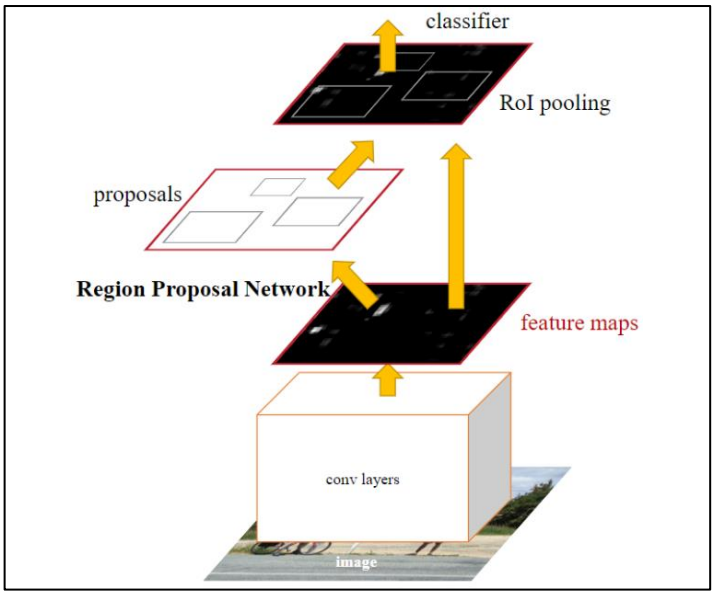


Figure 19: Faster R-CNN high-level architecture (Ren et al., 2015).

dimensionality reduction and in turn saving heavily on computational power, hence, allowing to build of deeper and more efficient neural networks.

As the goal is tumor detection, there is only one object (binary class, i.e., pathological, or non-pathological lesions). The model produces a binary classification. For each region submitted to the classifier, either there is or there isn't a tumor. The classifier produces a SoftMax probability score. In this work, only scores equal to or greater than 0.5 are considered detections.

4.2.2.1. Data Augmentation

As stated in the previous sub-section (4.2.1 Pre-processing), the data augmentation is performed by the pre-implemented network. So, it is crucial to configure the network to run extra epochs to process sufficient times each image as well as each image's tweaked versions. The data augmentation procedures used on this project are horizontal flip, vertical flip, 90º rotation, brightness adjustment, and contrast adjustment. Inversion and rotation contribute greatly to the diversification of the data and to covering a larger number of possible scenarios. Also, MRI images are produced with different orientations and even with the patients in different positions, as in Figure 20 slices *A* and *B*. In some cases, patients are scanned starting from the feet to the head and in other cases from the head to the feet. Another particularity, regarding the annotations, is that some bounding boxes include the boundary between the breast mass and the air, i.e., part of the bounding box is beyond the breast mass, as depicted in Figure 20 slice *C*.

This will cause some filters in the convolutional layers to learn these patterns. Overall, by rotating and inverting the images we can generalize these cases to patients being examined in different positions.

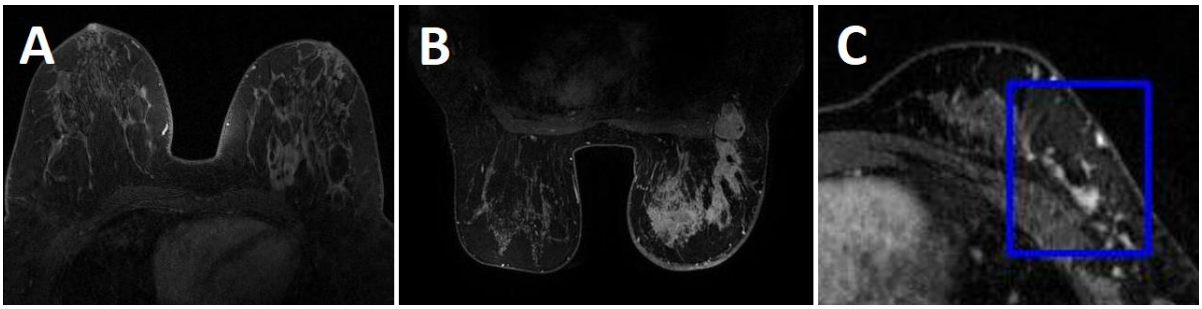


Figure 20: Three MRI slices with tumors from different patients. Slice C is marked without a bounding box surrounding the tumor. The top left vertex of the bounding box in slice C is clearly beyond the breast mass.

4.2.3. Evaluation

Traditionally, in CNNs, and Machine Learning systems in general, performance is measured by evaluating the scores of the trained models with some pre-established metrics. This evaluation is done by assessing the model on the elements of a previously set aside subset of the data used in the process, commonly referred to as the test dataset.

For our problem of detecting breast pathological lesions (tumors) in patients, the element is not each object of any image (concerning the object – object-wise - or, in this case, concerning the tumor – tumor-

wise). In this problem, a specific set of images is targeted, i.e., a set of images of each patient (concerning the patient – patient-wise). Assuming this position, there is no interest in achieving optimal scores on every image of every patient. In this work, our goal is to find the best score among all the images (slices) on each patient's case to reduce the false negatives (Type II errors) to a minimum.

Concerning object detection networks, the most popular metrics to evaluate the produced models are AP, mAP, and IoU, having each object as a base unit (object-wise). As explained in section 4.1 "Dataset", we have used a fully annotated and anonymized dataset comprising a total of 922 patient cases, in which radiologists have only identified (annotated) one pathological lesion (tumor) per patient case, even in those cases where a patient has more than one biopsied tumor. With this, in this work, we decide to perform an evaluation patient-wise with adjusted metrics to the information available on the dataset. In this sense, we have used a patient-wise metrics approach that includes accuracy, recall, and IoU.

Regarding the IoU metric, it is expected to have poor IoU results due to the ground truth bounding box looseness issue observed in the 3D representation of the tumor annotations (Figure 16). It is also noticeable in Figure 21:*B* where can be seen the oversized ground truth bounding box clearly larger than the actual tumor.

The opposite effect may also occur. In many cases the trained samples include noisy areas (background) surrounding the actual lesion, as depicted in Figure 21:*B*. This effect is due to the existence of several slices with loose bounding boxes being fed to the model. Similarly, training will be performed using the 10-fold cross-validation method which is explained in more detail in the section 5.4 "Cross-Validation".

The metrics will be evaluated for all cross-validation folds. Standard deviation will also be calculated to estimate more clearly how can the model be expected to perform in production settings (i.e., clinical workflows).

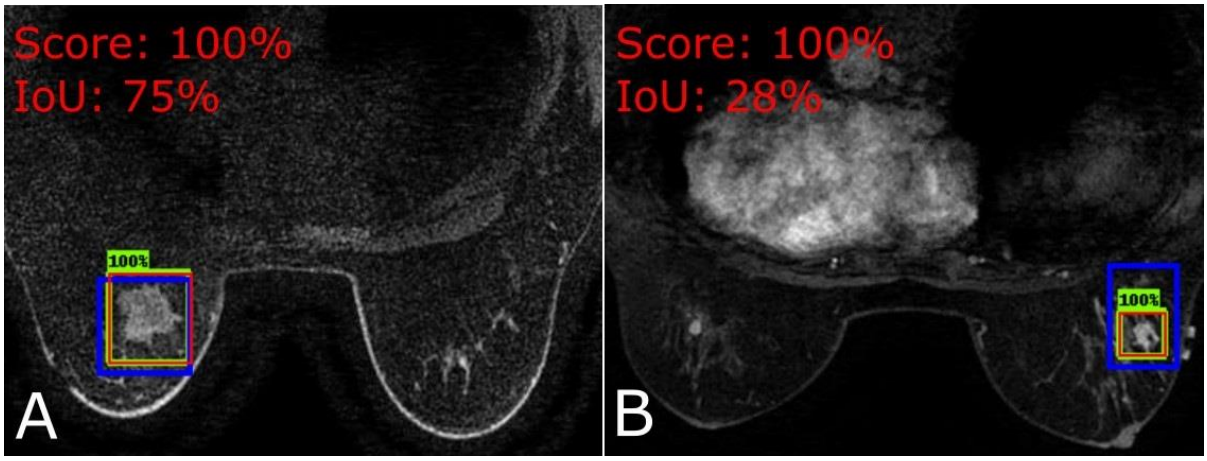


Figure 21: Blue box: ground truth; green box: detection; red box: true positive. A: Slice from a patient with code 45 denoting a tight fit to the tumor. B: Slice 65 from patient 640 denoting a loose fit to the tumor.

4.3. Implementation

4.3.1. Environment Setup

Table 4: System specifications with CPUs, GPUs, and memory details. Cluster B belongs to the VISTA Lab Research Center of the University of Evora, Portugal.

		Cluster A	Cluster B (Vision at VISTA Lab)
Nodes		*1 Node*	*2 Nodes* Each node
CPU	*Designation*	Intel i7-12700K	Dual AMD Rome 7742
	Cores	12	128
System memory		64GB	1TB
GPUs	*Designation*	GeForce RTX 3060	NVIDIA A100 Tensor Core GPU
	N. of GPUs	1	8
	Cores	3584	6912 (total 55296)
	Memory	12GB	40GB (total 320GB)

Computing is performed by 2 clusters of systems, cluster A and cluster B. Cluster A consists of a single-node workstation with 1 GPU and cluster B is a supercomputer with 2 nodes of 8 GPUs each, as described in Table 4. While cluster A was entirely allocated to this project, cluster B is a shared resource and for that reason, it would not be possible to guarantee consistent use of all the capacity of an entire node throughout all the experiments. Therefore, only 50% of the capacity of one of the two nodes (4 GPUs) was used to ensure consistent usage throughout the project.

Cluster *A* was used for all software programming, data preprocessing, parameter tuning, model inference on new data, and metric calculations. Cluster *B* was used exclusively for model training with all artifacts pre-generated. Although cluster *A* also has a configuration suitable for training deep learning models, it is, in comparison, considerably slower than cluster *B*. Using cluster *A* for this task would result in a significant delay concerning project planning. Therefore, the most intensive GPU work was performed on cluster *B*.

A benchmark was executed to find the performance of the cluster using the same dataset. As the goal of the benchmark is to find each cluster's performance for the available resources, this test was carried out using all GPU computational power of cluster *A*. As for cluster *b*, for the reasons described, we only use 4 of the total GPUs. Both configuration and results for the benchmark can be found in Table 5.

In all phases, the scripts developed were programmed using Python 3. Python is one of the best-known programming languages for data science projects. It is a high-level language suitable for various types of goals. Its multi-paradigm characteristics allow different approaches to be used for different challenges, which is ideal for this project. It also counts for a considerable community, which means that it can be easy to find help in any difficulties that may arise. The Python environment and packages were managed using *miniconda* version 4.12.0.

For training the models, the TensorFlow 2 Object Detection API is used. It is an easy-to-use framework built on top of TensorFlow and provides a comprehensive set of tools to perform any of the tasks related to training and implementing detection models. Also, it counts on a collection of pre-trained models which facilitates the network implementation.

For pre-processing stage, several python libraries were used to help achieve the desired operations. Namely, pillow, *pydicom*, *dict2xml*, *xmltodict*, *NumPy*, *pandas*, *scikit-image,* and *OpenCV*.

Table 5: Clusters benchmark for GPUs settings using the same dataset and network configurations.

Cluster configuration				Result	
Cluster	GPUs used	Total GPU memory (GB)	Batch size	Steps per second (avg)	Images processed per second (avg)
A	1	12	2	1.77	3.54
B	4	160	32	1.90	60.8

4.3.2. Pre-Processing Implementation

One major challenge is the selection of the set of images (per patient) that improves breast tumor detection. This process is performed by a collection of scripts that generate the training artifacts fed to the TensorFlow Object Detection API. On one hand, the dataset has complex pre-processing requirements, due to its specifications described in section 4.1 “Dataset”, on the other hand, the dataset is great in size, over 340GB, which takes a lot of time to process and can result in memory shortage if handled incorrectly.

Most of the patients in this dataset have slices with tumors not tightly fit into the bounding box as depicted in Figure 21 and illustrated as well in Figure 16. In Figure 16 the initial and last slices show significantly loose tumors within the bounding boxes. This can lead to the decline of the model performance as it introduces noise that will be backpropagated to the network (model) weights (Famouri et al., 2016).

Preliminary tests, both with all slices and with too few slices, revealed difficulty for the model in learning the patterns. Removing slices is removing information, not only do we not want to remove too many slices, but we also don't want to remove too few slices which would result in the inclusion of excessive noise due to loosen boxes. To mitigate this problem, the need arose to develop a generic function capable of calculating the number of slices to remove.

By observation of the dataset, in most cases, it would be appropriate to remove half of the slices. However, since there are many smaller tumors with total slices between 2 and 10, it would not be a good approach to remove half of these images. In this sense, a logarithmic funneling function (see Equation 6) was developed for helping to dynamically select what are the slices to keep and what are the slices to remove. Of course, configured so as not to remove too many slices in the smaller tumors. In Equation 6 *a* and *b* are constant values to control the funneling effect. These *a* and *b* values were heuristically obtained by

Equation 6: Logarithmic funneling function.

$$f(x) = total\_number\_of\_slices\_with\_tumor \ * a * ln(b)$$

trials starting with *a* valuing 1 and *b* valuing 2. In the end is used *a* = 0.75, *b* = 1.9625. The resulting value of that equation is then rounded up. That will be the total amount of slices to maintain for a given patient.

To calculate the total number of slices to be removed, the value resulting from Equation 6 is subtracted from the total number of tumor-containing slices for that patient. The value resulting from the subtraction is then divided by two to get the number of slices to be removed from each side (left and right). If the result is not an integer value, the value is rounded down for the initial slices to be removed and rounded up for the final slices to be removed.

With these settings, for example, a patient with 108 slices with a tumor, would keep only 55 central slices after applying this equation. A total of 53 slices would be removed, 26 slices from the beginning of the tumor and 27 slices from the end of the tumor. For a patient with a total of 10 slices with a tumor, we would get to keep only 6 slices and would remove 2 from each side. As illustrated in Code 1 and Code 2.

Code 1: Pseudo-code: Application of the funneling function. As the result of the funneling function is a real number, the value must be converted to an integer. We do so by rounding the value up (ceiling).

```
FUNCTION: CalcNumOfSlicesToRemove
Input:
  Integer: total_number_of_slices_with_tumor
  Integer: total_number_of_slices
Output:
  Integer


BEGIN
  Real funnel_result := funneling_function(total_number_of_slices_with_tumor)

  RETURN total_number_of_slices - ceil(funnel_result)
END
```

Code 2: Pseudo-code: Based on the total number of slices, this algorithm calculates the number of slices to remove from each end of the slices with tumor range. When the total number of slices to remove is an odd number, we round down (floor) the number of slices to remove from the beginning of the range and round up (ceil) the number of slices to remove from the end of the range.

```
FUNCTION NumberOfSlicesToRemoveFromEachSide
Input:
  Integer: total_number_of_slices_with_tumor
  Integer: total_number_of_slices
Output:
  Tuple(integer, integer)


BEGIN
  Integer number_of_slices_to_remove := CalcNumOfSlicesToRemove(
      total_number_of_slices_with_tumor, total_number_of_slices
    )
  Real number_of_slices_to_remove_on_each_side := num_of_slices_to_remove / 2

  RETURN
    (
       floor(number_of_slices_to_remove_on_each_side),
       ceil(number_of_slices_to_remove_on_each_side)
    )
END
```

In terms of the methodology for excluding slices with loose bounding boxes, for example in the tumor represented in Figure 22, we would want to exclude slices CS-4, CS-3, CS+3, and CS+4. Slice CS-2 and CS+2 will already introduce much noise, but anyway, still have some beneficial info, and the bounding boxes are not as loose as the end slices.

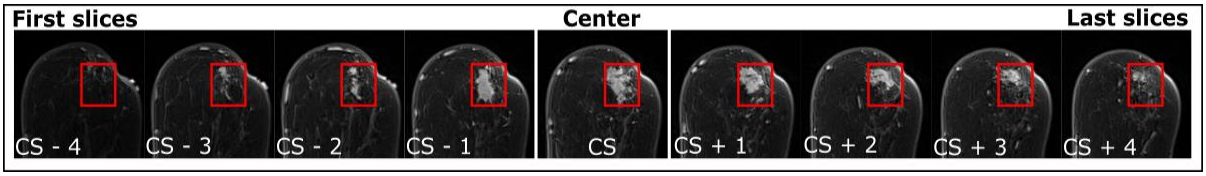


Figure 22: Illustration of bounding boxes looseness on a 9-slice tumor. Slice CS represents the central slice. CS-4 represents the first slice and CS+4 represents the last slice. The central slice (CS) is well fit to the bounding box, from CS to each of the sides, the bounding boxes have increasing looseness (Centeno Raimundo et al., 2022).

By applying this slice exclusion, patients end up removing 4 slices and keeping 5 slices, which is the result we would get from applying the set of calculations in Code 1 and Code 2 on this sample of 9 slices, as illustrated in Figure 23. As illustrated in Table 6, it is seen from the color gradients the effect of the funneling function. In the descending gradient, in green, it can be observed that gradually the total number of slices is decreasing when compared to the original count. In the minimal case - patients with tumors with a maximum count of 2 - no slice will be removed by the algorithm. As the total number of slices increases, the number of slices to be removed also increases. This is also observable in the red ascending gradients, which represent the number of slices to remove at the beginning and end of the tumor.

Given the complex requirements for pre-processing of the dataset and due to its size, the pre-processing stages are well delimited both in script files and in key phases on which data is saved to the filesystem. The stages workflow is illustrated in Figure 18.

The slice selection described in the section is only applied to the train data set. As mentioned, the goal is to reduce the number of images where the tumor was not fit into the bounding box. For the test dataset, all slices are used as the model can detect scaled versions of the object for which it was trained.

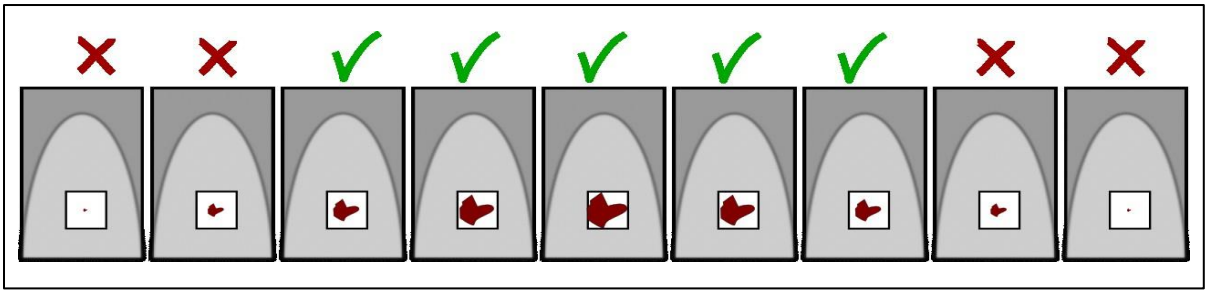


Figure 23: Illustration of the slices to be removed from a case with a tumor size distributed in 9 slices.

4.3.2.1. Pre-processing Input

This module has as input a raw dataset itself and a configuration file. The configuration file must be divided into five main sections (image, patterns, paths, dataset split, and annotation template) as the described in following child section. These settings must be part of a configuration python file to be imported by the sub-modules structured as exemplified in Code 5.

4.3.2.1.1. Image settings prototype

This configuration section holds settings related to both input and output images. Its prototype is demonstrated in Code 6 where can be specified the known image types to interpret, the image extraction extension, resize size to output, and the original mandatory configuration present in DICOM meta. As for this last setting, as referenced in section 4.1 “Dataset” the dataset only has an annotation for fat-suppressed MRI sequences, which are represented in DICOM metainformation with the initials FS or SFS.

Table 6: Calculated values by applying the funneling function on Equation 6. Some rows are skipped to save vertical space.

Patients Total number of slices	Patient Total Number of slices to remove	Number of slices to remove from start	Number of slices to remove from end	Number of slices to maintain
2	0	0	0	**2**
3	1	0	1	**2**
4	1	0	1	**3**
5	2	1	1	**3**
6	2	1	1	**4**
7	3	1	2	**4**
8	3	1	2	**5**
9	4	2	2	**5**
10	4	2	2	**6**
		...(omitted rows)...		
20	9	4	5	**11**
		...(omitted rows)...		
30	14	7	7	**16**
		...(omitted rows)...		
40	19	9	10	**21**
		...(omitted rows)...		
50	24	12	12	**26**
		...(omitted rows)...		
60	29	14	15	**31**
		...(omitted rows)...		
70	34	17	17	**36**
		...(omitted rows)...		
80	39	19	20	**41**
		...(omitted rows)...		
102	50	25	25	**52**
108	53	26	27	**55**
111	54	27	27	**57**
119	58	29	29	**61**
131	64	32	32	**67**

4.3.2.1.2. Patterns settings prototype

This configuration (Code 7) purpose is to declare the regular expression patterns for identifiers (patient codes, sequences codes, slice numbers) both raw and pre-processed sets. And, also, the patterns intended to exclude (filter out) some sequences.

4.3.2.1.3. Paths settings prototype

This configuration (Code 8) purpose is to specify the dataset working directory, raw directory name, pre-processed directory name, split sets directories, split patients' declaration file when not in cross-validation mode and cross-validation patients split declarations files when in cross-validation mode. Also, this configuration holds the path for the original annotations file which must include for each patient the bounding box coordinates and start and ending slices where the lesion can be found.

4.3.2.1.4. Annotation template prototype

This configuration is itself the prototype for the annotation files. One annotation file is created for each image. It is structured as a python dictionary object but before writing it to the filesystem it is converted to XML. It follows the Pascal VOC annotation format, and its python dictionary version is represented in Code 9.

4.3.2.2. Pre-processing Output

Depending on the stage, this module outputs a directory with all preprocessed images with corresponding XML annotation files, a set of declaration files describing which patients were included in which image set (training, validation, and testing), and directories with the actual preprocessed images split (patient-wise) alongside with their annotation XML files.

4.3.3. Model parameterization

Epoch configuration: One epoch is one network pass through the data. One step is one network passing through one batch of images, backpropagation included. Using TensorFlow Object Detection API, the model and network are configurable by editing a settings file called *pipeline. config*. It has settings for all stages of the Faster R-CNN workflow. The network has parameters to define the total number of. Batch size, which is a setting highly related to the GPU memory available, is shown in Table 5. The batch size must be divisible by the number of GPUs used. In this project, on cluster *B*, 4 GPUs were used and a batch size of 32, which means a batch of 8 for each GPU and, 5 GB of GPU memory allocated to each image. The number of epochs is a relation between batch size and the total number of steps, as demonstrated in Equation 7.

Equation 7: Calculation of the number of epochs

$$numberOfEpochs = \frac{totalNumberOfSteps * batchSize}{trainSetSize}$$

Learning Rate: The learning rate is one of the most important parameters, it controls the magnitude of the values used in back-propagation to update the weight of each node. As referred to in section 3.3.1.2.4 “Gradient Descent”, a learning rate that is too small will cause the model to take longer to train. On the other hand, a higher learning rate will train faster, but it may never reach the optimal value of the weights. This is because in this scenario the weights would be constantly updated with values higher than the difference between the current value and ideal value, and thus may never converge to an optimal value or at least to point near the optimal value.

This project is used cosine learning rate decay. The goal is to initially use larger values to update the weights, but as the model trains, the value will gradually decrease. This leads to a higher probability of finding the optimal value and thus avoiding the described effect. The decay effect depicted in Figure 24 is the visualization of the learning rate decay in a 200-thousand-step trained model.

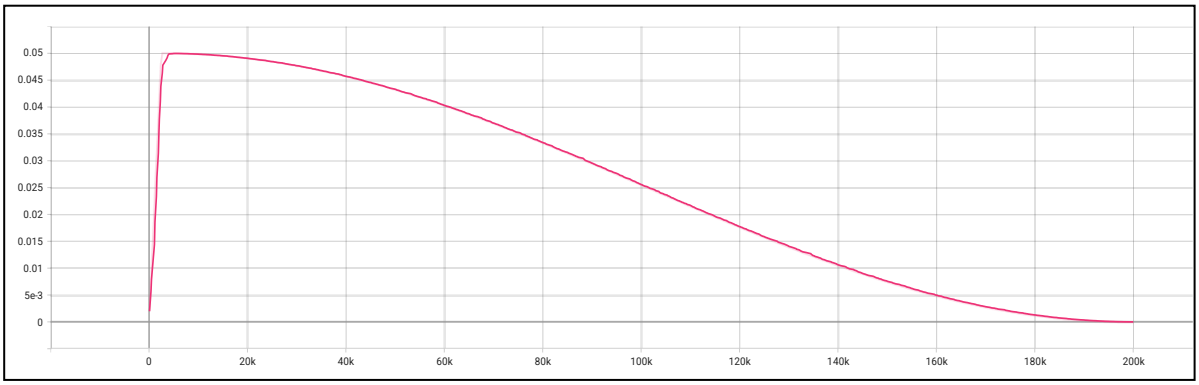


Figure 24: 0.05 learning rate cosine decay visualization on 200 000 steps trained model.

Data augmentation: As mentioned in section 4.3.2 “Pre-Processing”, the model was configured to apply the following data augmentation procedures: horizontal flip, vertical flip, 90º rotation, brightness adjustment, and contrast adjustment. This process, along with a high number of epochs, resulted in the data diversifying and expanding in number.

Classification (scoring): The network will be configured to use the SoftMax score converter on the last layer to output probabilistic value because of the output nodes.

4.3.4. Inference

This module must mark each image with metrics and bounding boxes of detection finding and generates a patient summary to export. Figure *25* is the high-level representation of the inference module. The controller is responsible for interpreting the configuration file, dealing with IO tasks, submitting images to the trained model, and calculating IoU for each image. The model classification score and detected boxes are returned by default from the detection function provided by the TensorFlow 2 Object Detection API.

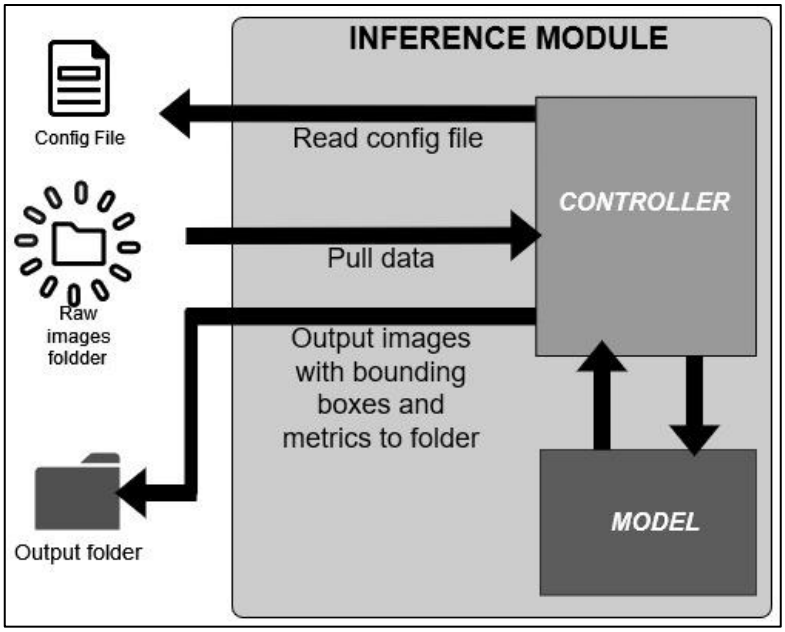


Figure 25: Inference Module: High-Level Architecture.

4.3.4.1. Inference Input

These module input artifacts are the MRI resulting images. They should be uploaded to a previously declared path. In the python configuration file (Code 10, Code 11) must be declared the regular expression pattern for identifying the sequence code, patient code, and slice number, as usual in MRI outcome files. The MRI system output is already structured in this fashion as depicted in Figure 26. For a structure like the one illustrated in Figure 26 a DICOM file path would be for instance as exemplified in Code 3 and for this structure, being the case of our dataset, we use the following regular expressions in Code 4.

Code 3: Example of DICOM image file relative path.

```
".\Breast_MRI_002\01-01-1990-NA-MRI BREAST BILATERAL W  WO-51972\601.000000-Ph1ax
3d dyn-36797\1-005.dcm"
```

Code 4: Regular expressions for identifying patient code, sequence code and slice number.

```
Patient code: “(?<=\/)Breast_MRI_\d{2,}(?=\/)”
Sequence code: “(?<=\/)[\w\d\.\-\s]+(?=\/[\w\d\.\-\s]+\.(dcm|jpg|png|tiff|bmp))”
Slice number: “(?<=\-)\d+?(?=\.dcm)”
```

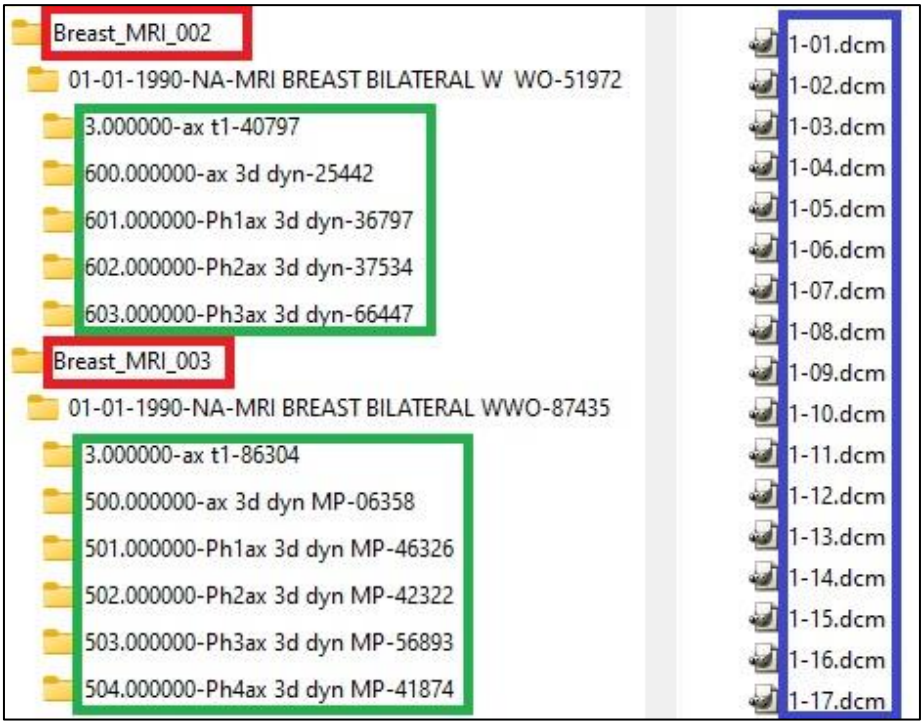


Figure 26: MRI result structure. red: patient code; green: sequence code; blue: file-names with slice number.

4.3.4.2. Inference Output

For each input image, one output image is generated. Alongside this new image, a summary file is created with the same filename pattern. This file (with prototypes as described in Code 12 and Code 13), besides the identification of the patient as well as slice number and MRI sequence id, contains the score results, both the detection score and IoU score.

5. EXPERIMENTAL RESULTS AND ANALYSIS

5.1. Data Augmentation Setup

As for data augmentation, as previously mentioned in section 4.2.2.1 “Data Augmentation”, it was used the TensorFlow Object Detection API to perform transformations randomly. The transformations applied were horizontal flip, vertical flip, 90º rotation, brightness adjustment, and contrast adjustment.

Horizontal flip: 50% chance to flip the image horizontally.

Vertical flip: 50% chance to flip the image vertically.

90ª rotation: 50% chance to rotate the image.

Brightness: Changes the image brightness up to a maximum delta of 0.05.

Contrast: Changes the image contrast by a value between 0.5 and 0.95. The value is to be multiplied by the image’s initial contrast.

5.2. Network Weights Initialization

The network weights were initialized with values from a uniform distribution within -*limit* and +*limit.* Being the *limit* calculated as demonstrated in Equation 8.

Equation 8: Limit calculation for weight initialization. The *n* value is the average of the count of units from the input and output tensors.

$$limit = \sqrt{\frac{3}{n}}$$

5.3. Preliminary model Training

5.3.1. Preliminary Models Setup

The preliminary models (hereinafter referred to as model A and model B) were trained on cluster B (Table 4) and were carried out using the dataset setups, as described in Table 7. In model A we used all slices available. In model B we used the proposed slice selection method described in 4.3.2 Pre-Processing. Both setups share the same patients' cases for training (866) and test (56) datasets. The test dataset is the same for both models, as no slices are excluded for the test set and case selection is equal. Model A counts a total of 77963 images for the training set and Model B, which used the proposed slice selection method, has 53% of model A images count for training, i.e., 41317 images.

The training process ran 160 epochs on batch sizes of 32 with a learning rate of 0.05 and cosine decay.

Table 7: Dataset setups for preliminary models. The train set image count for model A is 144.5% of the size of the model B train set. The train set image count for model B is 69.2% the size of model A train set.

Designation	GPU Cluster	Slice selection method	Image Count	
			Train set	Test set
Model A	B	All slices with tumor	77963	Same test set with 5031 images
Model B	B	Funneling method described in Pre-processing section	41317	

5.3.2. Preliminary Results

These results allow us to evaluate the effect of the slice selection algorithm as opposed to using all images and draw some conclusions.

As summarized in Table 8, Model B achieved better results in all metrics. Not only Model B has fewer false negatives (patient-wise), a better SoftMax score, and better Intersection over Union, but also it has trained in fewer images (53% of total slices of Model A) and therefore, for the same number of epochs, the Model B is faster to train taking less 27.16 hours than Model B.

Table 8: Summarization of the preliminary results.

Metric	Model A	Model B	Std Dev
Train time (hours)	56.65	**29.34**	19.315
True Positives cases	53	**54**	0.707
False Negatives cases	3	**2**	0.707
False Negative slices	2611	**2212**	282.136
Accuracy	94.64%	**96.43%**	1.27%
Average IoU	24.63%	**31.58%**	4.91%
Average Score	48.78%	**59.05%**	7.26%
Average Max IoU	59.61%	**69.40%**	6.92%
Average Score @ Max IoU	81.61%	**91.30%**	6.85%

5.4. Cross-Validation

5.4.1. Cross-Validation Setup

The model is trained using cross-validation methods with 10 folds. Thus, 10 models were trained on different combinations of the training and test sets. To produce this partition the patients are shuffled and divided into 10 batches. The training set will always have 9 batches and the test set 1 batch. For each of the 10 training phases, a different batch (1/10 of the patients) is used as the test set.

At the end of all 10 training phases, the predicted output scores and bounding boxes are used to calculate the evaluation metrics. Thus, in each training phase, the model is trained on 90% of the total patients and tested on 10% of the total patients. The 10-fold cross-validation scheme is depicted in Figure 27. A particular caveat regarding this method is that the dataset has 922 patients. As 10% of 922 is 92.2 and the rest of the patients are not divisible by 10, the 2 extra patients will be used in the training sets.

The specific slices to be used in the training sets are selected according to the procedures described in section 4.2.1 “Pre-processing”. As mentioned in the same section, the slice selection algorithm is only applied to the training set as it is advantageous to test the model on all slices of each patient to properly evaluate the model.

The experiment was conducted on the Dukes’ Hospital Breast Cancer MRI Dataset described in section 4.1 “Dataset”. As referred, we split the dataset patient-wise, and only after patient splitting process the slice selection process starts. As forementioned, each tumor has its particular shape and size, and, of course, a different number of slices where the tumor is identified. This means that each of the cross-

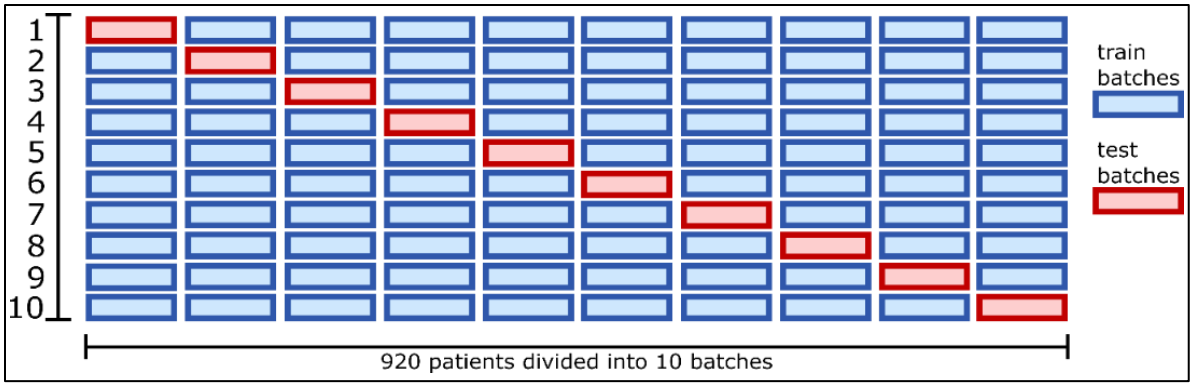


Figure 27:10-Fold cross-validation scheme.

validation folds has a different total image count, as depicted in Table 9. Nevertheless, the network parameterization was the same for all folds. The training process ran 162 epochs on 200000 steps, with a batch size of 32, a learning rate of 0.05, and cosine decay.

Although it was used in cross-validation the same hyper parameters as the previous section model B, because in the training set for the cross-validation we are using less patients and less images and the same number of training steps, the number of epochs increases accordingly.

Table 9: Cross-validation of patient and image count for each fold.

Fold	1	2	3	4	5	6	7	8	9	10	AVERAGE
Total image	39903	39844	39515	39220	39821	39347	39159	39791	39551	39773	39592
Images per patient (avg)	48	48	48	47	48	47	47	48	48	48	48

5.4.2. Cross-Validation Results

From the cross-validation exercise, two tables with the result were produced. One table containing the results uses as a base metric the average of each patient's IoU and an average of each patient's average SoftMax score (Table 13 in Appendix). The other table contains the results using as base metrics the maximums of each patient's IoU and SoftMax scores (Table 12 in the Appendix).

Both tables have the same structure. In each table, there are 20 columns, 2 for each cross-validation fold. In the header, there is information identifying the fold numbers, the count of false negatives (marked FN), and the accuracy of the fold.

For each fold, there are two columns, the first relating to the IoU metrics and the second relating to the SoftMax score metrics. The first-row values right below the header are the average of all patients; below, the second-row values (marked with @IoU > 0) are the average of all patients where the tumor was successfully detected. In Table 12 the average represents the average of maximum values for each patient; in Table 13 the average represents the average of averages of each patient.

Table 10 summarizes the contents of Table 12 and Table 13 and presents the standard deviation. Across all folds, the average accuracy in detecting the breast cancer tumor is 94.46% with a standard deviation

of 2.43%, which is a good result. The average IoU and the average score are important results to evaluate the models and their performance object-wise. But, as for the ability to assist radiologists in prescriptive analysis, they are of minimal value.

The average IoU and scores are relatively low, reflecting the failure to detect several slices, generally corresponding to the edge slices. More important are the maximum metrics which generally score well, with 89.16% for the average maximum score and 69.08% for the average maximum IoU in cases where the tumor was detected. These two metrics effectively reveal the model's ability to detect and pinpoint the slices where a tumor is best identifiable, thus providing actionable insight and aiding physicians in prescriptive analysis.

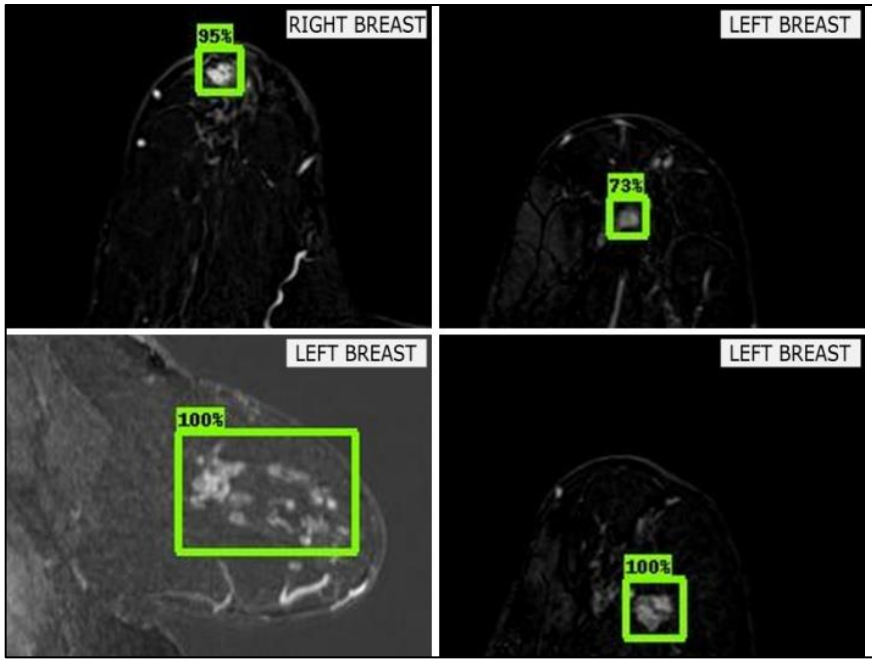
The folds which performed better are the 3rd fold and the 1st fold. The 3rd fold is for best accuracy (97.83%) and Average Score @ Max IoU (91.56%); the 1st fold is for best Average Max IoU (71.34%).

Table 10: Standard deviation calculation for each fold. Heatmap summarizing Table 12 and Table 13.

METRIC \ FOLD (where)		1	2	3	4	5	6	7	8	9	10	AVERAGE	STD DEV
False Negatives		3	4	**2**	6	3	5	6	8	5	9	5.10	2.23
Accuracy (%)		96.74	95.65	**97.83**	93.48	96.74	94.57	93.48	91.30	94.57	90.22	94.46	2.43
Average IoU (%)	all	**31.22**	28.00	27.89	29.31	27.67	28.08	28.14	27.93	27.96	23.29	27.95	1.96
	IoU > 0	**32.27**	29.27	28.51	31.36	28.60	29.69	30.11	30.59	29.57	25.81	29.58	1.77
Average Score (%)	all	**57.65**	52.20	53.07	52.56	55.44	54.18	50.60	54.76	51.24	44.95	52.66	3.42
	IoU > 0	59.16	54.57	54.04	56.06	57.26	57.25	54.06	**59.61**	54.05	49.34	55.54	3.01
Average Max IoU (%)	all	**69.02**	63.69	66.47	65.42	64.44	66.35	65.87	62.82	66.51	61.85	65.25	2.09
	IoU > 0	**71.34**	66.59	67.95	69.99	66.61	70.17	70.47	68.80	70.34	68.56	69.08	1.65
Average Score @ Max IoU (%)	all	**87.15**	84.07	86.18	85.58	86.39	85.23	83.71	79.99	82.33	81.49	84.21	2.34
	IoU > 0	90.08	87.89	88.09	**91.56**	89.30	90.12	89.55	87.61	87.06	90.32	89.16	1.44

5.5. Preliminary Results on Portuguese Patients

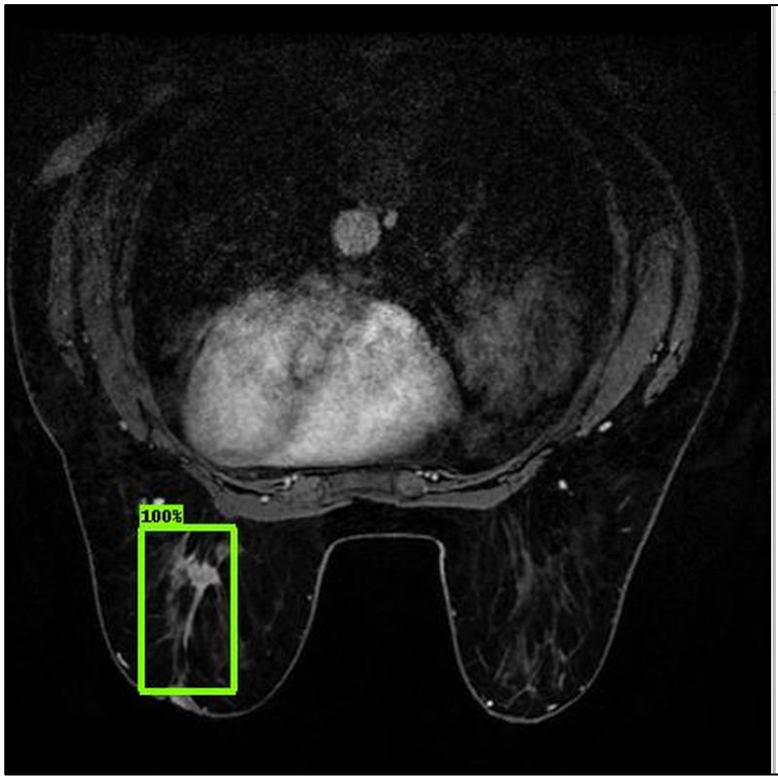
The Centro Hospitalar de Setúbal supplied us a total of 11 fully anonymized (biopsy proven) patients' cases from the historical archive for training purposes which we also used for testing model B. While 11 patients' cases are a statistically irrelevant number, the assessment achieved a high accuracy, the model B was able to correctly detect pathological lesions in all 11 cases. As can be observed in the Figure 28 and Figure 29 below, the bounding box markings created by the inference module reflect the comments in the clinical summary resulting from the MRI scans as depicted on the examples' figures. The findings are classified in terms of the Breast Imaging Reporting and Data System (BI-RADS) as well which has each category described in Table 11 (American Cancer Society, 2022).



Left breast: BI-RADS 6
Extensive proliferative lesion, including the biopsied nodule, in the mid transition of the upper quadrants/11 hours, 6 cm from the nipple, extending to the region of upper and internal tumourectomy, and aspect of focal intra ductal segmental uptake with small suspicious peri ductal nodules to the nipple.

Right breast: BI-RADS 4
Small non-specific retroareolar nodule at right, deserving a 2nd look ultrasound and ultrasound-guided microbiopsy.

Figure 28: Patient 1 MRI Scan conclusion and image with model B detection depicting the findings.



Left breast: BI-RADS 6: Nodule and ductal enhancement extending to the areola-papillary complex.
Intramammary ganglion on the left.

Right breast: BI-RADS 1: *No evidence of suspected contrast highlight.*

Figure 29: Patient 2 MRI detection results (findings) after applying the model B.

Table 11: BI-RADS categories (American Cancer Society, 2022).

Category	Definition
0	Incomplete - Additional imaging evaluation and/or comparison to prior mammograms (or other imaging tests) is needed.
1	Negative
2	Benign (non-cancerous) finding
3	Probably benign finding – Follow-up in a short time frame is suggested
4	Suspicious abnormality – Biopsy should be considered
5	Highly suggestive of malignancy – Appropriate action should be taken
6	Known biopsy-proven malignancy – Appropriate action should be taken

5.6. Results Dissemination

As mentioned in section 2.3 “Objective”, we aim to fully communicate/disseminate the results of our work in all possible formats/ways: (1) oral presentation at conferences, (2) proposals for scientific articles in journals, and (3) direct presentations to teams of radiologists/physicians in hospitals and health centers. However, at the time of writing, this objective has not yet been fully achieved.

A scientific extended abstract paper entitled "Breast Cancer Detection in MRI using a Faster R-CNN model" (Centeno Raimundo et al., 2022) was accepted and presented at the 28th Edition of the Portuguese Conference on Pattern Recognition (RECPAD 2022), held in Leiria, 28th October.

In addition, an abstract proposal entitled "Detection of breast cancer pathological lesions in MRI" has been submitted to the European Congress of Radiology (ECR) for oral presentation, however, we will only have confirmation whether it was accepted in December of 2022. Moreover, we have planned to develop and submit an article to a peer review journal (e.g., Computer Methods in Biology and Medicine, IEEE Journal of Biomedical and Health Informatics) to disseminate the main results of this thesis work.

6. CONCLUSION AND FUTURE WORK

The main result of this thesis is the development of an innovative methodology for slice selection in magnetic resonance imaging, which allows reducing background noise and computational time, building benchmarking datasets and thus, improving the accuracy of machine (deep) learning detection models. As have been demonstrated here, the proposed framework has been successfully validated in Breast Cancer settings on a dataset of 922 patients' cases.

Overall, the trained models using the 10-fold cross-validation technique achieved an average accuracy of 94.46% with a standard deviation of 2.43% in a population of 92 subjects.

The above presented results are based on heuristically computed arguments, which we consider has produced acceptable results. However, we do not assume this are the best results possible. Therefore, we recognize it is needed more experimentation. As the biggest constraint for this project being the time for trainings, finding the optimal arguments for the funneling function can take up several months.

The S.M.A.R.T objectives defined in section 2.2 were achieved. The models were trained and tested achieving the desired minimum accuracy of 95% in breast cancer tumor detection within 8 months.

A small set of MRI scans from 11 anonymized and biopsy proven patients' cases from Centro Hospitalar de Setúbal historical archive with confirmed pathological lesions was presented to our trained models and the identified lesions by radiologists were successfully detected.

Finally, with the purpose of having broad validation of the proposed method, preliminary results of this work were presented at the 28th RECPAD 2022 (Portuguese Conference on Pattern Recognition). It has also been submitted an abstract for ECR 2023 (European Congress of Radiology). In addition, it is being written a more complete article proposal to a peer review impact factor journal.

6.1. Future Work

Future work will aim to extent the dataset building a wide ranging anonymized and fully annotated Portuguese digital repository, MRI-based, of breast cancer patients' cases, and try to improve the precision of the developed detection models. With this, we expect that in a near future it will be possible to introduce the developed models in real clinical/radiological workflows.

7. REFERENCES

Abdel-Nasser, M., Solsona, P., & Puig, D. (2020). *Pectoral Muscle Segmentation in Tomosynthesis Images Using Geometry Information and Grey Wolf Optimizer*. https://orcid.org/0000-0002-1074-2441

Abdelrahman, L., al Ghamdi, M., Collado-Mesa, F., & Abdel-Mottaleb, M. (2021). Convolutional neural networks for breast cancer detection in mammography: A survey. *Computers in Biology and Medicine*, *131*. https://doi.org/10.1016/J.COMPBIOMED.2021.104248

Aloysius, N., & Geetha, M. (2018). A review on deep convolutional neural networks. *Proceedings of the 2017 IEEE International Conference on Communication and Signal Processing, ICCSP 2017*, *2018-January*, 588–592. https://doi.org/10.1109/ICCSP.2017.8286426

American Cancer Society. (2022, January 14). *Mammogram Results | Understanding Your Mammogram Report*. https://www.cancer.org/cancer/breast-cancer/screening-tests-and-early-detection/mammograms/understanding-your-mammogram-report.html

Aristokli, N., Polycarpou, I., Themistocleous, S. C., Sophocleous, D., & Mamais, I. (2022). Comparison of the diagnostic performance of Magnetic Resonance Imaging (MRI), ultrasound and mammography for detection of breast cancer based on tumor type, breast density and patient's history: A review. *Radiography*. https://doi.org/10.1016/j.radi.2022.01.006

Arya, P., & Shankar Modani, U. (2019). *Applications of Artificial Neural Network in Image Processing: A Survey*. http://www.ijser.org

Barrios, C. H. (2022). Global challenges in breast cancer detection and treatment. *The Breast*. https://doi.org/10.1016/j.breast.2022.02.003

Bębas, E., Borowska, M., Derlatka, M., Oczeretko, E., Hładuński, M., Szumowski, P., & Mojsak, M. (2021). Machine-learning-based classification of the histological subtype of non-small-cell lung cancer using MRI texture analysis. *Biomedical Signal Processing and Control*, *66*. https://doi.org/10.1016/j.bspc.2021.102446

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., C. Martin, R., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Manifesto for Agile Software Development*. Https://Agilemanifesto.Org/Iso/En/Manifesto.Html. https://agilemanifesto.org/iso/en/manifesto.html

BiologyOnline. (2021, July 21). *Mitosis*. BiologyOnline. https://www.biologyonline.com/dictionary/mitosis

Borkowski, K., Rossi, C., Ciritsis, A., Marcon, M., Hejduk, P., Stieb, S., Boss, A., & Berger, N. (2020). Fully automatic classification of breast MRI background parenchymal enhancement using a transfer learning approach. *Medicine (United States)*, *99*(29). https://doi.org/10.1097/MD.0000000000021243

Brown, A. L., Jeong, J., Wahab, R. A., Zhang, B., & Mahoney, M. C. (2021). Diagnostic accuracy of MRI textural analysis in the classification of breast tumors. *Clinical Imaging*, *77*, 86–91. https://doi.org/10.1016/j.clinimag.2021.02.031

Buda, M., Saha, A., Walsh, R., Ghate, S., Li, N., Święcicki, A., Lo, J. Y., & Mazurowski, M. A. (2020). *Detection of masses and architectural distortions in digital breast tomosynthesis: a publicly available dataset of 5,060 patients and a deep learning model*. http://arxiv.org/abs/2011.07995

Byra, M., Karwat, P., Ryzhankow, I., Komorowski, P., Klimonda, Z., Fura, L., Pawlowska, A., Zolek, N., & Litniewski, J. (2022). *Deep meta-learning for the selection of accurate ultrasound based breast mass classifier*.

Centeno Raimundo, J. N., Pereira Fontes, J. P., & Guevara-Lopez, M. A. (2022). Breast Cancer Detection in MRI using a Faster R-CNN model. *RECPAD 2022*.

Clark, K., Vendt, B., Smith, K., Freymann, J., Kirby, J., Koppel, P., Moore, S., Phillips, S., Maffitt, D., Pringle, M., Tarbox, L., & Prior, F. (2013). The Cancer Imaging Archive (TCIA): Maintaining and Operating a Public Information Repository. *Journal of Digital Imaging 2013 26:6*, *26*(6), 1045–1057. https://doi.org/10.1007/S10278-013-9622-7

Cui, N. (2018). Applying Gradient Descent in Convolutional Neural Networks. *Journal of Physics: Conference Series*, *1004*(1). https://doi.org/10.1088/1742-6596/1004/1/012027

Dong, S., Wang, P., & Abbas, K. (2021). A survey on deep learning and its applications. In *Computer Science Review* (Vol. 40). Elsevier Ireland Ltd. https://doi.org/10.1016/j.cosrev.2021.100379

Durhan, G., Poker, A., Settarzade, E., Karakaya, J., Kösemehmetoğlu, K., Akpınar, M. G., & Demirkazık, F. B. (2021). Magnetic resonance imaging findings of invasive breast cancer in different histological grades and different histopathological types. *Clinical Imaging*, *76*, 98–103. https://doi.org/10.1016/j.clinimag.2021.01.039

Famouri, S., Morra, L., Mangia, L., & Lamberti, F. (2016). *Breast Mass Detection with Faster R-CNN: On the Feasibility of Learning from Noisy Annotations*. https://doi.org/10.1109/ACCESS.2021.3072997

Figueroa, J. D., Gray, E., Pashayan, N., Deandrea, S., Karch, A., Vale, D. B., Elder, K., Procopio, P., van Ravesteyn, N. T., Mutabi, M., Canfell, K., & Nickson, C. (2021). The impact of the Covid-19 pandemic on breast cancer early detection and screening. *Preventive Medicine*, *151*. https://doi.org/10.1016/j.ypmed.2021.106585

Ghojogh, B., Samad, M. N., Mashhadi, S. A., Kapoor, T., Ali, W., Karray, F., & Crowley, M. (2019). *Feature Selection and Feature Extraction in Pattern Analysis: A Literature Review*.

Gonzalez, R. C. (2018). Deep Convolutional Neural Networks. *IEEE SIgnal ProcESSIng MagazInE*. https://doi.org/10.1109/MSP.2018.2842646

Gupta, V., Vasudev, M., Doegar, A., & Sambyal, N. (2021). Breast cancer detection from histopathology images using modified residual neural networks. *Biocybernetics and Biomedical Engineering*, *41*(4), 1272–1287. https://doi.org/10.1016/j.bbe.2021.08.011

Han, S.-H., Kim, K. W., Kim, S., & Youn, Y. C. (2018). Artificial Neural Network: Understanding the Basic Concepts without Mathematics. *Dementia and Neurocognitive Disorders*, *17*(3), 83. https://doi.org/10.12779/dnd.2018.17.3.83

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition*. http://image-net.org/challenges/LSVRC/2015/

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., & Murphy, K. (2017). *Speed/accuracy trade-offs for modern convolutional object detectors*.

Indolia, S., Goswami, A. K., Mishra, S. P., & Asopa, P. (2018). Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach. *Procedia Computer Science*, *132*, 679–688. https://doi.org/10.1016/j.procs.2018.05.069

Isaac Abiodun, O., Jantan, A., Esther Omolara, A., Victoria Dada, K., AbdElatif Mohamed, N., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, *4*, 938. https://doi.org/10.1016/j.heliyon.2018

Jannesari, M., Habibzadeh, M., Aboulkheyr, H., Khosravi, P., Elemento, O., Totonchi, M., & Hajirasouliha, I. (2019). Breast Cancer Histopathological Image Classification: A Deep Learning Approach. *Proceedings - 2018 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2018*, 2405–2412. https://doi.org/10.1109/BIBM.2018.8621307

Jost, A. E., Springenberg, T., Spinello, L., Riedmiller, M., & Burgard, W. (2019). *Multimodal Deep Learning for Robust RGB-D Object Recognition*.

Khaliliboroujeni, S., He, X., Jia, W., & Amirgholipour, S. (2022). End-to-end metastasis detection from breast cancer histopathology whole slide images. *Computerized Medical Imaging and Graphics*, 102136. https://doi.org/10.1016/j.compmedimag.2022.102136

Konde, A., & Thakur, A. (2021). *Fundamentals of Neural Networks*. *9*. https://doi.org/10.22214/ijraset.2021.37362

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). *Feature Pyramid Networks for Object Detection*.

McCann, M. T., Ozolek, J. A., Castro, C. A., Parvin, B., & Kovačević, J. (2015). Automated Histology Analysis: Opportunities for signal processing. In *IEEE Signal Processing Magazine* (Vol. 32, Issue 1, pp. 78–87). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/MSP.2014.2346443

McCulloch, W. S., & Pitts, W. (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity*. *Bulletin of Mathemantical Biology*, *5*, 115–133.

Motlagh, M. H., Jannesari, M., Aboulkheyr, H., Khosravi, P., Elemento, O., Totonchi, M., & Hajirasouliha, I. (2018). *Breast Cancer Histopathological Image Classification: A Deep Learning Approach*. https://doi.org/10.1101/242818

Munir Shah, S., Ahmed Khan, R., Arif, S., & Sajid, U. (2021). *Artificial Intelligence For Breast Cancer Detection: Trends & Directions*.

Murtaza, G., Shuib, L., Abdul Wahab, A. W., Mujtaba, G., Mujtaba, G., Nweke, H. F., Al-garadi, M. A., Zulfiqar, F., Raza, G., & Azmi, N. A. (2020). Deep learning-based breast cancer classification through medical imaging modalities: state of the art and research challenges. *Artificial Intelligence Review*, *53*(3). https://doi.org/10.1007/s10462-019-09716-5

National Cancer Institute. (2021, May 5). *What Is Cancer?* . https://www.cancer.gov/about-cancer/understanding/what-is-cancer

Padilla, R., Passos, W. L., Dias, T. L. B., Netto, S. L., & da Silva, E. A. B. (2021). A comparative analysis of object detection metrics with a companion open-source toolkit. *Electronics (Switzerland)*, *10*(3), 1–28. https://doi.org/10.3390/ELECTRONICS10030279

Rawat, W., & Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, *29*(9), 2352–2449. https://doi.org/10.1162/NECO_A_00990

Ren, S., He, K., Girshick, R., & Sun, J. (2015). *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. https://doi.org/10.48550/arxiv.1506.01497

Rezaei, Z. (2021). A review on image-based approaches for breast cancer detection, segmentation, and classification. *Expert Systems with Applications*, *182*. https://doi.org/10.1016/j.eswa.2021.115204

Saha, A., Harowicz, M. R., Grimm, L. J., Kim, C. E., Ghate, S. v., Walsh, R., & Mazurowski, M. A. (2018). A machine learning approach to radiogenomics of breast cancer: a study of 922 subjects and 529 DCE-MRI features. *British Journal of Cancer 2018 119:4*, *119*(4), 508–516. https://doi.org/10.1038/s41416-018-0185-8

Saha, A., Harowicz, M. R., Grimm, L. J., Weng, J., Cain, E. H., Kim, C. E., Ghate, S. v., Walsh, R., & Mazurowski, M. A. (2021). Dynamic contrast-enhanced magnetic resonance images of breast cancer patients with tumor locations [Data set]. In *Dynamic contrast-enhanced magnetic resonance images of breast cancer patients with tumor locations [Data set]*.

Sung, H., Ferlay, J., Siegel, R. L., Laversanne, M., Soerjomataram, I., Jemal, A., & Bray, F. (2021). Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries. *CA: A Cancer Journal for Clinicians*, *71*(3), 209–249. https://doi.org/10.3322/caac.21660

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2016). *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*.

Szegedy, C., Vanhoucke, V., Ioffe, S., & Shlens, J. (2015). *Rethinking the Inception Architecture for Computer Vision*.

Tariq, M., Iqbal, S., Ayesha, H., Abbas, I., Ahmad, K. T., & Niazi, M. F. K. (2021). Medical image based breast cancer diagnosis: State of the art and future directions. In *Expert Systems with Applications* (Vol. 167). Elsevier Ltd. https://doi.org/10.1016/j.eswa.2020.114095

Ting, F. F., Tan, Y. J., & Sim, K. S. (2019). Convolutional neural network improvement for breast cancer classification. *Expert Systems with Applications*, *120*, 103–115. https://doi.org/10.1016/j.eswa.2018.11.008

Viale, G. (2012). The current state of breast cancer classification. *Annals of Oncology*, *23*(SUPPL. 10). https://doi.org/10.1093/annonc/mds326

Wikipedia. (2021, March 18). *Tomosynthesis*. Https://En.Wikipedia.Org/Wiki/Tomosynthesis. https://en.wikipedia.org/wiki/Tomosynthesis

Wikipedia. (2022a). *Accuracy and precision - Wikipedia*. https://en.wikipedia.org/wiki/Accuracy_and_precision

Wikipedia. (2022b). *Precision and recall - Wikipedia*. https://en.wikipedia.org/wiki/Precision_and_recall

Wikipedia. (2022c). *Type I and type II errors - Wikipedia*. https://en.wikipedia.org/wiki/Type_I_and_type_II_errors

Wikipedia. (2022d, January 6). *Histopathology*. Https://En.Wikipedia.Org/Wiki/Histopathology. https://en.wikipedia.org/wiki/Histopathology

Wikipedia. (2022e, January 13). *Mammography*. Mammography. https://en.wikipedia.org/wiki/Mammography

Wikipedia. (2022f, January 25). *Neoplasm*. Https://En.Wikipedia.Org/Wiki/Neoplasm. https://en.wikipedia.org/wiki/Neoplasm

Wikipedia. (2022g, February 21). *Magnetic Resonance Imaging*. Https://En.Wikipedia.Org/Wiki/Magnetic_resonance_imaging. https://en.wikipedia.org/wiki/Magnetic_resonance_imaging

Wikipedia. (2022h, February 28). *Tomography*. Https://En.Wikipedia.Org/Wiki/Tomography. https://en.wikipedia.org/wiki/Tomography

World Health Organization. (2021, March 26). *Breast cancer*. https://www.who.int/news-room/fact-sheets/detail/breast-cancer

World Health Organization. (2022, February 3). *Cancer*. https://www.who.int/news-room/fact-sheets/detail/cancer

Wu, Y., Wu, J., Dou, Y., Rubert, N., Wang, Y., & Deng, J. (2022). A deep learning fusion model with evidence-based confidence level analysis for differentiation of malignant and benign breast tumors using dynamic contrast enhanced MRI. *Biomedical Signal Processing and Control*, *72*. https://doi.org/10.1016/j.bspc.2021.103319

Xie, J., Hu, K., Guo, Y., Zhu, Q., & Yu, J. (2021). On loss functions and CNNs for improved bioacoustic signal classification. *Ecological Informatics*, *64*. https://doi.org/10.1016/j.ecoinf.2021.101331

Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, *9*(4), 611–629. https://doi.org/10.1007/S13244-018-0639-9

Youk, J. H., Gweon, H. M., & Son, E. J. (2017). Shear-wave elastography in breast ultrasonography: the state of the art. *Ultrasonography*, *36*(4). https://doi.org/10.14366/usg.17024

Yu, H., Yang, L. T., Zhang, Q., Armstrong, D., & Deen, M. J. (2021). Convolutional neural networks for medical image analysis: State-of-the-art, comparisons, improvement and perspectives. *Neurocomputing*, *444*, 92–110. https://doi.org/10.1016/J.NEUCOM.2020.04.157

Zhang, Y., Wang, X., Blanton, H., Liang, G., Xing, X., & Jacobs, N. (2020). *2D Convolutional Neural Networks for 3D Digital Breast Tomosynthesis Classification*.

Zou, Z., Shi, Z., Guo, Y., Ye, J., & Member, S. (2019). *Object Detection in 20 Years: A Survey*.

8. APPENDIX

Code 5: Prototype: configuration file prototype (pre_processing_config.py.proto)

```
image = <image_configs.proto>
patterns = <pattern_configs.proto>
paths = <path_configs.proto>
dataset_split = <dataset_split.proto>
template = <annotation_template.proto>
```

Code 6: Prototype: image configuration prototype (pre_processing_config_image.proto)

```
{
    "known_types": array<string>,
    "extraction_extension": string,
    "resize": {
        "width": integer,
        "height": integer
    },
    "mandatory_scan_options": array<string>,
}
```

Code 7: Prototype: patterns configuration prototype (pre_processing_config_patterns.proto)

```
{
    "raw": {
        "patient_id_pattern": string,
        "slice_number_pattern": string,
        "series_id_pattern": string,
    },
    "preprocessed": {
        "patient_id_pattern": string,
        "slice_number_pattern": string,
        "series_id_pattern": string,
    },
    "series_to_exclude": array <string>
}
```

Code 8: Prototype: paths configuration prototype (pre_processing_config_paths.proto)

```
{
    "dataset_base_working_directory": string,
    "raw_dir": string,
    "preprocessed_dir": string,
    "set_dirs": {
        "train": string,
  "validation": string,
  "test": string,
    },
    "split_patients_ids_file": string,
    "cross_validations_split_files": {
        "subpath": string,
        "base_name": string,
        "extension": string,
    },
    "annotations_file": string
}
```

Code 9: Prototype: annotation configuration prototype (pre_processing_config_annotation.proto)

```
{
    "new_filename": string,
    "annotations": {
        "meta": {
            "annotation": {
                "folder": string,
                "filename": string,
                "path": string,
                "size": {
                    "width": integer,
                    "height": integer
                }
            }
        },
        "object": {
            "name": string,
            "bndbox": {
                "xmin": integer,
                "ymin": integer,
                "xmax": integer,
                "ymax": integer,
            }
        }
    }
}
```

Code 10: Prototype: Inference set configuration (inference_config_set.proto)

```
{
    "validation_set_path": string,
    "result_path": string,
    "model_path": string,
    "labels_path": string
}
```

Code 11: Prototype: Inference configuration (inference_configuration.py.proto)

```
inference = {
  "min_score_thresh": float,
  "sets": array<inference_set_configuration.proto>
}

patterns = {
  "patient_id_pattern": string,
  "slice_number_pattern": string,
  "series_id_pattern": string,
}
```

Code 12: Prototype: Image inference result (inference_result.json.proto)

```
{
    "patient_id": string,
    "slice_number": integer,
    "series_id": string,
    "image_path": string,
    "inference_path":string,
    "metrics": array<metric.json.proto>
}
```

Code 13: Prototype: Inference metric (inference_result_metric.proto)

```
{
    "iou": float,
    "box": {
        "ymin": integer,
        "xmin": integer,
        "ymax": integer,
        "xmax": integer
    },
    "truth_box": {
        "xmax": integer,
        "xmin": integer,
        "ymax": integer,
        "ymin": integer
    },
    "score": float
}
```

MAXs	FOLD 1 Acc 96.74% FN 3		FOLD 2 Acc 95.65% FN 4		FOLD 3 Acc 97.83% FN 2		FOLD 4 Acc 93.48% FN 6		FOLD 5 Acc 96.74% FN 3		FOLD 6 Acc 94.57% FN 5		FOLD 7 Acc 93.48% FN 6		FOLD 8 Acc 91.30% FN 8		FOLD 9 Acc 94.57% FN 5		FOLD 10 Acc 90.22% FN 9	
	IoU	Score	IoU	Score	IoU	Score	IoU	Score	IoU	Score	IoU	Score	IoU	Score	IoU	Score	IoU	Score	IoU	Score
AVG	69.02%	87.15%	63.69%	84.07%	66.47%	86.18%	65.42%	85.58%	64.44%	86.39%	66.35%	85.23%	65.87%	83.71%	62.82%	79.99%	66.51%	82.33%	61.85%	81.49%
@IoU>0	71.34%	90.08%	66.59%	87.89%	67.95%	88.09%	69.99%	91.56%	66.61%	89.30%	70.17%	90.12%	70.47%	89.55%	68.80%	87.61%	70.34%	87.06%	68.56%	90.32%
1	0.6592	0.9994	0.9444	0.9999	0.9123	1.0000	0.8551	0.8695	0.5045	0.7965	0.8750	0.9995	0.8963	1.0000	0.4157	0.9617	0.7614	0.8707	0.7304	0.9986
2	0.9310	1.0000	0.9058	0.9996	0.7745	1.0000	0.6118	1.0000	0.7554	0.9811	0.8079	0.9990	0.5554	0.9283	0.7246	0.7800	0.9271	0.9997	0.6684	0.5573
3	0.9837	1.0000	0.7842	0.9040	0.6280	0.8857	0.5895	0.6493	0.4096	0.8275	0.7290	1.0000	0.6717	0.8059	0.8136	0.9999	0.7199	1.0000	0.0000	0.0000
4	0.3093	0.9532	0.8400	0.5755	0.8530	0.9583	0.7637	1.0000	0.7684	0.7899	0.8246	1.0000	0.8724	1.0000	0.7732	1.0000	0.9187	0.9991	0.9044	0.5681
5	0.5810	0.9994	0.7556	0.5970	0.7587	1.0000	0.8514	1.0000	0.7321	0.9210	0.6785	0.8087	0.6920	1.0000	0.1270	0.9036	0.8760	0.7670	0.7743	0.8425
6	0.7600	0.9505	0.6988	0.9993	0.8478	0.9998	0.0000	0.0000	0.8522	0.9998	0.8474	0.9967	0.8909	0.9990	0.7398	1.0000	0.9400	1.0000	0.7245	0.9947
7	0.8432	0.9987	0.5749	0.9814	0.8580	0.7926	0.0000	0.0000	0.8480	1.0000	0.8722	0.9992	0.5018	0.6090	0.7135	0.9514	0.8437	0.9336	0.8981	0.9894
8	0.7688	0.9985	0.8507	0.8317	0.7972	0.9877	0.6697	0.8644	0.8154	0.9998	0.7277	0.9976	0.7897	0.8513	0.3323	0.9470	0.8401	1.0000	0.9677	1.0000
9	0.6966	0.7287	0.5327	1.0000	0.5829	0.9776	0.8462	1.0000	0.8478	0.9849	0.9532	1.0000	0.8701	0.9874	0.7193	0.9765	0.7239	0.7105	0.5907	0.9134
10	0.4553	0.5138	0.8889	1.0000	0.7192	1.0000	0.9003	1.0000	0.1241	0.6398	0.6826	0.8617	0.9031	0.9814	0.7526	0.9973	0.9778	0.9995	0.8830	0.9585
11	0.5476	0.9996	0.7700	1.0000	0.8635	0.9999	0.6914	0.9989	0.8008	0.7102	0.6516	0.8002	0.6045	0.5846	0.6667	0.9959	0.7570	0.9997	0.6149	0.8429
12	0.9209	1.0000	0.3541	0.9998	0.4561	0.9952	0.7534	0.6840	0.8940	1.0000	0.8539	1.0000	0.8478	1.0000	0.6709	0.5789	0.6709	0.9997	0.0000	0.0000
13	0.7350	1.0000	0.7982	0.8401	0.6144	1.0000	0.3986	0.9858	0.8786	0.8972	0.6693	0.9989	0.6375	0.8292	0.5587	0.7023	0.5333	0.5187	0.7676	0.7931
14	0.7913	0.9926	0.2894	0.6703	0.2214	0.6994	0.7408	0.9811	0.2212	0.7907	0.6756	0.7969	0.7133	0.7694	0.0000	0.0000	0.9211	0.5702	0.2802	0.9163
15	0.9095	0.6821	0.4986	0.8676	0.8194	0.9994	0.6968	0.9915	0.8013	1.0000	0.3918	0.7506	0.8660	0.9999	0.7895	0.7095	0.4320	0.6908	0.8696	0.9965
16	0.8675	1.0000	0.6536	0.7299	0.7808	0.9999	0.9356	1.0000	0.6919	0.7240	0.9012	0.9994	0.5833	0.9866	0.9305	0.9382	0.9144	1.0000	0.8488	0.9968
17	0.8001	0.9719	0.4667	0.8171	0.8639	1.0000	0.8819	1.0000	0.7351	0.9998	0.9134	1.0000	0.8299	1.0000	0.8758	0.7326	0.7254	0.5924	0.3647	0.9786
18	0.5132	0.9998	0.2861	0.7087	0.6341	0.9022	0.8947	1.0000	0.0000	0.0000	0.8480	1.0000	0.8666	0.9961	0.7064	0.9823	0.8078	1.0000	0.7388	0.7117
19	0.5571	0.5498	0.8804	1.0000	0.8182	0.9919	0.7341	0.9982	0.5418	0.8780	0.4711	0.6886	0.0000	0.0000	0.8678	0.9881	0.5561	0.5068	0.0641	1.0000
20	0.1739	0.5673	0.7394	0.9782	0.5957	0.7019	0.8804	0.9994	0.7112	1.0000	0.4974	0.8373	0.7265	1.0000	0.8402	0.9998	0.6124	0.6865	0.8376	1.0000
21	0.8417	0.9907	0.2383	0.9953	0.3400	0.5940	0.9330	0.9999	0.7778	0.9809	0.7644	0.9875	0.6015	1.0000	0.0000	0.0000	0.9082	0.9998	0.6955	1.0000
22	0.9615	0.9997	0.8028	0.9977	0.6420	1.0000	0.4211	0.6037	0.7777	0.8558	0.6835	0.5819	0.7824	0.8583	0.7455	0.9999	0.7038	1.0000	0.0000	0.0000
23	0.9332	1.0000	0.9121	1.0000	0.8000	1.0000	0.7785	1.0000	0.7381	0.9946	0.6923	0.5195	0.7946	0.9193	0.7004	0.9894	0.6723	0.8802	0.8791	0.9999
24	0.8158	0.8294	0.9556	1.0000	0.5872	0.8627	0.0000	0.0000	0.6524	0.7858	0.7940	0.9170	0.7435	0.7199	0.5672	0.9534	0.8546	0.9996	0.3333	0.9379
25	0.6337	0.9999	0.8468	0.9968	0.8851	0.9191	0.8875	0.9998	0.8472	0.6208	0.8456	1.0000	0.8147	0.9996	0.7323	1.0000	0.8145	0.7318	0.9583	0.9972
26	0.4483	1.0000	0.7692	1.0000	0.9149	0.7903	0.5827	0.8268	0.8644	0.9999	0.7196	0.6853	0.7975	0.9883	0.8145	1.0000	0.7450	0.9999	0.5455	0.9084
27	0.7586	0.9855	0.1879	0.9740	0.7989	0.9982	0.8961	1.0000	0.8910	0.9996	0.9100	0.9973	0.9358	1.0000	0.7079	0.5688	0.7005	0.6923	0.9390	0.7668
28	0.9114	1.0000	0.5820	0.8536	0.7913	1.0000	0.9524	0.9638	0.8507	1.0000	0.8214	0.9996	0.8675	1.0000	0.6592	0.6324	0.8588	0.9687	0.6782	0.9870
29	0.8889	1.0000	0.8928	1.0000	0.8876	0.9999	0.7505	0.7346	0.3407	0.9376	0.6694	0.8912	0.3117	0.9592	0.7301	0.9968	0.6888	1.0000	0.9032	1.0000
30	0.8298	0.9495	0.0000	0.0000	0.7027	0.6176	0.5556	0.9999	0.8865	0.9825	0.8340	0.6178	0.4748	0.9852	0.8886	1.0000	0.2169	0.7576	0.8603	0.9986
31	0.8964	0.7251	0.8110	0.9282	0.6002	1.0000	0.3122	0.9907	0.8178	0.9985	0.8467	1.0000	0.5530	0.6564	0.5922	0.9999	0.7768	0.6379	0.9634	1.0000
32	0.5941	0.8846	0.8101	0.8927	0.7705	0.7287	0.5759	0.9180	0.4175	0.7929	0.8094	0.9997	0.9175	1.0000	0.0000	0.0000	0.4451	0.7580	0.8903	0.9983
33	0.8935	0.9809	0.8731	1.0000	0.7176	0.9783	0.7627	0.9871	0.6384	0.7359	0.8991	1.0000	0.7770	0.9999	0.6501	0.5717	0.0000	0.0000	0.7945	0.7531
34	0.8546	0.9844	0.8647	0.7521	0.8410	0.9535	0.3422	0.5896	0.6667	0.9004	0.8657	0.9994	0.4124	0.5449	0.7175	0.5494	0.2347	0.9095	0.6857	0.9909
35	0.4506	0.9012	0.5904	0.8013	0.7819	0.5505	0.9706	1.0000	0.6573	0.6477	0.9049	1.0000	0.8809	0.9997	0.9630	0.9864	0.8663	1.0000	0.0000	0.0000
36	0.8678	0.9990	0.8521	0.9923	0.5400	0.9761	0.8947	0.9988	0.8023	1.0000	0.7968	0.9993	0.7624	0.9244	0.7181	1.0000	0.9466	1.0000	0.8160	0.9996
37	0.7979	0.9995	0.0893	0.5681	0.7875	0.6413	1.0000	0.9966	0.8681	1.0000	0.7959	0.9789	0.8611	0.9965	0.7754	0.6712	0.6159	0.9965	0.6328	0.9772
38	0.8292	0.8941	0.7097	0.9703	0.7209	0.9515	0.8051	1.0000	0.5581	0.5862	0.6390	0.5026	0.3771	0.7295	0.6912	1.0000	0.7617	0.9998	0.7060	0.6410
39	0.6478	0.5985	0.6875	0.9993	0.5874	1.0000	0.5316	0.9746	0.5469	0.9086	0.9199	0.9999	0.0000	0.0000	0.8750	0.9999	0.7125	0.5548	0.5423	0.9666
40	0.7610	0.9999	0.6486	0.6945	0.8675	1.0000	0.7296	0.9898	0.4321	0.6341	0.6667	0.7957	0.6775	1.0000	0.7586	0.9981	0.5362	0.6353	0.2731	0.6715
41	0.8393	0.6434	0.8835	1.0000	0.7094	0.9994	0.6595	0.7368	0.7734	1.0000	0.6964	0.9859	0.0493	0.9986	0.9381	0.9991	0.3467	0.9984	0.4560	0.8748
42	0.7771	0.9971	0.3596	0.9927	0.8570	0.9582	0.5287	0.9998	0.6688	0.9122	0.6152	0.9993	0.4029	0.7196	0.3637	0.9668	0.0000	0.0000	0.6405	1.0000
43	0.8159	1.0000	0.6111	0.6013	0.8933	0.9211	0.6810	1.0000	0.8830	1.0000	0.4769	0.7211	0.4504	0.9537	0.5667	1.0000	0.9322	1.0000	0.5185	0.8843
44	0.8947	0.9999	0.7152	0.5152	0.4937	0.6505	0.2459	0.5800	0.7290	0.9360	0.7309	0.9697	0.4622	0.9957	0.7813	0.7773	0.8963	0.9974	0.6614	0.9443
45	0.7133	0.9983	0.7269	0.9353	0.7109	0.8406	0.2089	0.8975	0.6863	0.9966	0.8455	0.6301	0.7732	0.9998	0.9628	1.0000	0.8148	1.0000	0.8000	0.9999
46	0.8128	0.9999	0.7321	0.9653	0.3478	0.7911	0.2573	0.6701	0.9196	1.0000	0.6034	1.0000	0.8947	0.9928	0.8730	0.9868	0.7969	0.9796	0.6819	0.9794
47	0.6525	0.9945	0.4630	0.9397	0.8630	0.9993	0.6893	0.9999	0.6803	1.0000	0.7059	0.7906	0.6523	1.0000	0.6771	0.5540	0.2526	0.9994	0.8269	0.5922
48	0.6788	0.9984	0.7197	0.9867	0.8732	0.8919	0.7428	0.9998	0.4957	1.0000	0.0000	0.0000	0.9630	1.0000	0.7273	0.9995	0.4985	0.7780	0.5687	0.9656
49	0.7267	1.0000	0.7426	1.0000	0.6562	0.6377	0.8961	1.0000	0.8589	0.6199	0.7819	0.9996	0.0000	0.0000	0.3617	0.6862	0.6941	1.0000	0.9011	0.9954
50	0.9737	0.9867	0.7647	0.9746	0.1795	0.9503	0.6130	0.9587	0.4800	0.7737	0.6571	0.8550	0.0000	0.0000	0.0000	0.0000	0.7483	1.0000	0.6553	0.8177
51	0.0000	0.0000	0.0000	0.0000	0.0920	0.5076	0.7203	0.8784	0.8462	1.0000	0.6270	0.7100	0.9223	1.0000	0.0000	0.0000	0.8694	0.9513	0.4931	1.0000
52	0.7500	0.9987	0.7391	0.9190	0.7872	0.7626	0.7826	0.9960	0.7796	0.6459	0.5758	0.9999	0.5418	0.8925	0.5888	0.8489	0.8860	0.9406	0.8184	0.9946
53	0.4819	0.9651	0.2371	0.6910	0.6084	0.8981	0.5921	0.8589	0.7098	1.0000	0.8586	0.8737	0.7078	0.9987	0.6818	0.9935	0.7330	0.9844	0.5490	0.8608
54	0.5364	0.7876	0.6567	0.9768	0.7771	0.9600	0.6622	0.9867	0.4917	0.8256	0.2728	0.9787	0.7970	0.6450	0.8032	1.0000	0.7115	0.8987	0.7711	0.9318
55	0.0176	0.8015	0.5385	0.9978	0.7069	1.0000	0.5680	0.9130	0.7793	0.6805	0.0000	0.0000	0.8815	1.0000	0.8847	0.9872	0.7453	0.8615	0.6867	0.9690
56	0.6188	0.9611	0.4071	0.9826	0.8889	0.9784	0.8882	0.9208	0.5346	1.0000	0.5037	1.0000	0.8974	1.0000	0.8367	0.9754	0.4109	0.7425	0.6997	0.9761
57	0.6117	0.5550	0.7460	0.9811	0.8006	0.7365	0.8233	0.9842	0.7931	0.9514	0.1132	0.8885	0.7849	0.9861	0.8972	1.0000	0.6847	0.9198	0.0000	0.0000
58	0.6626	0.9995	0.9060	0.9989	0.8967	0.9991	0.8127	0.9448	0.4286	0.9993	0.7047	0.9972	0.8120	0.9802	0.6325	0.6913	0.8929	1.0000	0.4418	0.5657
59	0.8532	0.9777	0.2100	0.7429	0.5294	0.6092	0.5902	0.9969	0.0000	0.0000	0.2397	0.6396	0.5684	0.7192	0.6680	0.9943	0.8067	0.9991	0.7037	0.9322
60	0.8746	0.9990	0.4505	0.5490	0.9436	1.0000	0.7352	0.7422	0.5000	0.6553	0.6759	0.9992	0.8947	0.5172	0.8159	0.9951	0.6777	0.9999	0.5310	0.8560
61	0.9143	0.9994	0.9234	1.0000	0.6568	1.0000	0.7242	0.5084	0.8760	0.9999	0.8229	1.0000	0.6522	0.9608	0.5228	0.7060	0.0000	0.0000	0.8631	0.9956
62	0.1900	0.6862	0.7153	0.6928	0.6207	0.5077	0.2566	0.5090	0.5287	0.9994	0.8339	1.0000	0.4198	0.7009	0.7105	0.5070	0.6786	0.5721	0.7813	0.9998
63	0.8367	0.5190	0.7500	1.0000	0.5578	0.7270	0.7876	0.9973	0.9608	1.0000	0.3417	0.9766	0.8746	0.9981	0.9008	1.0000	0.4636	0.8978	0.8000	0.8420
64	0.9272	0.9939	0.6947	0.5679	0.5696	0.8554	0.3810	0.7693	0.7792	0.9975	0.2427	0.5655	0.8720	0.8668	0.6809	0.5052	0.1804	0.9999	0.8247	1.0000
65	1.0000	0.9595	0.8121	0.9955	0.9272	0.9993	0.8225	0.9995	0.9444	0.9989	0.8701	0.9983	0.9134	1.0000	0.6937	0.6999	0.8071	0.9311	0.8267	1.0000
66	0.7121	0.9520	0.4240	0.9099	0.6563	0.9974	0.6701	0.9615	0.8411	0.9959	0.5785	0.8514	0.7192	0.6955	0.0000	0.0000	0.0000	0.0000	0.3667	0.9984
67	0.0000	0.0000	0.8786	0.9999	0.6394	0.9362	0.9310	0.8771	0.4420	0.9034	0.7559	1.0000	0.0000	0.0000	0.9600	0.9997	0.7413	0.6298	0.6259	0.6765
68	0.9600	0.9995	0.8533	0.7805	0.4802	0.5280	0.2865	0.9085	0.1639	0.9917	0.0000	0.0000	0.6297	0.9949	0.5707	0.9748	0.3083	0.5124	0.0000	0.0000
69	0.6869	0.6611	0.3316	0.9935	0.8567	0.9987	0.3690	0.6406	0.7932	0.5119	0.1034	0.7638	0.9486	0.9992	0.5104	0.7247	0.7773	0.5933	0.5535	0.9999
70	0.6635	0.9957	0.6303	0.9996	0.6217	0.5684	0.8041	0.9672	0.1855	0.5781	0.0000	0.0000	0.5536	0.7264	0.4706	0.9970	0.9657	1.0000	0.4492	0.9999
71	0.7407	0.8220	0.8645	0.8538	0.7369	0.9529	0.8278	1.0000	0.5838	1.0000	0.7867	0.9997	0.6839	0.9998	0.4909	0.5816	0.3403	0.6054	0.9033	1.0000
72	0.7773	0.9958	0.6738	0.8708	0.4800	0.7498	0.8365	0.9911	0.8654	0.9951	0.8205	0.9357	0.6234	0.9997	0.7384	0.9081	0.8317	0.9678	0.0000	0.0000
73	0.7586	0.9999	0.5944	0.9998	0.7989	0.9490	0.6122	0.9947	0.7171	0.9999	0.6574	0.9997	0.7689	0.9926	0.2976	0.9931	0.8015	0.9993	0.8152	0.9971
74	0.8846	0.6034	0.7143	0.8535	0.8519	0.7405	0.8293	1.0000	0.2918	0.7640	0.5133	0.9276	0.6047	0.6817	0.4164	0.7267	0.5953	0.9239	0.8032	0.9997
75	0.8105	0.9984	0.6632	0.5628	0.2600	0.8361	0.5153	0.9960	0.3644	0.9334	0.6667	0.9555	0.7522	0.6247	0.0597	0.6926	0.6937	0.5621	0.7200	0.6430
76	0.6442	1.0000	0.9615	0.7484	0.0000	0.0000	0.2502	0.9374	0.0053	0.9896	0.6759	0.9900	0.8684	0.5543	0.9178	0.8114	0.8124	1.0000	0.9032	0.9990
77	0.2656	0.5249	0.0000	0.0000	0.8286	1.0000	0.7333	0.7204	0.5418	0.9998	0.8063	0.5790	0.8144	0.9941	0.8537	1.0000	0.8441	0.9609	0.8974	0.6259
78	0.8926	1.0000	0.0000	0.0000	0.6507	0.9968	0.9286	0.9978	0.8054	0.9965	0.8606	1.0000	0.4548	0.9993	0.9036	0.9041	0.4048	0.9842	0.3157	0.5770
79	0.7438	0.9995	0.7647	0.9999	0.0000	0.0000	0.4792	0.8822	0.0000	0.0000	0.6876	0.9970	0.5285	0.9003	0.7257	0.9910	0.8786	1.0000	0.0000	0.0000
80	0.8679	0.9145	0.7143	0.9934	0.7071	1.0000	0.6703	1.0000	0.3107	0.9996	0.6992	0.8846	0.6168	0.8655	0.5287	0.9909	0.8218	0.9996	0.8082	0.9745
81	0.7508	0.5161	0.3200	0.5808	0.7712	0.9937	0.8984	1.0000	0.3970	0.5373	0.6400	0.9834	0.4523	0.6394	0.8065	0.9840	0.7860	0.5168	0.7032	0.9752
82	0.6261	1.0000	0.2071	1.0000	0.6460	0.9999	0.8714	0.9969	0.6259	0.9999	0.8200	0.9869	0.7542	0.9760	0.5823	0.6283	0.7843	0.9852	0.7034	0.9445
83	0.8330	0.8946	0.5928	0.6061	0.2964	0.7509	0.9032	1.0000	0.8654	0.9996	0.2207	1.0000	0.3326	0.6285	0.7389	0.5535	0.3721	0.8010	0.5867	0.9008
84	0.6135	0.6322	0.7536	0.9751	0.5350	0.9056	0.3917	0.9941	0.7907	0.9995	0.5647	0.8190	0.8500	0.6321	0.6316	0.8072	0.8033	0.9354	0.4024	0.7022
85	0.5176	0.8385	0.8491	0.8782	0.8824	0.7614	0.8902	0.9181	0.8421	0.9747	0.5169	0.7049	0.4848	0.9998	0.1765	0.8517	0.6707	0.9999	0.7644	0.9431
86	0.7619	0.7355	0.8000	0.9892	0.3833	0.9994	0.0000	0.0000	0.7224	0.9777	0.8391	0.5408	0.7516	0.9985	0.3047	0.9997	0.9098	0.9523	0.5139	0.8589
87	0.6980	0.8810	0.3559	0.5383	0.6842	0.8164	0.9545	0.9894	0.7037	0.9931	0.8235	0.9229	0.7166	0.9587	0.0000	0.0000	0.4452	0.9555	0.2857	0.8262
88	0.4778	0.8724	0.7844	1.0000	0.8018	0.9985	0.7778	0.9877	0.8333	0.9999	0.9348	0.9990	0.8235	0.8924	0.8282	0.8841	0.6944	0.7773	0.0000	0.0000
89	0.5091	0.9261	0.7648	1.0000	0.3170	0.9994	0.0000	0.0000	0.5801	0.5128	0.8750	0.9980	0.7587	0.9207	0.9375	0.9945	0.8396	0.9997	0.5726	0.8335
90	0.5106	0.9642	0.4632	0.6033	0.7000	0.9939	0.9009	0.8252	0.5758	0.7577	0.8096	0.9991	0.6295	0.8747	0.6024	0.9267	0.0000	0.0000	0.7886	0.9579
91	0.2101	0.9986	0.8229	1.0000	0.1964	0.5122	0.0000	0.0000	0.5013	0.9964	0.8421	0.9844	0.0000	0.0000	0.0000	0.0000	0.4160	0.6802	0.6771	0.9089
92	0.0000	0.0000	0.6723	0.6821	0.4598	0.6627	0.8125	0.7078	0.8421	0.7782	0.0000	0.0000	0.6306	0.9291	0.8688	0.7327	0.6085	0.5811	0.6178	0.9972

Table 12: Cross-validation results: Averages of IoU and Score maximums for each fold (@IoU>0 is the calculation of the average counting only detected tumors).

AVGs	FOLD 1 Acc 96.74% FN 3		FOLD 2 Acc 95.65% FN 4		FOLD 3 Acc 97.83% FN 2		FOLD 4 Acc 93.48% FN 6		FOLD 5 Acc 96.74% FN 3		FOLD 6 Acc 94.57% FN 5		FOLD 7 Acc 93.48% FN 6		FOLD 8 Acc 91.30% FN 8		FOLD 9 Acc 94.57% FN 5		FOLD 10 Acc 90.22% FN 9	
	IoU	Score	IoU	Score	IoU	Score	IoU	Score	IoU	Score	IoU	Score	IoU	Score	IoU	Score	IoU	Score	IoU	Score
AVG	31.22%	57.65%	28.00%	52.20%	27.89%	86.18%	29.31%	52.56%	27.67%	55.44%	28.08%	54.18%	28.14%	50.60%	27.93%	54.76%	27.96%	51.24%	23.29%	44.95%
@IoU>0	32.27%	59.16%	29.27%	54.57%	28.51%	88.09%	31.36%	56.06%	28.60%	57.26%	29.69%	57.25%	30.11%	54.06%	30.59%	59.61%	29.57%	54.05%	25.81%	49.34%
1	0.3622	0.6075	0.4892	0.6180	0.6202	1.0000	0.4028	0.6869	0.2215	0.6259	0.2105	0.2862	0.5016	0.7031	0.2183	0.7625	0.2037	0.4176	0.4081	0.6956
2	0.4747	0.8180	0.4979	0.7002	0.4643	1.0000	0.4251	0.8429	0.3704	0.8171	0.3468	0.5885	0.3833	0.9310	0.0595	0.4083	0.4691	0.6814	0.1126	0.3134
3	0.5162	0.7487	0.2094	0.4612	0.2293	0.8857	0.1861	0.3346	0.0737	0.2114	0.4200	0.7541	0.3721	0.6536	0.4978	0.7333	0.4242	0.7481	0.0000	0.2770
4	0.0370	0.2778	0.2855	0.5481	0.6447	0.9583	0.4458	0.7194	0.0772	0.3520	0.3678	0.7938	0.5035	0.7128	0.4385	0.7274	0.4570	0.6981	0.1706	0.8392
5	0.0610	0.1124	0.2445	0.5436	0.4212	1.0000	0.5202	0.9470	0.0520	0.4222	0.0041	0.1246	0.3983	0.7783	0.0079	0.0786	0.2444	0.4245	0.2528	0.5822
6	0.2885	0.4356	0.3651	0.6577	0.4345	0.9998	0.0000	0.0477	0.4962	0.8758	0.3779	0.5520	0.5982	0.8139	0.4291	0.7900	0.5174	0.6556	0.2162	0.5698
7	0.2579	0.4416	0.1006	0.2901	0.2447	0.7926	0.0000	0.0000	0.4203	0.6444	0.4605	0.8311	0.0170	0.0741	0.2047	0.5333	0.2899	0.4246	0.2094	0.7122
8	0.2345	0.3902	0.4556	0.7424	0.1526	0.9877	0.2445	0.4047	0.2610	0.4796	0.5359	0.8907	0.1985	0.4427	0.0925	0.3147	0.5367	0.7483	0.2183	0.2489
9	0.2397	0.6790	0.2473	0.6063	0.1942	0.9776	0.3810	0.5507	0.2960	0.4740	0.5392	0.7540	0.3517	0.5303	0.2926	0.4818	0.0649	0.2804	0.2017	0.4267
10	0.0253	0.0702	0.5967	0.8012	0.3709	1.0000	0.3091	0.4812	0.0021	0.0382	0.2600	0.4617	0.4932	0.7205	0.3697	0.5899	0.5930	0.8249	0.2324	0.5970
11	0.3264	0.8396	0.5328	0.8499	0.6935	0.9999	0.3790	0.7071	0.1482	0.2787	0.1413	0.2542	0.0250	0.0663	0.1525	0.3843	0.3304	0.5520	0.0522	0.1290
12	0.6736	0.9127	0.1234	0.5742	0.2134	0.9952	0.0898	0.2841	0.5173	0.7176	0.4626	0.7913	0.4734	0.8012	0.0317	0.2142	0.3522	0.8105	0.0000	0.0000
13	0.5237	0.8069	0.4774	0.8318	0.1082	1.0000	0.0879	0.4402	0.6215	0.8346	0.5416	0.9811	0.1129	0.3081	0.1543	0.3897	0.1412	0.5033	0.2566	0.4888
14	0.2136	0.5263	0.0306	0.3013	0.0942	0.6994	0.4228	0.6938	0.1196	0.6030	0.5198	0.9026	0.1003	0.3040	0.0000	0.0156	0.7196	0.9376	0.0576	0.2617
15	0.3491	0.4661	0.0138	0.0241	0.4894	0.9994	0.4851	0.8280	0.4975	0.8446	0.0735	0.1950	0.4557	0.6551	0.2678	0.5917	0.0508	0.3665	0.1541	0.2109
16	0.4690	0.7165	0.1378	0.5468	0.2495	0.9999	0.2858	0.6092	0.3642	0.7790	0.3804	0.5487	0.2867	0.5691	0.4922	0.9004	0.3818	0.4839	0.3490	0.4926
17	0.1627	0.4124	0.1183	0.2572	0.2959	1.0000	0.3724	0.6253	0.3227	0.6087	0.5630	0.8448	0.5826	0.9529	0.3938	0.6411	0.3580	0.5834	0.0606	0.1610
18	0.3429	0.7853	0.0086	0.0313	0.1445	0.9022	0.6577	0.9128	0.0000	0.0000	0.4193	0.5629	0.4907	0.7673	0.2573	0.7075	0.6273	0.8878	0.2225	0.4365
19	0.1015	0.6584	0.5183	0.6586	0.3766	0.9919	0.5256	0.9507	0.3958	0.8545	0.1921	0.5089	0.0000	0.0000	0.5831	0.7068	0.2990	0.8089	0.0063	0.2176
20	0.0304	0.8161	0.3963	0.7592	0.2590	0.7019	0.3796	0.5768	0.4195	0.7444	0.0029	0.0328	0.4127	0.7379	0.4579	0.6169	0.2316	0.5948	0.5410	0.8358
21	0.2697	0.3991	0.0555	0.4295	0.0451	0.5940	0.1491	0.3230	0.3091	0.7023	0.4701	0.7628	0.4786	0.9170	0.0000	0.2344	0.1902	0.5057	0.3639	0.6626
22	0.4268	0.5560	0.4791	0.9460	0.5318	1.0000	0.0468	0.0671	0.0951	0.2460	0.3693	0.9006	0.1316	0.1867	0.2818	0.4558	0.4846	0.9659	0.0000	0.0000
23	0.7378	0.9275	0.4907	0.8239	0.5089	1.0000	0.5284	0.7824	0.0919	0.3662	0.0784	0.2874	0.1907	0.4108	0.3340	0.6319	0.1549	0.3836	0.5925	0.8476
24	0.2962	0.4330	0.5878	0.8029	0.0786	0.8627	0.0000	0.0000	0.2102	0.5903	0.4079	0.7317	0.1291	0.4155	0.2380	0.4942	0.1890	0.6340	0.0624	0.3938
25	0.1737	0.6422	0.5340	0.7122	0.6481	0.9191	0.5959	0.8762	0.2785	0.5606	0.3202	0.4439	0.2632	0.4804	0.5361	0.8753	0.3499	0.5431	0.2827	0.6171
26	0.3525	0.9914	0.3880	0.7689	0.2863	0.7903	0.0551	0.2206	0.4817	0.7104	0.1599	0.6187	0.2165	0.3536	0.1977	0.3933	0.2120	0.3506	0.1751	0.3215
27	0.2300	0.4423	0.0868	0.5385	0.4451	0.9982	0.5509	0.7169	0.6276	0.7936	0.5440	0.7837	0.5633	0.7741	0.1709	0.4483	0.1086	0.2162	0.4291	0.6244
28	0.5658	0.8380	0.1527	0.3991	0.4886	1.0000	0.3743	0.4220	0.4437	0.7776	0.2484	0.3833	0.3717	0.4921	0.3228	0.5875	0.1910	0.4722	0.2572	0.5428
29	0.4991	0.7595	0.6526	0.9091	0.5234	0.9999	0.5197	0.9391	0.0538	0.1937	0.3522	0.7179	0.1158	0.6038	0.2634	0.5871	0.0576	0.1931	0.7389	0.9382
30	0.1134	0.1814	0.0000	0.0000	0.1194	0.6176	0.0563	0.1354	0.5529	0.8858	0.1667	0.3629	0.0845	0.4207	0.4685	0.7604	0.0049	0.0162	0.1674	0.2405
31	0.3232	0.7297	0.1811	0.3850	0.2153	1.0000	0.0122	0.0397	0.5923	0.9574	0.4265	0.7341	0.1456	0.4186	0.1872	0.8699	0.4776	0.7746	0.6950	0.9718
32	0.0682	0.1070	0.3313	0.5048	0.2388	0.7287	0.3091	0.7265	0.0248	0.0669	0.4558	0.7121	0.5432	0.8913	0.0000	0.0195	0.0082	0.0140	0.2002	0.4277
33	0.3722	0.6660	0.6482	0.9224	0.1377	0.9783	0.3619	0.6548	0.2404	0.4046	0.6203	0.8893	0.5330	0.8596	0.2664	0.5929	0.0000	0.0000	0.2821	0.4936
34	0.3560	0.5454	0.4230	0.7638	0.2921	0.9535	0.0037	0.0568	0.1345	0.2174	0.5973	0.8369	0.0037	0.0575	0.2677	0.3930	0.0628	0.3671	0.3692	0.6827
35	0.3058	0.8038	0.0390	0.1272	0.3343	0.5505	0.6347	0.9192	0.2503	0.4461	0.4109	0.5852	0.5479	0.9024	0.4235	0.6591	0.3977	0.6386	0.0000	0.0234
36	0.5148	0.8649	0.3412	0.7203	0.2254	0.9761	0.5835	0.8630	0.6155	0.9007	0.3262	0.5416	0.2073	0.3599	0.5283	0.9345	0.8250	0.9999	0.4326	0.6734
37	0.3650	0.6233	0.0014	0.0149	0.4178	0.6413	0.5518	0.6896	0.5363	0.8536	0.3942	0.7442	0.4976	0.8670	0.1022	0.1517	0.3493	0.7020	0.2480	0.4643
38	0.2685	0.5859	0.4281	0.6702	0.4636	0.9515	0.3617	0.6890	0.3169	0.8588	0.0634	0.3094	0.0208	0.0342	0.4085	0.8689	0.1945	0.3147	0.4030	0.7186
39	0.2194	0.4795	0.5249	0.9277	0.0535	1.0000	0.3005	0.7260	0.1867	0.6278	0.5449	0.8476	0.0000	0.0191	0.7063	0.9831	0.1573	0.4493	0.2838	0.6378
40	0.6053	0.9125	0.0282	0.0302	0.4771	1.0000	0.2564	0.4693	0.0816	0.2426	0.1662	0.3231	0.1578	0.2946	0.1300	0.1905	0.0875	0.1323	0.0253	0.1050
41	0.1291	0.1977	0.6517	0.8911	0.1914	0.9994	0.3523	0.5197	0.4055	0.6645	0.2659	0.7460	0.0050	0.0972	0.6342	0.9083	0.2221	0.7950	0.0592	0.2008
42	0.4594	0.8727	0.0895	0.3446	0.4625	0.9582	0.1774	0.5176	0.3147	0.7707	0.0610	0.1511	0.0536	0.1666	0.0600	0.1882	0.0000	0.1202	0.3272	0.7076
43	0.5672	0.9186	0.1257	0.3268	0.3764	0.9211	0.2408	0.7088	0.5930	0.8966	0.1100	0.3154	0.2678	0.6012	0.2388	0.5750	0.6008	0.8375	0.1260	0.2687
44	0.3359	0.4657	0.1329	0.2025	0.1218	0.6505	0.1117	0.5318	0.0926	0.2670	0.3369	0.4935	0.2823	0.7875	0.3226	0.5921	0.4441	0.5761	0.1621	0.2457
45	0.4607	0.7936	0.3852	0.7089	0.2402	0.8406	0.0312	0.2416	0.2161	0.5134	0.3874	0.6902	0.3813	0.6891	0.7101	0.9716	0.6136	0.9177	0.1364	0.2592
46	0.5451	0.8756	0.1477	0.3521	0.0378	0.7911	0.0856	0.5901	0.5114	0.7254	0.4300	0.9528	0.5178	0.7214	0.3591	0.4719	0.2459	0.4541	0.1740	0.5077
47	0.3883	0.7579	0.1009	0.3786	0.4563	0.9993	0.2882	0.5832	0.1635	0.3413	0.1898	0.5628	0.3839	0.8188	0.0704	0.0969	0.1892	0.8264	0.1152	0.1628
48	0.1067	0.2801	0.2509	0.4031	0.1152	0.8919	0.3796	0.7068	0.2870	0.8126	0.0000	0.0000	0.6237	0.9174	0.3393	0.5393	0.1916	0.6292	0.0380	0.0706
49	0.3853	0.7289	0.3831	0.6810	0.1749	0.6377	0.4073	0.6633	0.2096	0.4755	0.3434	0.6791	0.0000	0.0000	0.1014	0.6939	0.3507	0.5952	0.3039	0.4244
50	0.6188	0.7198	0.5468	0.9814	0.0236	0.9503	0.0488	0.2962	0.0533	0.0860	0.0957	0.4841	0.0000	0.0284	0.0000	0.0000	0.2902	0.4304	0.0350	0.2395
51	0.0000	0.3775	0.0000	0.0000	0.0028	0.5076	0.1865	0.4646	0.5894	0.8751	0.1990	0.3451	0.6246	0.9582	0.0000	0.0000	0.4391	0.7302	0.1767	0.4722
52	0.2518	0.4104	0.1854	0.2922	0.3600	0.7626	0.5247	0.9108	0.1703	0.6584	0.2137	0.5670	0.0108	0.0430	0.2440	0.5479	0.1567	0.2429	0.3421	0.6375
53	0.3635	0.8559	0.0489	0.5473	0.2940	0.8981	0.2538	0.6662	0.5349	0.9688	0.1316	0.2241	0.4671	0.7648	0.3986	0.8239	0.4501	0.6947	0.1493	0.4139
54	0.3110	0.6683	0.3427	0.7227	0.3764	0.9600	0.5592	0.9943	0.1872	0.4559	0.0504	0.6343	0.0428	0.0739	0.6194	0.9037	0.2423	0.4358	0.2747	0.4085
55	0.0021	0.1521	0.4256	0.9323	0.3182	1.0000	0.2450	0.6894	0.2449	0.4889	0.0000	0.0264	0.6560	0.9472	0.6019	0.9431	0.3365	0.5266	0.1907	0.3770
56	0.1914	0.4732	0.2371	0.7154	0.3505	0.9784	0.2969	0.5769	0.3746	0.8998	0.3453	0.8569	0.5442	0.9952	0.4693	0.6987	0.0451	0.1012	0.1698	0.5094
57	0.3030	0.5447	0.2525	0.3890	0.3543	0.7365	0.1766	0.2577	0.4940	0.7437	0.0089	0.0788	0.2919	0.5516	0.6188	0.8977	0.0581	0.1563	0.0000	0.0000
58	0.2917	0.5641	0.5515	0.7136	0.1457	0.9991	0.0982	0.1291	0.2041	0.5259	0.4047	0.7660	0.2680	0.4826	0.1862	0.4298	0.6134	0.9618	0.0221	0.0896
59	0.4907	0.7777	0.0044	0.0155	0.1040	0.6092	0.3029	0.5745	0.0000	0.0434	0.0363	0.5217	0.1213	0.3356	0.3309	0.5588	0.4599	0.6936	0.4978	0.9254
60	0.4714	0.6699	0.1314	0.2773	0.4885	1.0000	0.1177	0.3177	0.0289	0.3636	0.4526	0.9500	0.2299	0.2739	0.1861	0.2763	0.3231	0.5751	0.0809	0.1705
61	0.4406	0.7452	0.5180	0.7082	0.4439	1.0000	0.2764	0.4609	0.6397	0.9140	0.3034	0.6609	0.4519	0.7957	0.1243	0.6633	0.0000	0.0000	0.5773	0.7855
62	0.0131	0.0597	0.1974	0.3616	0.3381	0.5077	0.0258	0.0898	0.2328	0.5078	0.3064	0.7825	0.0750	0.2872	0.2282	0.3677	0.1097	0.1818	0.4263	0.6130
63	0.0981	0.1042	0.4703	0.8729	0.2097	0.7270	0.3233	0.5678	0.4410	0.7091	0.0361	0.1082	0.4827	0.6305	0.7404	0.9277	0.3421	0.8264	0.3074	0.5336
64	0.6186	0.8255	0.0197	0.0171	0.0753	0.8554	0.0950	0.2636	0.2934	0.6580	0.0109	0.0520	0.3585	0.5028	0.0992	0.6400	0.0608	0.3721	0.4262	0.6855
65	0.5648	0.7590	0.1963	0.2570	0.7160	0.9993	0.3805	0.5169	0.5452	0.6942	0.4700	0.7108	0.5910	0.8028	0.1500	0.1849	0.2146	0.5776	0.5041	0.7749
66	0.2144	0.5249	0.2012	0.6799	0.3704	0.9974	0.0604	0.1448	0.4197	0.6066	0.0825	0.1431	0.1287	0.2896	0.0000	0.0000	0.0000	0.0000	0.0068	0.0300
67	0.0000	0.0000	0.4718	0.6669	0.0315	0.9362	0.2790	0.4806	0.0813	0.5295	0.4233	0.8160	0.0000	0.0110	0.7057	0.9253	0.1981	0.4500	0.2671	0.5980
68	0.3497	0.3931	0.1842	0.3974	0.0742	0.5280	0.0643	0.2837	0.0245	0.1436	0.0000	0.0130	0.1852	0.4401	0.3254	0.7742	0.0104	0.0535	0.0000	0.0000
69	0.4301	0.8054	0.1445	0.6139	0.3499	0.9987	0.0305	0.0919	0.2393	0.5802	0.0044	0.0343	0.4341	0.6833	0.1289	0.3503	0.2453	0.3928	0.1668	0.4205
70	0.2457	0.4941	0.3765	0.6633	0.0712	0.5684	0.4594	0.7013	0.0781	0.5110	0.0000	0.0000	0.1927	0.4576	0.1413	0.4773	0.5645	0.8314	0.2088	0.9020
71	0.1577	0.5813	0.3028	0.4766	0.1098	0.9529	0.6132	0.9579	0.3711	0.8000	0.4172	0.6695	0.2416	0.4864	0.0822	0.1800	0.0400	0.1574	0.1447	0.2075
72	0.3407	0.5862	0.0241	0.0896	0.0532	0.7498	0.5108	0.7586	0.4594	0.6615	0.4221	0.6887	0.3448	0.7637	0.2615	0.5160	0.4161	0.7843	0.0000	0.0954
73	0.5484	0.7969	0.1618	0.3612	0.2013	0.9490	0.2866	0.7058	0.1999	0.8443	0.2000	0.6533	0.2608	0.6156	0.2548	0.9747	0.4964	0.7012	0.4131	0.9135
74	0.3474	0.5473	0.3333	0.4419	0.4201	0.7405	0.5474	0.7730	0.0820	0.3372	0.3386	0.8667	0.2616	0.5571	0.1054	0.6213	0.2556	0.6294	0.4894	0.7764
75	0.4365	0.7954	0.2153	0.3377	0.0031	0.8361	0.2469	0.6113	0.1714	0.6830	0.3998	0.7476	0.0769	0.0870	0.0097	0.1476	0.4013	0.9355	0.3008	0.8331
76	0.4140	0.9663	0.1384	0.1824	0.0000	0.0000	0.1418	0.6281	0.0002	0.7544	0.1585	0.5804	0.2462	0.2171	0.4643	0.7104	0.4741	0.9017	0.5653	0.7943
77	0.0373	0.1267	0.0000	0.0000	0.4417	1.0000	0.2817	0.3559	0.0294	0.1146	0.2360	0.5327	0.4278	0.6833	0.4074	0.8302	0.3487	0.5341	0.3540	0.4737
78	0.5683	0.7825	0.0000	0.0000	0.3249	0.9968	0.6972	0.9017	0.2537	0.3644	0.5918	0.8694	0.2522	0.5431	0.3357	0.4127	0.1377	0.5503	0.0079	0.0144
79	0.5377	0.9148	0.5521	0.8568	0.0000	0.0000	0.2829	0.7083	0.0000	0.0000	0.2828	0.5986	0.0515	0.1006	0.2873	0.4964	0.4871	0.6282	0.0000	0.0000
80	0.2928	0.3849	0.2867	0.5293	0.4686	1.0000	0.2692	0.5247	0.1643	0.6795	0.3628	0.6169	0.1929	0.4145	0.2413	0.5030	0.3826	0.5165	0.4076	0.6884
81	0.2225	0.3655	0.1618	0.5426	0.2685	0.9937	0.5254	0.8384	0.1547	0.3765	0.1285	0.4969	0.0420	0.0986	0.5041	0.7299	0.6136	0.9127	0.4173	0.8574
82	0.3942	0.7105	0.1247	0.7932	0.3659	0.9999	0.2425	0.3415	0.4697	0.9195	0.2409	0.4730	0.1441	0.4745	0.1355	0.5377	0.2461	0.3043	0.4422	0.8056
83	0.1708	0.4142	0.0863	0.1497	0.0090	0.7509	0.4812	0.5605	0.5787	0.8239	0.1832	0.9898	0.1369	0.6335	0.1315	0.1737	0.0895	0.2702	0.1064	0.2489
84	0.2454	0.3645	0.5013	0.8627	0.0752	0.9056	0.0112	0.0660	0.3322	0.8138	0.0721	0.1024	0.4357	0.5958	0.1676	0.5346	0.3733	0.7713	0.0727	0.1523
85	0.3630	0.8389	0.5315	0.8385	0.6142	0.7614	0.1909	0.2868	0.2979	0.3399	0.0377	0.2598	0.3471	0.8607	0.0840	0.7987	0.2929	0.5726	0.1538	0.1970
86	0.1298	0.1557	0.4769	0.7935	0.0704	0.9994	0.0000	0.0900	0.2498	0.4063	0.1836	0.2801	0.3555	0.5348	0.2136	0.7970	0.2340	0.7127	0.1762	0.4600
87	0.2544	0.5028	0.0395	0.0598	0.2548	0.8164	0.5930	0.8102	0.1082	0.1953	0.5645	0.6509	0.0822	0.1434	0.0000	0.0367	0.0632	0.1321	0.0195	0.0581
88	0.1907	0.4882	0.4553	0.7106	0.2637	0.9985	0.5072	0.7873	0.3228	0.4940	0.5186	0.8702	0.3200	0.4371	0.5138	0.9147	0.1673	0.3013	0.0000	0.0000
89	0.2758	0.5288	0.4028	0.6927	0.1842	0.9994	0.0000	0.0000	0.0876	0.1011	0.0584	0.0753	0.3409	0.5393	0.5744	0.7257	0.2222	0.2951	0.3438	0.6796
90	0.2419	0.5569	0.2834	0.7617	0.5335	0.9939	0.4548	0.6661	0.0365	0.0622	0.3643	0.6652	0.2721	0.5814	0.0982	0.8077	0.0000	0.0000	0.5085	0.8515
91	0.1515	0.8001	0.3783	0.5102	0.0050	0.5122	0.0000	0.0000	0.3040	0.7961	0.4306	0.6172	0.0000	0.0000	0.0000	0.0000	0.1501	0.4212	0.1629	0.2686
92	0.0000	0.0000	0.3085	0.7219	0.0893	0.6627	0.3500	0.4776	0.3127	0.3956	0.0000	0.0000	0.4293	0.7512	0.4570	0.7794	0.0184	0.0176	0.3358	0.6578

Table 13: Cross-validation results: Averages of IoU and Score averages for each fold (@IoU>0 is the calculation of the average counting only detected tumors).