NOVA IMS
Information Management School

MAA

Mestrado em Métodos Analíticos Avançados
Master Program in Advanced Analytics

Conditional Random Fields Improve the CNN-based Prostate Cancer Classification Performance

Paulo Alberto Fernandes Lapa

Dissertation submitted in partial fulfillment
of the requirements for the degree of

Master of Science in Advanced Analiytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

Professor Mauro Castelli, Supervisor

Doctor Leonardo Rundo, Cosupervisor

Acknowledgements

I would like to thank everyone who directly and indirectly have helped during my academic path and particular in my thesis.

This work would not have been possible without Mauro Castelli and Leonardo Rundo's guidance. Thank you for your time, infinite patience and knowledge.

A very special note to Maria, for teaching me that taking life too seriously is a function with diminishing marginal returns (and what diminishing marginal returns are in the first place!) and to Jorge, that among various life skills, has taught me how to give a proper handshake. Thanks to both of you for investing in me - who would have thought that one could become close friends with his professors'?

A freaking huge thanks to Jan for teaching me everything I know about data science, making me explore and always pushing me to believe in myself, more than a friend, a brother. Danke for all the brilliant ideas and projects that came out of our heads over caffeine-fueled discussion in the pond.

Obrigado to everyone in invited researchers room, namely Illya and Carina, for all the lunch breaks and nights in Lux (that I managed to evade).

This work, and my happiness, would not have been possible without your patience and love, Daniela. Thank you for enduring with me (and me) more than I could have ever asked. Of all the motives I have to love you, your smile is the biggest.

A man is nothing without his family, and this one is no exception. Ana thank you for being the light in my life and my moral compass, there is no kindness that can even compare to yours. Vasco, I may have been the fastest spermatozoid, but you are most definitely the nicer, funnier and coolest half of us. Mãe, Pai, thank you for believing in me and being my safety net throughout all these years, one can cannot express in words the importance of having two person loving and caring without doubts or restraints. There is no love like mothers' and no pride like fathers'. To you I am most grateful and words cannot describe it.

Xica, Catarina, Paulo and Dona Paula, my second family: I may be happy now, but all I know about living and enjoying the small things in life, you have taught me.

Thank you all for enduring with me the beautiful rollercoaster that is my life.

Life is a neural network with many hidden neurons

Marisa Fernandes, 2018

Abstract

Prostate cancer is a condition with life-threatening implications but without clear causes yet identified.

Several diagnostic procedures can be used, ranging from human dependent and very invasive to using state of the art non-invasive medical imaging. With recent academic and industry focus on the deep learning field, novel research has been performed on to how to improve prostate cancer diagnosis using Convolutional Neural Networks to interpret Magnetic Resonance images.

Conditional Random Fields have achieved outstanding results in the image segmentation task, by promoting homogeneous classification at the pixel level. A new implementation, CRF-RNN defines Conditional Random Fields by means of convolutional layers, allowing the end to end training of the feature extractor and classifier models.

This work tries to repurpose CRFs for the image classification task, a more traditional sub-field of imaging analysis, on a way that to the best of the author's knowledge, has not been implemented before.

To achieve this, a purpose-built architecture was refitted, adding a CRF layer as a feature extractor step.

To serve as the implementation's benchmark, a multi-parametric Magnetic Resonance Imaging dataset was used, initially provided for the PROSTATEx Challenge 2017 and collected by the Radboud University.

The results are very promising, showing an increase in the network's classification quality.

Keywords: Prostate Cancer Convolutional Neural Networks Conditional Random Fields

Resumo

Cancro da próstata é uma condição que pode apresentar risco de vida, mas sem causas ainda corretamente identificadas.

Vários métodos de diagnóstico podem ser utilizados, desde bastante invasivos e dependentes do operador humano a métodos não invasivos de ponta através de imagens médicas. Com o crescente interesse das universidades e da indústria no campo do *deep learning*, investigação tem sido desenvolvida com o propósito de melhorar o diagnóstico de cancro da próstata através de *Convolutional Neural Networks* (CNN) (Redes Neuronais Convolucionais) para interpretar imagens de Ressonância Magnética.

Conditional Random Fields (CRF) (Campos Aleatórios Condicionais) alcançaram resultados muito promissores no campo da Segmentação de Imagem, por promoverem classificações homogéneas ao nível do pixel. Uma nova implementação, CRF-RNN redefine os CRF através de camadas de CNN, permitindo assim o treino integrado da rede que extrai as características e o modelo que faz a classificação.

Este trabalho tenta aproveitar os CRF para a tarefa de Classificação de Imagem, um campo mais tradicional, numa abordagem que nunca foi implementada anteriormente, para o conhecimento do autor.

Para conseguir isto, uma nova arquitetura foi definida, utilizando uma camada CRF-RNN como um extrator de características.

Como meio de comparação foi utilizada uma base de dados de imagens multiparamétricas de Ressonância Magnética, recolhida pela Universidade de Radboud e inicialmente utilizada para o PROSTATEx Challenge 2017.

Os resultados são bastante promissores, mostrando uma melhoria na capacidade de classificação da rede neuronal.

Palavras-chave: Cancro da próstata Redes Neuronais Convolucionais Campos Condicionais Aleatórios

Contents

List of Figures

List of Tables

Acronyms

ADC Apparent Diffusion Coefficient image.

AUROC Area under the ROC curve.

BCE Binary Cross Entropy.

CNN Convolutional Neural Network.

CoM Centre of Mass registration.

CRF Conditional Random Field.

DCE Dynamic Contrast Enhanced imaging.

DW Diffusion Weighted imaging.

EDC Endorectal Coil.

GS Gleason Score.

ILSVRC ImageNet Large Scale Visual Recognition Challenge.

K-TRANS Transfer constant.

MI Mutual Information Criterion.

PCa Prostate cancer.

PD Proton Density image.

PSA Prostate-specific antigen.

RB Rigid Body registration.

ReLU Rectified Linear Unit.

ROC Receiver Operating Characteristic.

SGD Stochastic Gradient Descent.

T2w T2-weighted image.

Chapter 1

Introduction

1.1 Prostate cancer

The prostate is a gland present in the pelvic district of men, located between the penis and the bladder and typically the size of a walnut. Its main function is to produce the liquid that forms the semen. Prostate cancer (PCa) is characterized by the abnormal growth of cancerogenous cells in that gland. According to the World Cancer Research Fund and the American Cancer Society, Prostate Cancer (PCa) is the second most common form of cancer in men and fourth overall. In 2018 there were 1.3 new millions cases and in the US 30.000 men die every year of related causes [8].

PCa usually develops slowly and without the presence of major symptoms. Because of the typically slow onset, not every diagnosed patient will develop a clinically significant condition to warrant active treatment [36, 51].

1.1.1 Causes

It still is not understood what factors cause PCa, but some have been identified as possible causes [45]:

1. *Age*, it is well understood that men over 50 years old are at a high risk of developing PCa;
2. *Unhealthy habits* such as smoking and alcohol drinking have a strong relationship. Not only a relationship between smoking and PCa incidence has been identified, but also stronger smoking habits with PCa mortality;
3. *Unhealthy eating habits* like lack of consumption of fresh fruits and vegetables. A study showed that there is a strong association with consuming tomato-rich products and lower PCa incidence;

4. *Geography*, PCa is more common in developed regions (ie.e North America, Australia, New Zealand, Western and Northern Europe) but the highest mortality rate is found in low- and middle-income regions (sub-tropical Africa, South America and the Caribbean); the PCa mortality rate in Asia, Africa and Central America is lower than in the other parts of the world. This may be related to a higher life expectancy of men in developed countries and more advanced diagnosis techniques, leading to more diagnoses.

5. *Family History and Genetics* the PCa diagnosis on a family member of a diagnosed PCa patient is estimated at around 20%. The reasons may be related to similar genes, lifestyles and environmental conditions. At the genetic level, several genes and chromosomal regions have been found to be associated with PCa.

6. *Ethnicity* African-American Caribbean men have the highest incidence rates and the mortality rate of PCa among African-American men is the double of white men [45].

7. *Occupation* PCa risk is lower among forestry workers, police officers, office workers and white-collar occupations when compared to others. The risk of PCa is higher in farmers, but this is is generally associated with the exposure to pesticides [45].

1.1.2 Diagnosis and treatment

PCa symptoms are related to an increase of the prostate size, affecting the urethra. This leads to an increase in the need to urinate, pain when doing so or the feeling that the bladder was not fully emptied.

It is important to note that just a single exam is not able to uniquely diagnose PCa, and each has drawbacks and advantages. The most common diagnostic methods are [49]:

1. *Digital Rectal Exame(DRE)* this is a physical exam where the doctor finger is inserted in the patient's body to feel the prostate and surrounding tissue. With this is possible to see if any particular bumps or textures that may indicate the presence of PCa;

2. *Prostate-specific antigen (PSA)* blood test. This exam measures PSA levels, which are associated to be higher values in the presence of PCa. This exam is unreliable, high PSA levels can be associated with other conditions and, in some cases, PCa itself is not associated with high PSA values;

3. *Transrectal ultrasound (TRUS)* a small probe is inserted in the patient's body that emits sound waves, thus creating echoes. This data is then transformed into a computer image. TRUS can be used as a second exam after the DRE or PSA

exams give abnormal results. It can also be used to guide the needle during the biopsy procedure;

4. *Biopsy* a spring-loaded instrument with a needled is used to extract a sample of the prostate tissue. The sample is then sent to a lab and evaluated. While a biopsy provides a quantitive result (the Gleason Score) it has some drawbacks: namely discomfort to the patient; and false-negative diagnosis, because the probe can miss the cancerigenous cells;

5. *Magnetic Resonance Image scan* Using a MRI, the doctor can visually evaluate the prostate and, if any suspicion arises, can recommend a biopsy to be performed. This is normally a non-invasive method.

Currently, medical consensus recognizes the potential of using MRI as a mean to guide the biopsy to larger and probably more significant tumours [56].

Depending on the stage of the cancer several treatments can be proposed: watchful waiting (delay the treatment and wait if any the symptoms develop), active surveillance (regular exams to ensure any PCa progression is found early), radical prostatectomy (i.e., removal of the prostate) and radio- or hormone- therapy.

1.1.3 Biopsy Gleason Score

The Gleason Score (GS) analyses the tissue extracted from a biopsy, based on its appearance and on how much it looks like healthy tissue. More abnormal looking cancers, that are more likely to grow and spread, are given a higher grade[49].

The cancer is measured on a scale from 1 (normal tissue) to 5 (very abnormal). Almost all of the cancers are graded 3 or higher [49].

The GS measures the two areas that make up most of the cancer, each area is given a grade and their addition yields the GS. The first number is the most common grade in the tumor tissue. For example, if a GS is given as 3+4=7, most of the tumor is of grade 3 and less of it is grade 4, then adding to a GS of 7 [49].

1.2 Computer assisted diagnosis

Several methods have been presented that proposes the usage of medical images and Machine Learning applied to the task of correctly detecting and staging cancer, in a process called Computer Assisted Diagnosis (CAD).

The usage of CAD can range from data preprocessing tasks (i.e registration, Region of Interest selection, feature extraction and selection) [27], to several cancer-related applications that benefit from Deep Learning (DL) [30].

Models like linear regression, ensemble learning classifiers, Gaussian processes or support vector machines have been used, with varying degrees of success.

A particularly popular set of models are the Deep Convolutional Neural Networks (CNNs) that have found success in the medical image analysis (MIA) field, in various tasks: of unsupervised learning problems (problems that do not have a target variable to measure the model quality) to supervised learning problems [30].

In the field of supervised MIA, three main challenges can be identified: image classification, detection, and segmentation. All three have a series of exams or images as input, but the desired output differ [30]. Image classification problems (such as this work) have a single variable as output (e.g. cancer present or not). Image detection models define boundaries around objects of interests (i.e. organs, regions or lesions) [46]. Lastly, the segmentation problem's goal is to identify the voxels that make up the object of interest (i.e the boundaries or the interior).

Various anatomical applications have been found [30]: in the head region, DL with MRIs has been used for brain MIA (e.g. disorder classification, lesion/tumor segmentation/classification or survival prediction) or eye MIA (e.g. blood vessel segmentation, glaucoma detection). In the torso region, DL has been used for cardiac MIA (e.g. Ventricle slice detection, heart structure detection or coronary calcium detection), liver lesion segmentation or kidney localization. Lastly, DL has been used for musculoskeletal MIA (e.g knee cartilage segmentation, vertebrae localization or even hand age estimation).

In the anatomical region of the prostate, CNNs networks have been employed in CAD tasks like PCa segmentation [17] [54] [53] and classification [32] [57] [2].

With regards to Conditional Random Fields, they have been used for segmentation tasks in PCa [5], [39] [19] and for brain cancer segmentation [58] as well.

In all these applications, some challenges are always present [30] [46], namely the lack of large training data sets, absence of reliable ground truth data or the difficulty in training large models. Nonetheless, some factors can always be considered important in the success of DL models [30]: expert knowledge, novel data preprocessing or augmentation techniques, and the application of task-specific architectures.

The goal of this work is to merge the classification abilities of CNN and the local segmentation provided by CRFs and develop a novel way of diagnosing PCa, that to the best of my knowledge has not been proposed.

This work is organized as follows: in chapter 2 a brief introduction to Magnetic Resonance Imaging, Convolutional Neural Networks and Conditional Random Fields is given; in chapter 3, subsection 3.1 introduces the dataset and the treatments performed before using it; subsection 3.2 presents four off-the-shelf CNN architectures used and the one created for this work. Finally, chapter 4 discusses the training methodology and the results obtained.

Chapter 2

Introduction to deep learning for medical imaging

To better understand the research work that has been carried out, some clinical background and theoretical knowledge of its various parts are recommended.

This chapter aims to provide a short introduction to them, by organizing the contents as follows: section 2.1 is devoted to Magnetic Resonance Imaging (MRI) data acquisition and preparation, with particular interest to prostate cancer diagnosis. Section 2.2 introduces Convolutional Neural Networks (CNNs) and their strengths in computer vision, while section 2.3 presents Conditional Random Fields (CRFs). Lastly section 2.4 gives an intuition on the working of the Semantic Learning Machine[20], a neuroevolutionary algorithm.

2.1 Prostate Magnetic Resonance Imaging

MRI is an acquisition modality that allows for studying both the body human anatomy and physiology, thus providing insights into the diagnosis of different diseases and conditions. It does not only provide high-resolution images, especially for analyzing the structure of soft tissues, but also information at the molecular level, without requiring an invasive procedure [40].

As a matter of fact, the human body is composed of different tissues containing mainly water molecules that contain protons. When protons are excited (through a pulse caused by the MRI scanner), they emit a radio frequency signal that is received by a coil.

From the moment the pulse is produced, two-time sequences can be identified: when the protons receive the pulse and go to an excited state (i.e., T1 or longitudinal relaxation time) and how long they take from returning from their excited state to their

initial state (i.e., T2 or transverse relaxation time) [38] [44]. These times are measured and serve as the source of contrast in the MR image—the premising being different tissues type have different relaxation times. [38] This allows certain tissue properties to be enhanced by careful parameter tuning.

Sometimes, a contrast mean (typically based on Gadolinium) can be administered to the patient for a higher image contrast, also allowing for dynamically evaluating the vascularity of the tumor microenvironment by means of Dynamic Contrast Enhanced (DCE). Moreover, depending on the magnetic field strength, an endorectal coil is generally be used to increase the Signal-to-Noise (SNR) especially in the case of 1.5T MRI scanners. When 3.0T MRI scanners are exploited, the acquisition can be performed *via* a pelvic coil guaranteeing a good SNR.

The existing methods for PCa diagnosis have been characterized by overdiagnosing low-risk lesions and underdiagnosing high-risk cancers [56]. Usually, random biopsies are performed but this comes with serious disadvantages, namely: a likely increase in complications due to the over-sampling of healthy tissue; tumors outside the biopsy site could be easily missed and it may be difficult to determine the site of a previous biopsy when repeating the exam [9], which might cause hemorrhages.

2.1.1 Multiparametric MRI

Multiparametric MRI (mpMRI) of the prostate comprehensively depicts of the prostate, allowing for better tumor detection, and has "recently emerged as the most promising imaging modality for this application"[56], when compared to other biopsy or traditional Prostate-Specific Antigen (PSA) assessment.

By bearing this in mind, there has been crescent recognition of mpMRI as mean to guide the biopsy to larger and probably more significant tumors [56], essentially creating a synergy between two very different diagnostic methods.

An mpMRI can be obtained by the capture of multiple MRI sequences carefully tuned. An MRI sequence is defined by a particular set of parameters that change the types of tissues or features that are emphasized during the acquisition process. An mpMRI consists of anatomical—i.e., T1-weighted (T1W), T2-Weighted (T2W), Proton Density (PD)—or functional sequences, such as Diffusion Weighted Imaging (DWI), DCE.

To better improve PCa diagnosis, several modalities, which often convey complementary information, should be used in combination. Clinical consensus defends that T2W imaging should be used together with at least two functional modalities, [29], because it can improve cancer detection, location, and staging, and then be used to help define personalized therapies [9].

2.1.1.1 MRI Sequences

2.1.1.2 T2W

T2W sequences measure the time taken by the excited protons to return to their normal state. It is particularly well-suited for cancer detection, characterization and localization [56].

On the prostate area, T2w is well suited to depict its anatomy because it returns high signal intensity in the peripheral zone, which is formed of muscle and glandular tissue, when compared to central and transitional zones [9].

T2w sequences are useful for PCa-related applications because both PCa and prostate have low signal intensities in the central and transitional zones. In the peripheral zone, where high-intensity values are expected, low values might be a clue, not only of cancer cells, but also other conditions, such as biopsy-related hemorrhages, fibrosis or lesions caused by other therapies [9].

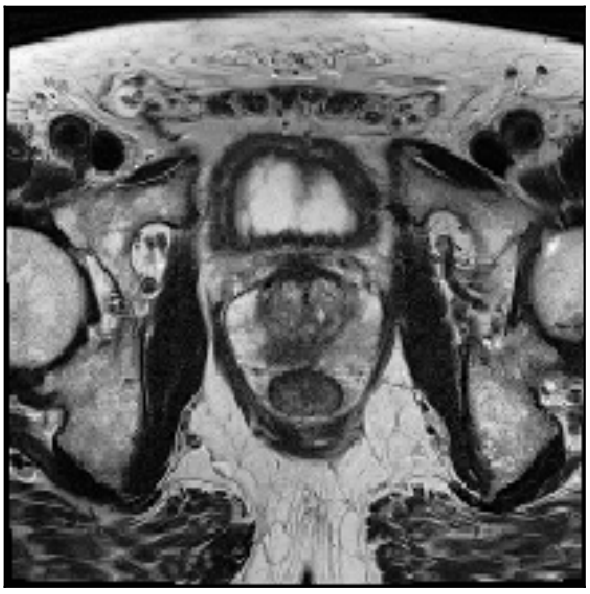


Figure 2.1: T2W slice extracted from patient #29

2.1.1.3 Diffusion-weighted Imaging (DWI)

DWI measures the water diffusion characteristics of tissue cells. A quantitative map can be achieved by means of the Apparent Diffusion Coefficient (ADC) [9].

Compared to normal tissue, PCa typically has tightly packed cells, dense surrounding regions and intra- and inter-cellular membranes that reduce water motion. PCa cells typically have lower diffusion values than healthy cells in ADC images. Furthermore, a relationship has been found between lower diffusion values and higher PCa aggressiveness.

The combination of DWI sequences and T2W showed to significantly improve the diagnosis quality of PCa [56].

Taking this into consideration, caution is still necessary: although ADC values are a good indicator, individual variability can strongly impact the accuracy of ADC in PCa diagnosis [9].

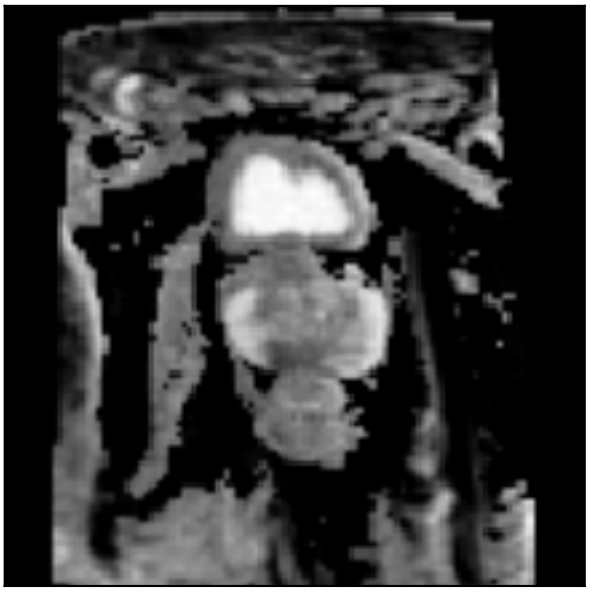


Figure 2.2: ADC slice extracted from patient #29

2.1.1.4 Proton Density (PD)

As the name suggests, PD reflects the presence of protons in the body tissue, with higher density regions appearing brighter. PD provides good distinction between fat, fluid and cartilage. [35] More specifically, PD images are formed as a mix between T1 and T2W images, by having long TR times and short LR times. [35]

2.1.1.5 Dynamic Contrast Enhanced (DCE)

DCE employs an external agent to improve image quality. In DCE, a Gadolinium-based contrast is injected into the patient blood flow. This agent then travels through the patient's vascular system, further characterizing it. Typically, the blood vessel structure of a tumor is very different from the one of healthy tissue. Indeed, tumors have an increased number of blood vessels, higher permeability and higher amount of interstitial tissue. These conditions make the patterns of cancer tissue different with respect to healthy tissues.

The contrast values can be decomposed into several factors: regional blood flow, size and number of blood vessels and their permeability. It is not possible to separate these components individually, but their combined effect can be modeled using a

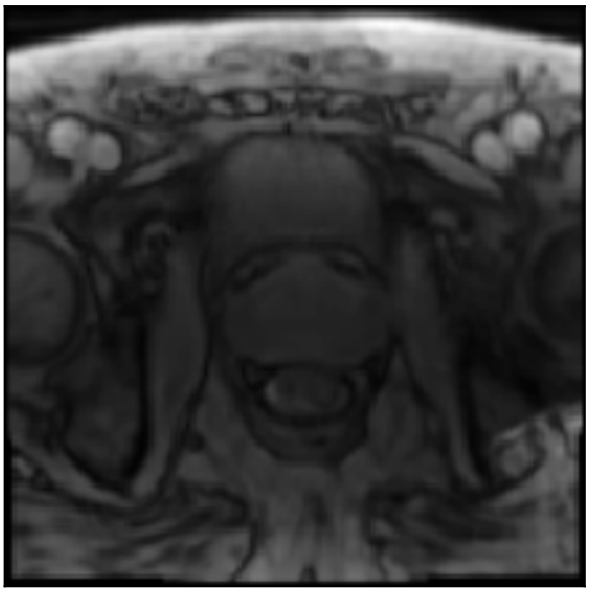


Figure 2.3: PD slice extracted from patient #29

transfer constant, K^{trans}. This K^{trans} is an index that characterizes the presence of gadolinium in the vascular endothelium (the membrane that covers the interior of blood vessels) [55].

From a health economic point of view, DCE images are more expensive to collect and cause more discomfort to the patient—since agents like Gadolinium need to be injected—as well as raise a safety risk because there is evidence of possible depositions in the body [21], such as in the brain [15].

2.2 Convolutional Neural Networks

Neural networks (NN) have become one of the most common supervised learning techniques. They are able to learn complex patterns from unstructured data (i.e. text or images) with little domain knowledge needed. NNs are arranged in a hierarchical fashion, that is in layers. Each layer is capable of extracting simple features from its inputs that are then refined by the next layers.

The element responsible for the feature extraction is the neuron (or hidden unit). Each layer has a variable number of neurons. The neurons take a varying number of inputs (from the previous layer) and perform a dot product using weights that are improved over the training procedure.

Convolutional Neural Networks (CNNs) go a step further. By using the convolution operation they can perform their task on two- or higher-dimensional inputs. They can consider not only the input pixel but also its neighboring region, making them well-suited for image applications.

Figure 2.4: K-trans slice extracted from patient #29

2.2.1 Training

After defining the network architecture details (more details in sections 2.2.2 and 3.2.1), it must be trained. The training is the procedure that creates the model, making it learn the features in the data. It is an marginal procedure, where improvements are incremental over time.

For the training, two parameters need to be defined: a loss function and an optimizer (more details about this in sections 2.2.4 and 4.1.2, respectively).

The loss function measures how well the model is able to predict the data when compared to the ground truth.

An optimizer adjusts the parameters of the network, taking into account the feedback it receives from the loss function. This adjustment is done by means of *backprograpagation*.

Initially, the parameters of the network (weights) are randomly chosen. This means that in the early epochs the network is just implementing random transformations without any predictive quality. This is why loss values are typically so high in the early epochs.

A network is typically trained in the following fashion:

1. Randomly initialize the network weights
2. For a pre-defined number of epochs, or until a convergence criterion is met:
    a) Present a batch of data points to the network and generate a prediction based on them ($X \rightarrow Y'$);

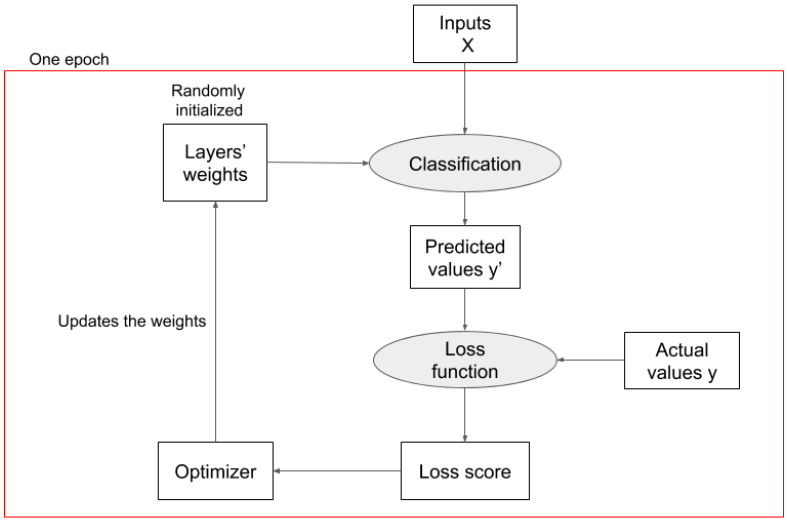


Figure 2.5: Training loop

b) Calculate the loss score of each pair {prediction, true value}, ($\{y', y\}$) - this measures of good the prediction compared is to the actual value;

c) Calculate the loss score based on the individual loss values (e.g., their the sum or mean);

d) Present the loss score to the optimizer;

e) The optimizer then performs weight updates on the networks output layers and propagates them to the network's hidden layers;

f) Go to step a).

This is the training loop, and when enough iterations are done in a dataset, it should return a trained network as optimal as possible. The way these weights are updated is shown in more detail in section 2.2.3 and 2.2.4.

2.2.2 Layers

Layers are the building blocks of a CNN: from the input of the network until a probability is returned, layers and their neurons communicate through weights to extract features, explore relationships in the data and calculate the network's output. Careful layer configuration can promote faster convergence and lower training times and has an impact on its predictions qualities.

The interactions among layers have extensively been studied, giving rise to the development of many state of the art CNN architectures. The most relevant architectures have been used in this work, like VGG16, AlexNet or ResNet, and are presented in more detail in section 3.2.1.

This section gives a quick introduction to the most common layer types that were used in this work, just to better understand their roles and mechanisms.

Please note that little mathematical notation is presented. For a more technical explanation the reader is directed to the bibliography, or the books "Deep Learning with Python" by Chollet [11] or "Deep Learning" by Goodfellow et all [14].

2.2.2.1 Convolutional

A convolution (conv) is an operation that allows the learning of local patterns, defined by a scope (also called the stride). These learnable patterns can be edges, textures, lines, etc and have two main characteristics:

- They are *translation invariant* meaning that a convnet can recognize a pattern in any location of the image regardless of the original learning location. E.g., it is able to recognize a lesion in any region of the image, even if during the training all the lesions were in the same position.
- They can learn *Spatial Hierarchies*, initial conv layers will learn small local patterns that will provide the next layers with more complex patterns and so on. This allows convnets to learn complex relationships in the data.

In image analysis, convolutions operate over three dimensions: height, width and channels- that in this work match three mMRI sequences.

A convolution extracts patches from the input and makes dot-product matricial operations between them and its weights. This produces a feature map: it represents the desired features, edges, textures, etc. The number of features the layer learns can be defined by the depth of the output. The depth, width, and height are hyperparameters of the network that need to be defined *à priori*.

With this information in mind, convolutional layers have two key parameters:

- *Patch size* extracted from the input when considering local patterns, typically $3x3$, $5x5$ or in some cases $1x1$, and
- *Output depth*, the number of channels computed by the layer.

Each conv layer will have $z \cdot z \cdot d$ parameters to learn, z being patch size and d output depth.

2.2.2.2 Max Pooling

Max pooling is an operation that allows the reduction of the feature-maps extracted by the conv layers, and introduces spatial-filter hierarchies.

Max pooling extracts sliding windows from the input and outputs the maximum value present in the window. This effectively downsamples the input size by a factor, determined by the stride [10].

The max pooling operation typically has two parameters :

- *Stride* how many pixels the filter shifts at a time, and
- *Windows size* the region to consider when applying the max function. Lower window size values keep more local information.

2.2.2.3 Batch normalization

Typically, normalization is a preprocessing step done one time before feeding the dataset to the model. This operation converts the input variables into the same scale. Usually, it is performed on supervised learning algorithms that are sensitive to the scale of the inputs. This operations, therefore, promotes faster convergence and better classification performance.

Batch Normalization (BN) extends that concept not only for the input of the first layers but also to the hidden layers, by centering and scaling the output of the previous layer.

This regularization effect reduces the values of the weights change, thus preventing overfitting [18].

2.2.2.4 Fully Connected (FC)

This layer is the building block any Neural Network, it simply outputs the dot product between the inputs and its weights.

This layer can be used as a hidden layer to extract feature or as the last layer, to output the model's prediction. An activation function can be applied on its output, so that its scales changes, or certain behaviors that ease training are enhanced.

2.2.2.5 Activation

The activation layer applies a function to the output of a previous layer, typically an FC or conv layer.

Activation functions are particularly important because they allow the modeling of non-linear relationships, improve the generalization ability or just make the output on the network in the range $[0,1]$ to represent a probability.

The activation functions used in this work's architectures are now presented:

Sigmoid The sigmoid is a very known function that has an S-shape, in the range $[0,1]$, as illustrated in figure 2.6. Because of this, it was used as the last layer in every architecture as a way to create the probability of an image having PCa.

In the case of the multiclass classification problem, the softmax function should be used instead, that is the generalization of the sigmoid function.

ReLU Rectified Linear Unit is function that takes the positive part of its arguments: $f(x) = max(0,x)$.

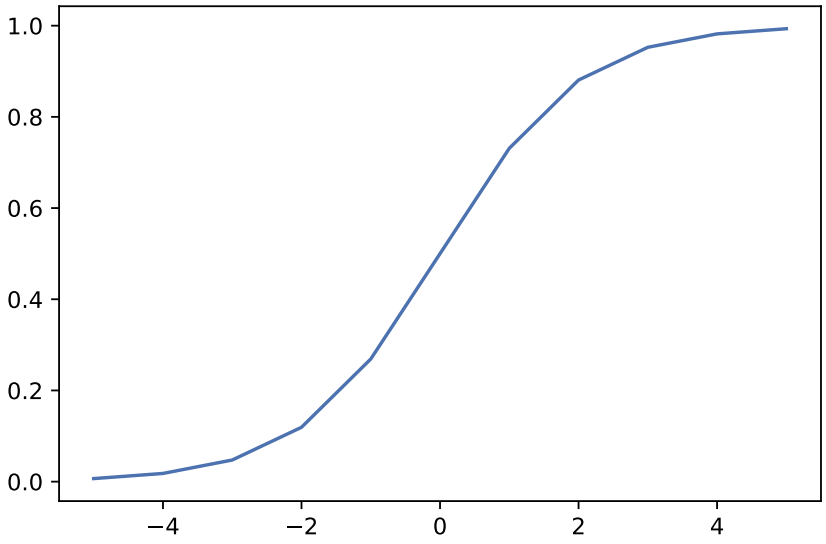


Figure 2.6: Sigmoid activation function.

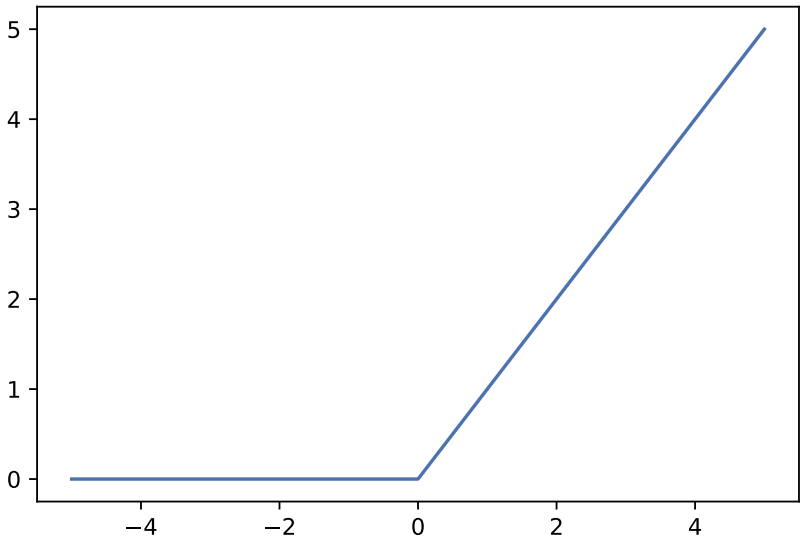


Figure 2.7: ReLU activation function. Notice how it resembles a ramp.

Graphically it looks like a ramp, as shown if figure 2.7 and increases training speed significantly, thus allowing the creation of deeper architectures.

But the ReLU function has very desirable properties compared to the previously popular activation functions because it promotes better gradient propagation, easier computation and faster backpropagation calculations [24].

Its importance in the DL community is explained in more detail in section 3.2.1.1.

2.2.2.6 Dropout

A dropout layer is a very simple way to prevent overfitting and reduce training time in NNs. In each training epoch, a random number of weights between two hidden layers are dropped out of the network and ignored [50]. The number of weights that are ignored depends on a hyperparameter defined by the user.

This significantly reduces the risk of overfitting because it reduces the reliance of the network in certain neurons.

2.2.3 Backprograpagation

While it is easy to calculate the error of the network's output (through the loss function) it is conceptually difficult to understand the impact of each neuron and layer during the classification process.

Backpropagation does the inverse path that an input observation does, and distributes the error through the network's layers.

This is done by computing the gradient of the loss function (with chain rule derivatives) with regard to the network's weights and then slightly adjusting them in the correct direction.

2.2.4 Optimizers

It is the job of the optimizer to perform the adjustment. Each optimizer uses different techniques to do this. Some optimizers have mechanisms that prevent sudden jumps, others allow for different parameters to have different updates, and others prefer certain local optima characteristics. This means that there is no overall best optimizer, but different problems and datasets require different solutions.

2.2.4.1 Stochastic Gradient Descent (SGD)

The simplest optimizer is the Stochastic Gradient Descent (SGD) and is the default. This is the default SGD formula:

$$w = w - w * \Delta_\theta$$

Overall it works fairly well but falls short in complex gradient landscapes (e.g. saddle points). In these situations, the network gets stuck in a local optimum, and in later iterations may not converge because of the learning rate (lr) being too high.

With this in mind, a more complex SGD implementation has been developed with particular features that solve most of these problems:

1. *Decay rate*: as the training progresses the learning rate (lr) slowly decreases by a set decay rate (d): $lr = lr \cdot d$. This prevents the weights from jumping around the optima in the latter stages of the training.

2. *Momentum* SGD has troubles navigation complex loss functions surfaces, like ravines, that are common around local optima. Momentum damps oscillations, thus making convergence faster.

3. *Nesterov Momentum* The previous approach has the shortfall that the updates can be too high and then miss the local optima. Nesterov momentum tries to predict where the next weight updated will be and calculates the gradient into that position.

All this features and hyperparameters come into place in this formula [10]:

$$lr = d * lr$$

$$v_t = \gamma \cdot v_{t-1} + lr \nabla \theta J(\theta - \gamma \cdot v_{t-1})$$

$$w = w - v_t$$

where,

1. w it the neurons weight;
2. $\nabla \theta J(\theta)$ is the gradient of the loss function w.r.t. to the weight;
3. γ is the momentum (usually a high value like 9.9);
4. d is the decay rate;

2.2.4.2 RMSPROP

RMSPROP is an optimizer that adapts the learning rate to each weight and is based on Adagrad. Most frequently activated weights (e.g. common features) have lower learning rates, preferring smaller updates; larger updates are done to more sparsely used parameters. This is implemented by Adagrad:

$$g_{t,i} = \nabla_\theta J(\theta_{t,i})$$

$$w_{t+1,i}, i = w_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} g_{t,i}$$

This is, $g_{t,i}$ is the partial derivative of the loss function w.r.t to parameter θ_i at time t. And $\sqrt{G_{t,ii}}$ is the sum of the square of the gradients of parameter w_i up to time t.

With this approach, Adagrad eliminates the need to tune down the lr but now accumulates quadratic growing gradients in the denominator, making it shrink towards 0, at which point no more significant changes are performed.

RMSPROP tackles this by dividing the learning rate by the square root of the moving average of the squared gradient. This adds a weighted average between the current gradient and past gradients, this weight is defined by the parameters β [10]:

$$E[g^2]_t = \beta E[g^2]_{t-1} + (1-\beta)(\frac{\delta C}{\delta w})^2$$

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{E[g^2]_t}} \frac{\delta C}{\delta w}$$

2.2.4.3 Adam

Similar to RMSPROP, ADaptive Moment Estimation (Adam) keeps an average of past gradients of v_t, but it is also keeping an exponentially decaying average of past gradients of m_t, as an approach similar to the SGD's momentum.

Adam is particularly efficient, requires little memory and is suited for problems large in terms of data and/or parameters [22]. Compared to others, Adam behavior prefers flat local optima in the error surface.

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$

$$b_t = \beta_2 m_{t-1} + (1-\beta_2) g_t^2$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$$

m_t and v_t are values related to the weight gradient (the mean and variance respectively), and β_1 and β_2 are the decaying rates. The authors empirically show that Adam [22] works well in practice.

2.3 Conditional Random Fields

When dealing with image analysis, a context can be considered: the value of a pixel certainly is related to the one of surrounding pixels: homogenous regions exist. It does not make sense to have a random pixel of a "cloud"on a region labeled as "grass".

A CRF allows for this dependence to be modeled, by defining a *discriminative undirected probabilistic graphical model*, representing relationships between two different sets of features: observed and unobserved [33].

A discriminative model learns the conditional probability distribution $P(y|X)$: the probability of y given X. Opposing this, a generative model learns the join probability distribution $P(X,Y)$: the probability of both X and y [41].

In our case, a generative model would learn the probability of a pixel being black and belonging to a cancerous region. A discriminative model would learn the probability of being a cancerous region, knowing that it has a black pixel.

An undirected probabilistic graphical model means that when inferring the class of an observation y_i, not only the input variables associated with y_i, X_i needs to be accounted for, but also its y_i neighbors, $y_{i-k}, y_{i-k-1}, \ldots, y_{i+k-1}, y_{i-k}$. This constraint promotes homogeneous regions.

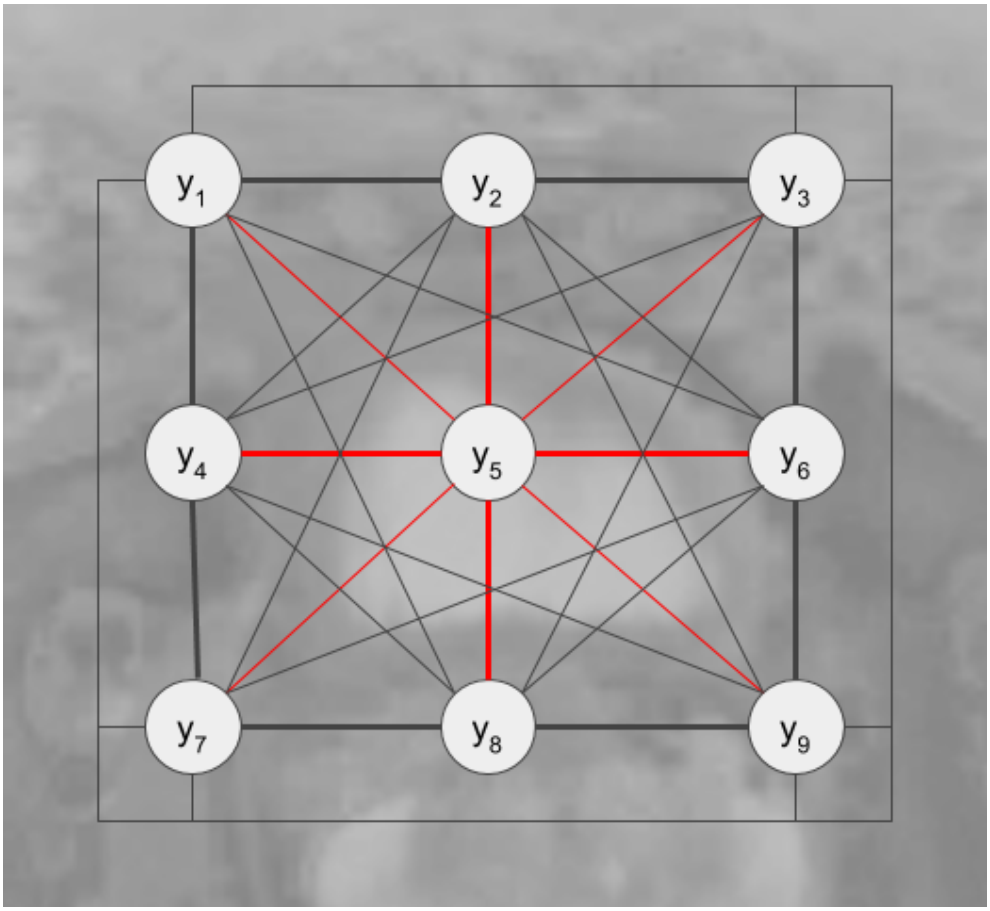
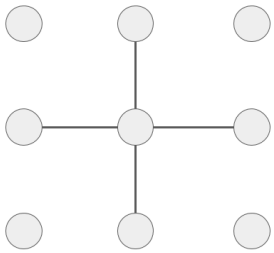


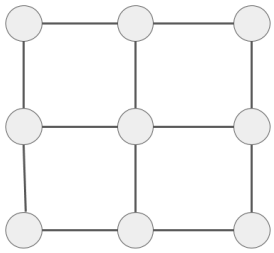
Figure 2.8: The predicted value y_5 does not depend only on the input image and the extracted features but also on the predicted values for the adjacent values $y_1, y_2, ...y_8, y_9$.

A CRF defines a Random Markov Field, by means of an undirected graph, (V,E). A graph defines a set of random variables with nodes V (in this case pixels) and the edges E that connect them.

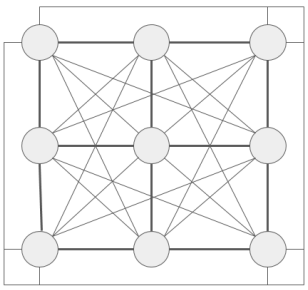
Furthermore, this relationship was structured as a pairwise model: each label y_i (e.g. cancer or not) has associated a set of observed values X_i on the image (traditionally RGB values, in this case, features extracted from MRI).



a



b



c

Figure 2.9: Common graph structures for image segmentation. a, b 4-,8-grid respectively. c Fully connected.

For image segmentation, the following graph structures are common: 4- or 8-grid graphs, where only neighboring pixels are connected and fully connected graphs, where all pairs of pixels are connected by edges. These grids are illustrated in Figure 2.9

Grid CRFs are very efficient when inferring but suffer from some limitations: only model local interactions and excessively smoothen object boundaries. Fully-connected

CRFs are slower when compared to Grid CRF, but are not limited to local interactions and allow better-defined object boundaries.

For this implementation, a Fully connected CRF was considered, where all nodes are connected.

To understand how a fully connected CRF works, it is necessary to introduce the notion of Energy. It can be defined as the cost associated with assigning a given label to a given data point.

Based on [23] and [59] lets define:

- X_i as a random variable associated to pixel i, which represents the label assigned to pixel i.
- X_i can take any values from a pre defined set of labels $\mathscr{L}$. In our work $\mathscr{L} = 0, 1$, denoting the presence or not of cancer, respectively.
- $\mathbf{X}$ is the vector formed of random variables $X_1, X_2, ..., X_N$, with N being the number of pixels in the image.
- A graph $G = (V, E)$, where $V = X_1, X_2, \ldots, X_N$.
- An image (global observation) $\mathbf{I}$.

The pair ($\mathbf{I}$, X) - mapping of the input $\mathbf{I}$ to the mask X- can be modelled as a CRF characterized by a Gibbs distribution of the form $P(X = x|\mathbf{I}) = \frac{1}{Z(1)} exp(-E(x|\mathbf{I}))$. $E(X|\mathbf{I})$ is the energy of the configuration and $Z(\mathbf{I})$ the partition function. $E(X|\mathbf{I})$ will be abbreviated to $E(X)$ from now on.

The energy $E(X)$ is composed of two parts: unary and pairwise, defined by [23] as:

$$E(x) = \sum_i \Psi_u(x_i) + \sum_p \Psi_p(x_i, x_j)$$

.

The component $\Psi_u(x_i)$ is the unary energy component: it measures the cost of the pixel i taking the label x_i. This unary energy predicts the label for a given data point without taking into consideration the smoothness and consistency of the assignment. The unary energy was obtained at the output of the feature extraction phase of a CNN. [59]

$\Psi_p(x_i, x_j)$ corresponds to the pairwise energy. It measures the cost of assigning label x_i, x_j to pixels i and j simultaneously. It ensures image smoothness and consistency: pixels with similar properties should have similar labels. It is defined as:

$$\Psi_p(x_i, x_j) = \mu(x_i, x_j) k(f_i, f_j)$$

$$k(f_i, f_j) = \Sigma_{m=1}^{K} w^{(m)} k^m(f_i, f_j)$$

$$k_(f_i, f_j) = w^{(1)} \underbrace{exp\left(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2} \right)}_{appearance\ kernel} + w^{(2)} \underbrace{exp\left(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2} \right)}_{smoothness\ kernel}$$

where θ_α, θ_β and θ_γ are hyper parameters and $\mu(x_i, x_j)$ is a label compatibility function. It introduces a penalty if nearby pixels i, j have different labels. In this implementation Potts model is used, $\mu(x_i, x_j) = [x_i \neq x_j]$ [23] [59].

$$\mu(x_i, x_j) = \begin{cases} 1\ if\ x_i = x_j \\ 0\ otherwise \end{cases}$$

The appearance kernel is inspired by the notion that nearby similar pixels are more likely to be of the same class. The smoothness kernel removes small isolated regions. The degrees of nearness, θ_α and θ_γ are hyperparameters defined a priori.

[59] defined the pairwise potentials as weighted Gaussians kernels in the form:

$$\Psi x_i, x_j = \mu(x_i, x_j) \Sigma_{m=1}^{M} k_G^m(f_i, f_j)$$

where $k_G^{(m)}$, m = 1,...,M is a gaussian kernel applied on feature vectors, derived from image features.

Minimizing the CRF energy $E(X)$ returns the most probable label for the input image. For ease of computation, the mean field approximation can be used [23].

2.3.1 Mean field approximation

To calculate P(**X**), an exact or approximate inference method can be used. Exact inference tries to learn the exact function P(**X**). The most popular method, junction tree, tries to convert the graph into a tree, by grouping variables.

This method can require exponential time in the worst case, so approximate methods were developed.

Several methods have been developed for approximating the CRF parameters: pseudo-likelihood, belief propagation and Markov Chain Monte Carlo (MCMC)[52].

In the context of image segmentation, the graph structure can quickly grow in complexity: a 64x64 pixel image will have 4096 nodes, and $C(4096, 2) = 8\ 386\ 560$ different edges. For higher-resolution images, this number grows even higher.

While faster to train, the training time of traditional CRF inference methods training time is still suboptimal, and another drawback arises: the traditional training methods for CRF are not adapted for training using backpropagation.

While the feature extraction layers could be trained separately and their outputs then fed to the CRF to be trained, this would not allow the desirable feature of end to end training.

Mean field approximation on a CRF solves both these problems. Training a fully connected CRF with this approach instead of MCMC can be two orders of magnitude

faster, [23]. Most importantly, Mean field approximation it can be rewritten as a set of Recurrent Neural Network (RNN) layers.

Mean field approximation consists in approximating the distribution P(**X**) with a simpler distribution Q(**X**), that can be written as the product of independent marginal distributions:

$$Q(\mathbf{X}) = \prod_i Q_i(\mathbf{X}_i)$$

, subject to:

$\sum_{x_i} Q_i(\mathbf{X}_i) = 1$

$Q_i(x_i)$ is defined as a function that can be updated iteratively, using this algorithm:

Initialize Q: $Q_i \leftarrow \frac{1}{Z_i} exp\{-\phi_u(x_i)\}$;
while *not converged do* **do**
- Message passing from all X_j to X_i:
$\tilde{Q}_i^{(m)}(l) \leftarrow \sum_{j \neq i} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) Q_j(l)$ for all m
- Compatibility transform:
$\hat{Q}_i(x_i) \leftarrow \sum_{l \in \mathcal{L}} \mu^{(m)}(x_i, l) \sum_m w^{(m)} \tilde{Q}_i^{(m)}(l)$
- Local update:
$Q_i(x_i) \leftarrow exp\{-\psi_u(x_i) - \hat{Q}_i(x_i)\}$
normalize $Q_i(x_i)$

end

Algorithm 1: Mean field approximation in fully connected CRFs

2.3.2 Conditional Random Fields with Convolutional Neural Networks

CRFs do a good job of using the inputs to create segmentation masks that make sense, are reasonable and accurate. Traditionally, the input given to the CRF were features defined à priori (e.g. color, texture, opacity) by the human operator, with manual tuning and selection. With this, the process was not streamlined: the feature extraction was independent of the classification and naturally often far from the optimum.

CNNs appear promising here because they are able to extract relevant feature maps, without needing them to be defined à priori. This happens because the model will naturally define and optimize the features better suited for the task - in a fashion no human operators could not do-.

The primary idea is that the features extracted from the convolutional layers will serve as the unary energy (e.g. inputs) fed to the CRF, as figure 2.10 illustrates.

Some CNN's have used the CRFs as a separate step of the pipeline (e.g. train the CNN separately and then train the CRF), while others have used implemented them directly in the architecture [4] [59]. The latter has achieved that of the art performance in several domains [4], and the results can be seen in figures 2.10 and 2.11 and have

the advantage of making the training streamlined, as the process is reduced to one task.

Figure 2.11 compares several image segmentation techniques. It is clear that more complex feature extraction methods (i.e. CNN's) and bigger CRF configurations (e.g. fully connected grids) achieve the best results. For example, notice the high marginal improvement, when compared to previous techniques, of integrating the CRF directly in the networks architecture.

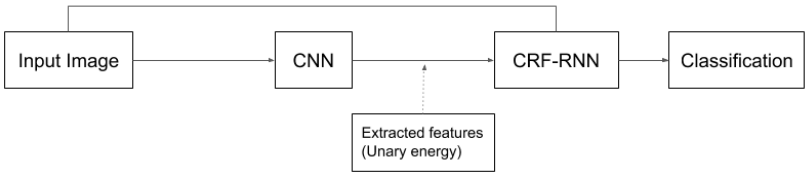


Figure 2.10: The features and classifications extracted by the CNNs can be further improved by applying a CRF model.

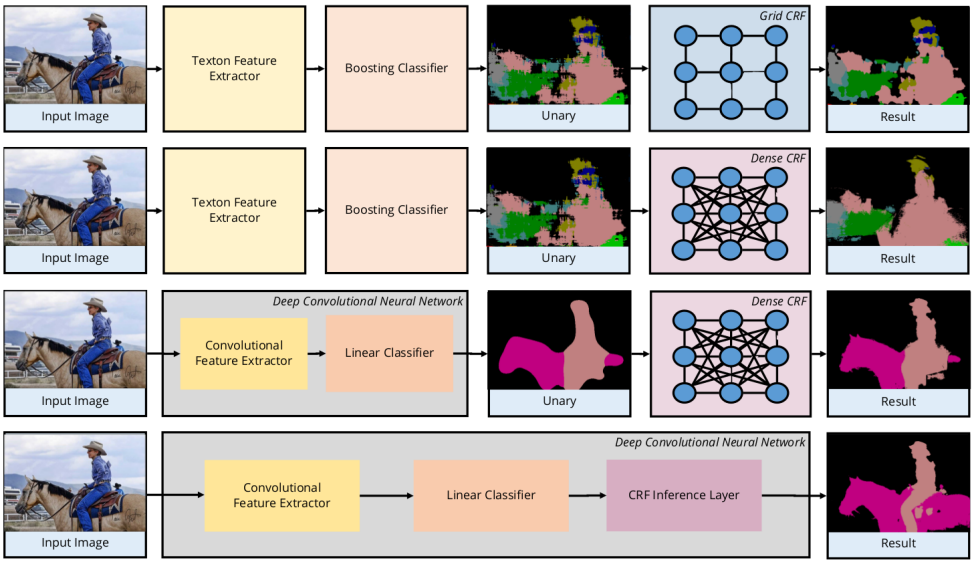


Figure 2.11: Evolution of CRFs at image segmentation. Notice the impact of different CRF grid configurations and improvement when using a CNN as a feature extractor. Extracted from [4]

2.3.3 Conditional Random Fields as Recurrent Neural Networks

As shown in section 2.3.2, it is highly desirable to integrate the CNN's and the CRF in the same process, allowing for end-to-end training. In a novel way, presented by [59], the Mean field approximation algorithm (introduced in section 2.3.1, in detail in 1) can be reformulated as a set of conv layers and multiple iterations of the algorithm can then be represented using a Recurrent Network. This section details this adaptation:

Initialize Q: $Q_i(l) \leftarrow \frac{1}{Z_i} exp(U_i(l))$ for all i;
while *not converged do* **do**
- Message passing
$\tilde{Q}_i^{(m)}(l) \leftarrow \sum_{j \neq i} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) Q_j(l)$ for all m
- Weighting filter outputs
$\check{Q}_i \leftarrow \sum_m w^{(m)} \tilde{Q}_i^{(m)}(l)$
- Compatibility transform:
$\hat{Q}_i \leftarrow \sum_{l' \in \mathcal{L}} \mu(l, l') \check{Q}_i(l')$
- Adding unary potentials
$\check{Q}_i(l) \leftarrow U_i(l) - (\hat{Q})_i(l)$
- Normalizing
$Q_i \leftarrow \frac{1}{Z_i}(\check{Q}_i(l))$

end

Algorithm 2: Mean field in fully connected CRFs as a stack of CNN layers [59]

Apart from the added steps, the new method is more general: instead of the traditional Potts models for the label comparability function, a custom function can be used, e.g. learned from the data.

2.3.3.1 Initialization

Before any update, the function $Q_i(l)$ needs to be initialized: $Q_i(l) \leftarrow \frac{1}{Z_i} exp(U_i(l))$, where $Z_i = \Sigma_i exp(U_i(l))$. This is simply a softmax function on the unary potentials of across all labels on every pixel, so at this stages it does not use neighbor information.

2.3.3.2 Image Passing

In a traditional dense CRF, image passing is performed via M Gaussian filters on Q values. These filter coefficients are based on pixel locations and RGB values and reflect how strongly a pixel is related to other pixels.

In [59] implementation, this is performed via a Permutohedral lattice implementation, with a $O(N)$ time, with N being the number of pixels in the image.

To apply backpropagation, the derivatives of the error regarding the inputs are calculated by sending the error's input value through the same Gaussian Filters in the inverse direction.

2.3.3.3 Weighting Filter Outputs

This step performs a weighted sum of the M filter outputs from the previous step for each class label.

If each class is considered individually, this can be seen as a $1x1$ convolution with M input channels and one output channel.

Backpropagation can also be performed in similar fashion of the Image Passing step.

2.3.3.4 Compatibility transform

The output of the previous iteration is shared between the labels, depending on their compatibility. The compatibility in this implementation is defined by the Potts models.

This step can be viewed as another $1x1$ convolution layer and the number of both input and output channels is L, the number of labels.

2.3.3.5 Adding Unary Potentials

In this step, the output of the compatibility transform step is subtracted element-wise from the unary inputs U.

2.3.3.6 Normalization

Finally, a Normalization is performed, by applying a softmax function.

2.3.4 General overview of CRF-RNN

This approach allows the construction of an end-to-end network, that has both the strengths of the CRF and the flexibility of a CNN, allowing it to be seamlessly integrated into any NN architecture.

The CNN stage performs pixel levels feature extraction, that is then followed by a prediction, taking into account the structure of the image.

In our case, a Fully Connected layer was used for calculating the overall image probability.

The final network will have three hyperparameters specific for the CRF-CNN implementation:

- θ_α degree of nearness required for appearance kernel
- θ_β associated
- θ_γ degrees of nearness required for the smoothness kernel
- number of iterations on each epoch to be performed by the algorithm.

2.4 Semantic Learning Machine

On the topic of CNN-based PCa classification, this work also explored a novel way to improve the model's performance.

Most of the contributions that try to improve CNN-based classification do so by focusing on the earlier layers of the network, and little attention is given to the last layers [53], responsible for the actual prediction. They typically form a very simple fully connected architecture and no thought is given to their design.

The idea explored in this work?s contribution was to improve the performance of the final model using a network generated by a neuroevolutionary algorithm, the Semantic Learning Machine, SLM [20].

The SLM constructs Neural Networks using hill-climbing, and not by relying on traditional backpropagation. The network definition actually occurs by means of a specially defined variation operator, a mutation operator that induces a unimodal fitness landscapes (i.e., without any local optima) [20].

In the original contribution, the SLM outperformed several NN algorithms: Neuroevolution of augmenting topologies (NEAT), Fixed-topology neuroevolution (FTNE), Multilayer Perceptron (MLP) and Support Vector Machines (SVM). SLM had the best performance in 24 out of 36 comparisons.

For our contribution, the SLM was used to build the neural network used to create a prediction, based on the features extracted by the CNN.

Chapter 3

Methods

This chapter highlights all the pieces necessary to execute this work, namely the dataset and data preparation steps employed or considered (section 3.1). In section 3.2 an introduction to the architectures used is provided. The chapter ends with the presentation of the proposed architecture, integrated with a CRF (subsection 3.2.2).

3.1 PROSTATEx Challenge 2017 data

The dataset used for this work was compiled at the Radboud University Medical Centre in Nijmegen, The Netherlands [29]. It was made available for the SPIE-AAPM-NCI Prostate MR Classification Challenge (PROSTATEx Challenge 2017) [3, 31]. It was compiled in-house for the purpose of developing and evaluating a CAD system under the supervision of Dr. Huisman [29].

The data contains multi-parametric images and corresponding lesion information for 344 patients. Of those 344, 204 comprise the training set and 140 the test set. Only the training set was considered because the test set did not have the target variable available.

The images were presented to an expert that identified regions in which he considered there could be cancerous cells present. In those regions of interest, a biopsy was performed and then analyzed. This information is considered the question of interest of the data, as well as the ground truth.

If the lesion had a biopsy Gleason score of 7 or higher, was considered Clinically Significant (CS).

For each lesion, the following information was available, provided in a comma-separated file.

Field	Description
ProxID	ProstateX patient identifier
fid	Finding ID
pos	Scanner coordinate position of the finding
ClinSig	Whether this is a clinically significant lesion or not (1 if so, 0 otherwise)

Table 3.1: Information available for a lesion

The last column, ClinSig is only available in the training set (204 patients), as the ground truth.

The data contains at least 5 images for each patient: T2-weighted (T2W), Proton Density-weighted (PDw), diffusion-weighted (DW) and Dynamic Contrast Enhanced (DCE) images in various planes. This data comes encoded in two formats: the Dynamic Contrast Enhanced image comes from a T1 weighted image encoded in two files, .mhd and .zraw. The remaining image modalities are provided in DICOM format.

Field	Description
ProxID	ProstateX patient identifier
fid	Finding ID
pos	Scanner coordinate position of the finding
WorldMatrix	Matrix describing image orientation and scaling
ijk	Image column (i),row (j), and slice (k) coordinates of the finding. Using the VTK/ITK/Python array convention, (0,0,0) represents the first column and first row of the first slice.
TopLevel	0 - Series forms one image 1 - set of series forming a 4D image NA - Series form one image, but part of a level 1 4D image
Spacing Between Slices	Scalar spacing between slices
VoxelSpacing	Vector with x, y, z spacing scalars
Dim	Vector with 4D dimensions of image
DCMSerDescr	Original DICOM series description
DCMSerNum	DICOM series number

Table 3.2: Information available for an image

An additional comma-separated file was made available with metadata for every image, as can be seen in table 3.2.

Further detailed metadata is available in the DICOM/ KTRANS encoded images, but because these details are not uniformly available, it was not considered further.

3.1.1 Descriptive analysis

3.1.1.1 Images

Every patient has the 3 modalities available, with T2w images captured in the three planes: sagittal, coronal and transverse. Because the data was collected on an ad-hoc

Measure	Value
Count	204
Max	114
Min	8
Mean	9.43
Std	9.43

Table 3.3: Description of exams per patient

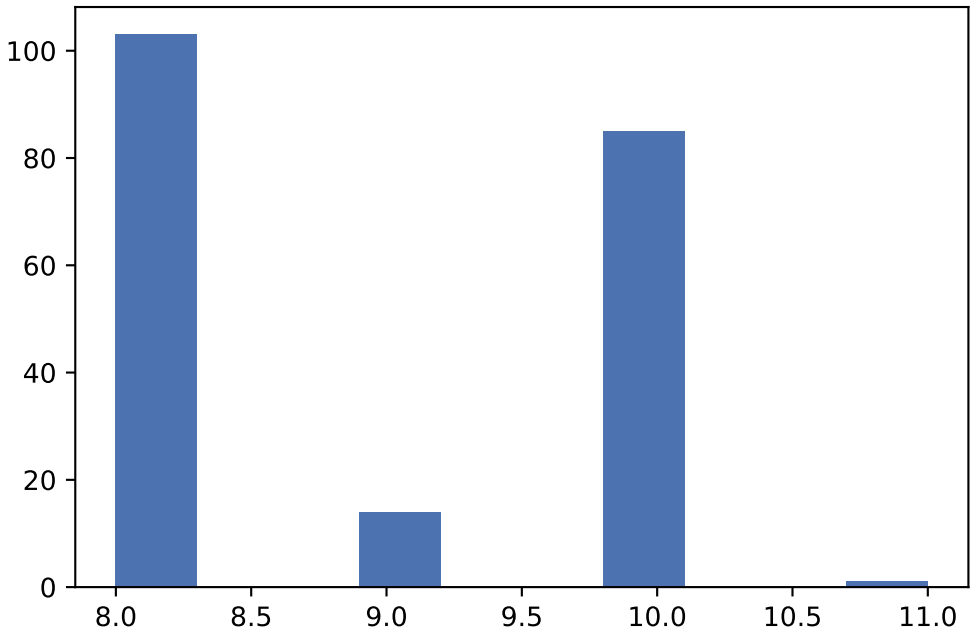


Figure 3.1: Number of exams per patient

basis, the patient may have other modalities available, as can be seen in table 3.3, but they were not considered.

It is possible to see that a patient has over 110 images available. These are probably DCE images collected at different time steps. Naturally, because this was the only patient with this information available, it could not be considered further.

In this work, only non-contrast enhanced images (T2w PD and ADC series) in the transverse plane were used, because DCE images present several disadvantages without evidence of impact in the model's predictive ability [9]. This was further discussed in section 2.1.1.5.

3.1.1.2 Lesion

A lesion correspondents to a region of interest where the technician suspects there could be prostate cancer present. As it originally stands, there are 359 lesions, of which 81 (22.6 %) are clinically significant - cancerigenous.

Measure	Value
Max	10.0
Min	1.0
Median	1.0
Average	1.768

Table 3.4: Lesion distribution per patient

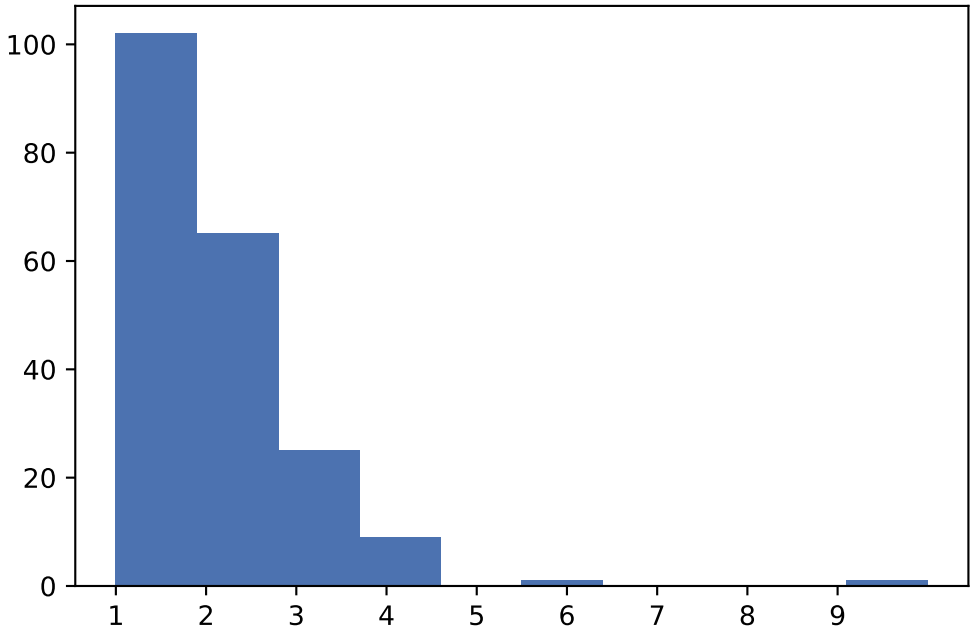


Figure 3.2: Number of lesions per patient

3.1.2 Data Processing

Being this a mMRI study, different modalities and images are available for each patient.

Therefore it is of the utmost interest to combine this information and use all of it when understanding the presence of PCa. This can be done, e.g. by assigning to each modality a channel in the final image.

In this section, the processing steps carried will be explained and given a short introduction. In this work, four steps were applied: interpolation, co-registration,

standardization and patch retrieval. Lastly, image augmentation was tried but ultimately not implemented. Outlier or anomaly removal methods were not considered.

3.1.2.1 Isotropic interpolation

The resolution of an image can be interpreted as the amount of information per pixel of that image. This means that an image with higher resolution will provide greater detail and it is easier to distinguish different tissues neighboring each other.

Depending on the collection conditions (e.g. different collection parameters) the different modalities will not have the same resolution in all three axes: a 3D image is actually not be perfectly 3D.

This can be analysed by verifying the slice thickness used: slice thickness is the distance between each slice. Higher thickness means that each slice has to carry more information - thus less detail and resolution, making images blurrier.

In the case of this dataset, the data is anisotropic: unequal slice thickness means unequal images resolution. The x- and y-axis (the short axes) have lower slice thickness (higher resolution) than the z-axis (long axis).

For example, the T2w images collected have a slice thickness of $Z = (0.5625 \;\; 0.5625 \;\; 3)$ mm on the x, z and y axes respectively. Therefore a pixel on the x axis represents $0.5625mm$ of tissue, same for the $y$axis. But the pixel on the y axis will contain $3mm$ of tissue.

Based on this information and the previous work of Liu et all [32], isotropic interpolation was performed.

The objective of this step is to create images that have the same resolution in all planes. The chosen slice thickness is of $Z* = (1 \;\; 1 \;\; 1)$ mm: each pixel will carry information at the resolution 1 mm of tissue. To achieve this cubic interpolation was performed, on the Dipy [13] package, a library implemented in Python that focuses on diffusion MRI analysis.

The process to carry interpolation requires access to the metadata of the image, provided in table 3.2, namely the World Matrix and the Voxel Spacing.

For example, if the original image had the dimensions of $S\begin{bmatrix}320 & 320 & 19\end{bmatrix}$ pixels, the new image will have the dimensions of $\frac{Z}{Z*}S = \begin{bmatrix}180 & 180 & 68\end{bmatrix}$. Now each pixel corresponds to 1 mm of tissue.

It is possible to see that the long axis carries less information (from a resolution of 0.5625 mm to 1 mm), but the short axis carries more (from 3 mm to 1 mm). To achieve the changes in image resolution, cubic interpolation was used.

3.1.3 Image co-registration

Two MRI sequences can not be simply overlayed, despite depicting the same location on the same patient: they may have the same features in different locations, additional noise, different representations, etc.

Several factors can be the cause of this: a patient may move during the examination; change in body water composition because the images were collected in different time periods, or the capture devices have different technical configurations.

To make the images comparable we need to match the images, making sure that their information overlap. This process is called co-registration.

Provided with two images, one is the reference (also called stationary) and the other is the moving. A set of transformations τ are applied to the moving image so that the landmarks and features on the moving image overlap the ones in the stationary image.

Registration is a pair-wise procedure, so it is done two images at the time. In this work, only affine transformations (i.e. rotation, translation, scaling and shear) were applied, and non-affine were not considered (e.g. distortion).

To measure the quality of the images matching, a quantitative metric needs to be defined. For this work, due to the prevalence in the literature and readily implementation in Dipy, Mutual Information (MM) was chosen [34].

The registration process was a pipeline of three registration methods: Center of Mass (CoM), affine and then rigid body (RB) registration, applied in this order.

The reference image was defined as the T2w image of each patient because it is the one that best depicts the prostate anatomy, as mentioned earlier. The moving images are the remaining modalities (i.e. PD and ADC).

Mutual Information Criterion The concept of Mutual Information is borrowed from information theory [34]. On its simplest idea, the Mutual Information Criterion (MIC) measures the amount of dependence between the pixels of two images and it is maximal if both are geometrically aligned. This is a good choice for image applications because it does not depend on à priori assumptions on the contents of the observation or on the nature of the studied dependence [34].

This criterion is defined by maximizing the joint and marginal probability distribution functions of both images.

For this to happen, the pixels values had to be discretized, into 32 bins specifically. Although much more computationally expensive, all pixels were used instead of a sample. The discretized values were then used to calculate the distribution functions.

Centre of Mass Registration This very simple method of registration starts by calculating the center of mass (CoM) of both images. Then a translation is applied to the moving image such that both CoM overlay. An example of the calculation of the center of an image is exemplified in [37], but the premise is to find the coordinates of an image that work as the fulcrum: where the mass (in the case of an image the sum of its pixel values) of an object is at equilibrium.

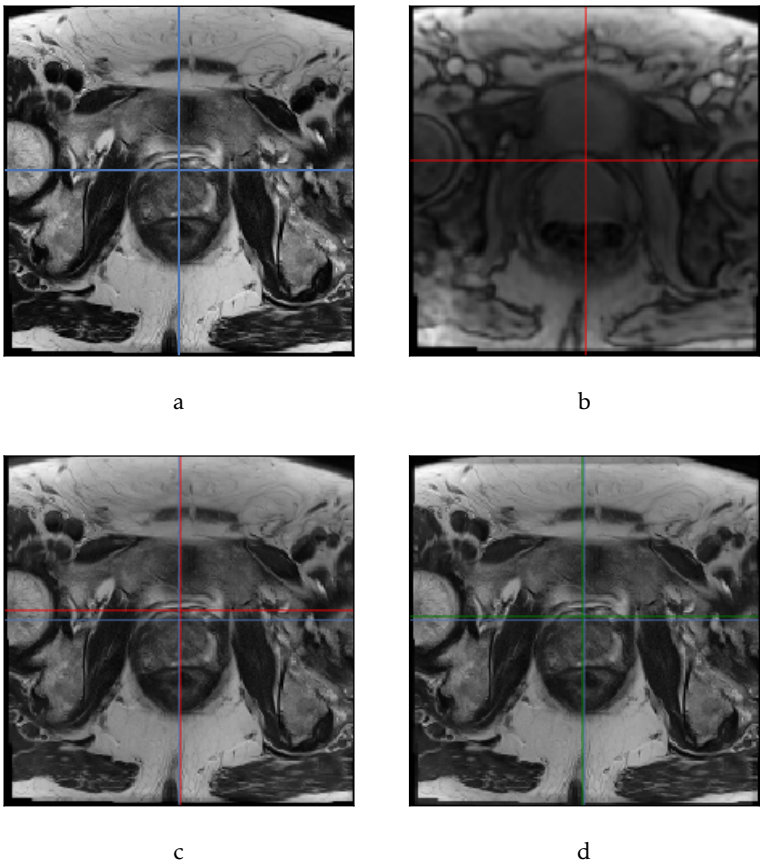
The horizontal centre of mass can be describe as:

$$R = \frac{\sum_{i \in I}(m_i \cdot i)}{\sum_{i \in I}(m_i)}$$

where I is the number of horizontal pixels and m is the sum of masses of each column. The mass is defined as the grey level value of each pixel.

The vertical CoM can be calculated by using the number of vertical pixels and the sum of masses of each row.

An applied example of CoM registration is shown in figure 3.3. Although this is a very basic approach, it is possible to see that a significant improvement is achieved.



a b c d

Figure 3.3: Centre of mass registration applied on two images of Patient 0001: T2 weighted is the reference image and Proton Density the moving.
a The reference image and the calculated centre of mass. b The moving image and the calculated centre of mass. c a and b overlayed, it is clear that there is a misalignment between the two images. d a (blue axis) and b (green axis) after a translation that overlaps both images centre of mass. The distance between both CoM is much smaller

Affine Registration Building on top of previous registration, affine registration was performed. This method applies affine image transformations to the moving image, and not just translation, like the CoM.

Affine registration assumes that not only the position of the prostate can change between images, but also it's geometry.

This method has two shortfalls: a) it is possible to get stuck in local optima and

b) it can be very computationally expensive. To tackle these issues, a multi-resolution Gaussian pyramid was implemented in Dipy, just as used by ANTS[7].

A Gaussian pyramid is composed of different representations of an image. Each layer is composed by a different scale (i.e. resolution and gaussian blur) of the image. In this work the pyramid has three layers (resolutions): the finest had no smoothing and the original resolution, the second had half the resolution and a smoothing factor of $\sigma = 1$ and the third layer a quarter of the original resolution and a smoothing sigma of $\sigma = 4$. This information is shown in table 3.5.

The affine registration was applied over a predefined number of iterations on the pyramid's lowest layer, then the result set of transformations (i.e τ matrix) was transferred to the next layer and re-tuned with its better resolution. This is done iteratively for every layer of the pyramid.

10000, 1000, 100 iterations were performed in the coarsest (lowest), medium and finest (highest) resolution, respectively.

As said, this is a multi-stage algorithm: the results of the previous iteration (i.e. previous layer of the Gaussian pyramid or previous iteration) are used as the input for the current iteration. An applied example of affine registration is shown in figure 3.4. Although not as noticeable such as when using CoM registration as the first step, it is possible to see improvements, namely in the upper right quadrant of the slice.

Layer	Resolution Factor	Gaussian smoothing factor σ	# iterations
1	1	0	100
2	2	1	1 000
3	4	3	10 000

Table 3.5: Gaussian pyramid parameters used for affine registration

Rigid Body Registration Rigid body registration (RB) assumes that the size and shape of the prostate are the same in both images. It only changes its position in space, through rotation and translation. Therefore, it can be considered a subset of affine transformations [6]. The major difference between CoM and RB is that RB iteratively tries to find the best set of transformations τ, although not necessarily the one that matches the image's CoM.

The results of applying RB registration are shown in figure 3.5.

3.1.3.1 Patch retrieval

The objective of this work is to detect if a set of coordinates in an image are CS or not.

Considering that an image may contain more than one lesion and that only the surrounding tissue should be relevant in determining if a lesion is or not cancerigenous, it does not make sense to use the whole image.

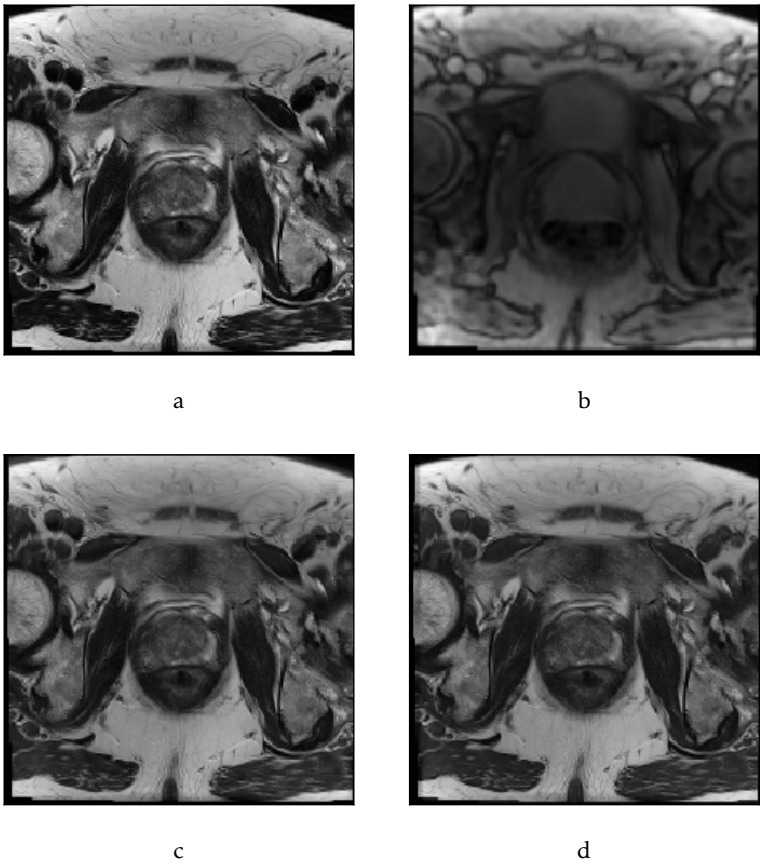


Figure 3.4: Affine registration applied on two images of Patient 0001: T2 weighted is the reference image and Proton Density the moving.
a The reference image. b The moving image. c a and b overlayed, it is apparent that there is a misalignment between the two images (blur is present). d a overlayed with b after affine registration. The blur appears smaller and the images quality higher, this is easier to see in the upper part of the image.

For the model, a patch was considered as the input corresponding to a lesion. A Region of Interest (RoI) was considered around the coordinates of each lesion and that area was extracted. For this purpose, the RoI considered is a buffer of 32 pixels in every direction with the lesion coordinates at it's the center.

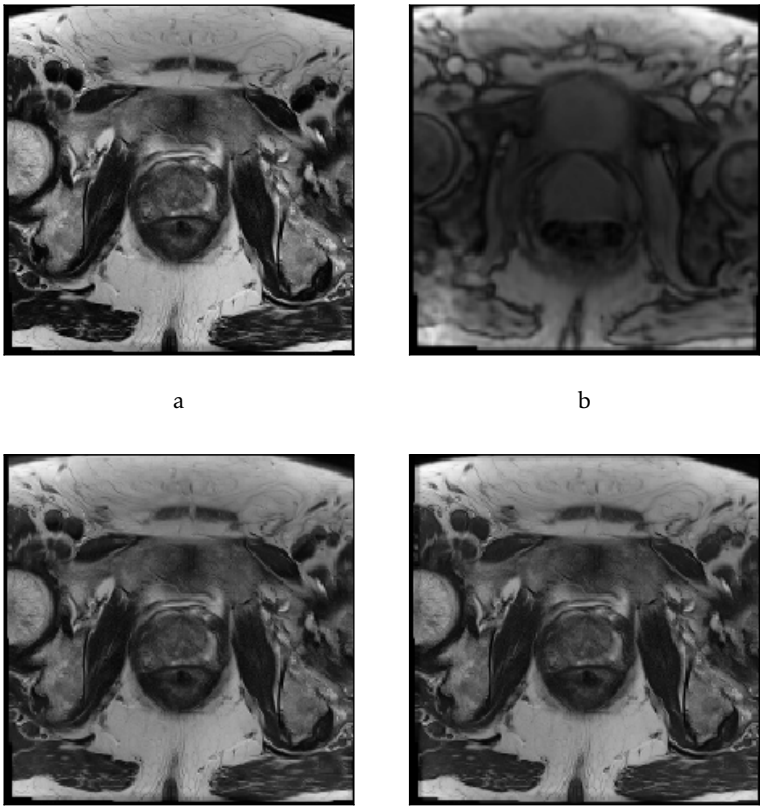
If a patch reached outside the image boundaries it was not considered.

A three-dimensional image of each lesion was composed of three channels: the T2w, PD, and ADC modalities of the lesion were assigned a channel, respectively.

3.1.3.2 Standardization

The last step was to center each image channel by subtracting it's mean and dividing by its standard deviation, leveling it to zero mean and unitary standard deviation:

$$X'_{ncij} = \Sigma_n^N \Sigma_c^C \Sigma_i^I \Sigma_j^J \frac{X_{ncij} - \mu_c}{\sigma_c}$$



a b c d

Figure 3.5: Rigid body registration applied on two images of Patient 0001: T2 weighted is the reference image and Proton Density the moving.
a The reference image. b The moving image. c a and b overlayed, it is apparent that there is a misalignment between the two images (blur is apparent). d a overlayed with b after rigid body registration.

$$\mu_c = \frac{\Sigma_n^N \Sigma_i^I \Sigma_j^J X_{ncij}}{NJK}$$

$$\sigma_c = \sqrt{\Sigma_n^N \Sigma_i^I \Sigma_j^J \frac{(X_{ncij} - \mu_c)^2}{NJK}}$$

This tries to approximate the pixels values to a normal distribution, $X' \sim N(\mu = 0, \sigma = 1)$.

This promotes faster convergence to the global optimum as well to prevent gradient explosion if the channels have very different scales.

In the case that the images channel standard deviation was 0, the image was not considered, as it did not have any diverse information.

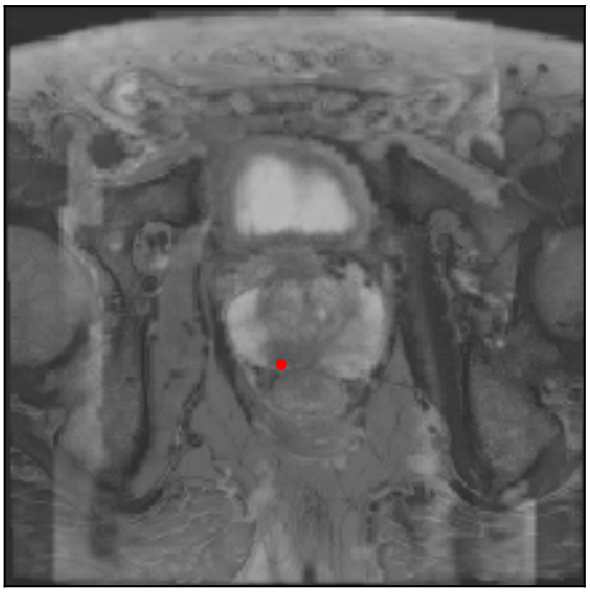


Figure 3.6: Concatenation of three MR images (T2W, PD and ADC) extracted of patient #29, after being interpolated and co-registered. The PCa coordinates are highlighted in red

3.2 Architectures

Several CNN architectures have been developed for image classification. The most common architectures (e.g. VGG, AlexNet) are general purpose and have been applied to various tasks. Some architectures have also been developed with specialized purposes in mind, although traditionally these specialized architectures evolve from well-established [12].

In the medical imaging field, both choices can be found in the literature. [47] used AlexNet, CifarNet, and GoogleLeNet for thoraco-abdominal lymph node (LN) detection and interstitial lung disease (ILD) classification, for example. Recently Liu et al [32] developed a VGG16 based architecture for the task of PCa classification in the context of PROSTATEx Challenge 2017, then XmasNet.

Architectures that have implemented CRF are traditionally found in the image segmentation tasks. CNN as feature extractor followed by a CRF classification phase have been used MIA segmentation, as introduced in 2.3.2.

To the best of my knowledge, no CRF-CNN based architectures for the purpose of image classification have been developed, regardless if applied to medical purposes or not.

This section is organized as follow: subsection 3.2.1 presents already established CNN architectures, that served as a state of art performance benchmark for this dataset. Subsection 3.2.2, presents the architecture developed for this task.

3.2.1 Convolutional Architectures

3.2.1.1 AlexNet

AlexNet [24] was the winner of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012, with a top-5 error of 15.3%. It was one of the first deep networks that used GPU acceleration to achieve reasonable training times, and it was this success that revived the interest of CNN's in computer vision.

According to the authors, AlexNet added several novelties to the CNN landscape:

- *ReLU Nonlinearity* Traditional non-linear activation functions are much slower to train than traditional functions. [24] showed that a network with ReLU achieves a similar error rate of 25% six times faster than an equivalent network with *tanh* activation function.

- *Multiple GPU training* By spreading the training into two GPUs (Graphical Processing Unit), it allowes to parallelize training (e.g., half of the neurons are assigned to each GPU).

- *Local Response Normalization*(RLN) ReLU have unbounded activations and LRN normalizes that. LRN encourages inhibition in neuron neighborhoods that have large responses and boosting in neighborhoods where a single neuron has a high response frequency.

- *Overlapping Pooling* A pooling operation serves to summarize information of a region. This operation has two parameters, a stride s and a filter of size zxz, as specified in section 2.2.2 . If $s = z$, then local pooling occurs, which is the traditional usage in CNN, but in AlexNet it is set to $s < z$. In this configuration, overlapping pooling is performed, where the input of a summarized region is shared with other regions. [24] found that networks with overlapping pooling had better chances of not overfitting while training.

The overall structure of the architecture is shown in figure 3.7 and the used parameters in table 3.6.

AlexNet is composed of eight learnable layers: five convolutional and three fully connected.

The five convolutional layers also have batch normalization and use ReLU activation functions. Of the five, three have Max Pooling performed on the outputs.

On the classification step, three fully connected layers are used on the outputs of the convolution step. The first two layers also have batch normalization and dropout to avoid overfitting. Like the convolutional layers, these have ReLU activation functions.

The last layer has just one neuron, using a sigmoid activation function. This serves to give a probability in the range $[0, 1]$.

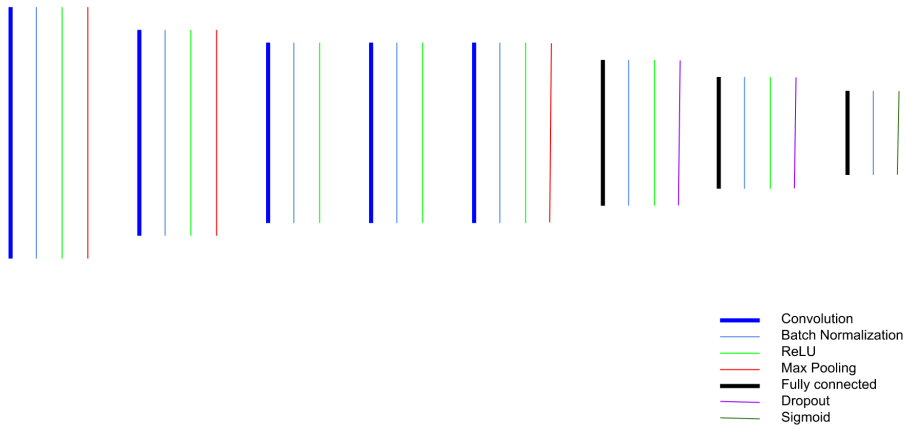


Figure 3.7: AlexNet architecture

Layers	Conv	MaxPooling	Conv	MaxPooling	Conv	Conv	Conv	MaxPooling	FC	Dropout	FC	Dropout	FC
Patch size Filter size # Neurons	11x11 x96	2x2	5x5 x256	2x2	3x3 x512	3x3 x512	3x3 x512	2x2	512	0.5	512	0.5	1

Table 3.6: AlexNet parameters

3.2.1.2 VGG16

VGG16 [48] was the first truly deep CNN, with 16 layers deep and achieved the first and second places in ILSVRC 2014.

After the success of the AlexNet [24], the focus of the deep learning community was to improve it, by tuning its configuration and structure [48].

The authors of [48] achieved the best improvement, by developing a much deeper network than the original AlexNet. While AlexNet has 5 convolutional layers, VGG16 has 16.

This depth increase was possible by using very small $3x3$ convolution filters.

A stack of two $3x3$ convolutional layers has a receptive field of $5x5$ and a stack of three has a $7x7$ receptive field.

The rationale of using a stack of three $3x3$ conv layers instead of a single $7x7$ layer, for example, is two-fold: first, by effectively using three convolutional layers, three non-linear rectification layers (ReLU) are used instead of just one - the decision function is more discriminative.

Second, and just as important, the number of parameters necessary to train is much lower. For a single three channel image, a stack three of $3x3$ convolutional layers has 243 learnable parameters; a single $7x7$ has 441.

Like the AlexNet, VGG16 uses ReLU activation functions wherever possible. A major difference though, is that does not use LRN.

The general structure of VGG16 can be seen in 3.8 and the parameters used for the convolutional and classification steps in tables 3.7 and 3.8, respectively, as provided by keras [10]. It is important to note that due to the high number of parameters in this network, adjustments were necessary, in order to be able to train the network. In the original architecture, the fully connected layers have both 4096 neurons. In this work, this number was reduced to 1024 in the first and 512 in the second layer.

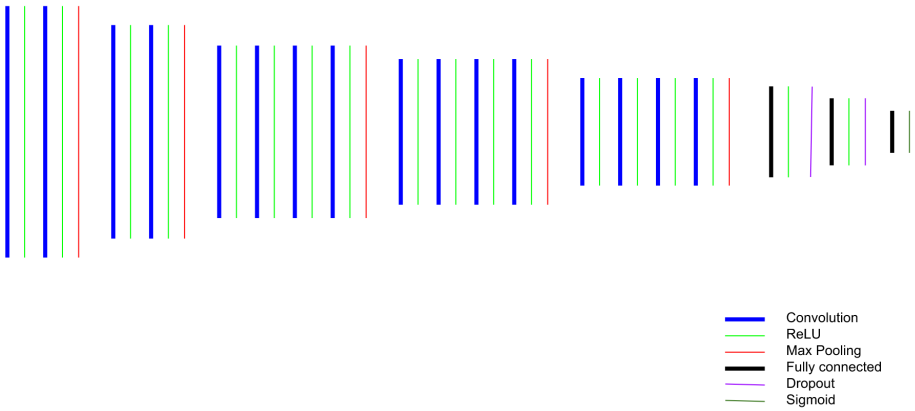


Figure 3.8: VGG16 architecture

Layers	Conv	Conv	Max	Conv	Conv	Max	Conv	Conv	Conv	Max	Conv	Conv	Conv	Max	Conv	Conv	Conv	Max
Patch size Filter size Neuron number	3x3 x64	3x3 x64	2x2	3x3 x128	3x3 x128	2x2	3x3 x256	3x3 x256	3x3 x256	2x2	3x3 x512	3x3 x512	3x3 x512	2x2	3x3 x512	3x3 x512	3x3 x512	2x2

Table 3.7: VGG16 - parameters of the convolutional layers.

Layers	FC1	Dropout	FC2	Dropout	FC3
Patch size Filter size Neuron number	1024	0.5	512	0.5	1

Table 3.8: VGG16 - parameters of the fully connected layers.

3.2.1.3 XmasNet

For the PROSTATEx Challenge 2017 [29], a promising network was developed, the XmasNet [32].

It is inspired by the VGG network and uses mMRI slices as multidimensional inputs as well. For the data preparation stage, the authors used isotropic interpolation and RB registration and data augmentation as well - 3D rotation and slicing.

It is an architecture that uses the traditional CNN operations: Convolution, Batch Normalization, Pooling and ReLu, in an structure presented in figure 3.9 and table 3.9.

It is organized in two convolution groups, each with two convolution blocks, formed by one convolutional layer, one batch normalization layer and one ReLU activation function. At the end of each unit, Max Pooling is applied.

For the classification step, two fully connected layers with ReLU activation are employed. To provide a probability, an activation function, softmax, is the very last layer of the architecture.

It achieved very good results in the competition, outperforming 33 participating groups and the second-highest AUC.

[32] also implements another interesting solution: using feature engineering to extract 87 features of each image and train a Random Forest model. These features reflect means and standard deviations of characteristics like intensities, textures, the contrast of the MRI, among others.

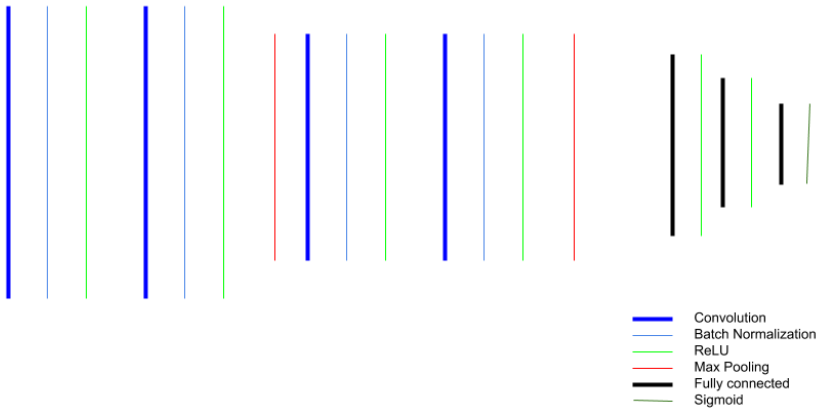


Figure 3.9: XmasNet architecture.

Layers	Conv1	Conv2	Max Pooling1	Conv3	Conv4	Max Pooling2	FC1	FC2	Softmax
Patch size / stride	3x3 / 1	3x3 / 1	2x2 / 2	3x3 / 1	3x3 / 1	2x2 / 2			
Output size	32x32 x32	32x32 x32	16x16 x64	16x16 x64	16x16 x64	8x8 x64	1024x1	256x1	2x1

Table 3.9: XmasNet parameters. Extracted from [32]

3.2.1.4 ResNet

Presented in the ILSVRC 2012, ResNet is arguably one of the most innovative frameworks in the field of computer vision and CNNs.

Layers	Conv2d	ConvBlock2a IdentityBlock2b IdentityBlock2c	ConvBlock3a IdentityBlock3b IdentityBlock3c	ConvBlock4a IdentityBlock4b IdentityBlock4c Identityblock4d Identityblock4e Identityblock4f	ConvBlock5a IdentityBlock5b IdentityBlock5c	FC1	FC2	FC3
Size	7x7x64	64x64x256	128x128x512	256x256x1024	512x512x2048	512	512	1

Table 3.10: ResNet50 parameters. Extracted from [10]

As the name implies, ResNet [16] presents a residual framework that eases the training of much deeper networks. That can be performed without such the problem of vanishing/exploding gradients or saturation and degradation of accuracy.

Instead of assuming that each set of layers will learn an underlying mapping, a defined residual mapping is presented, because the author hypothesis is that it is easier to fit the residual mapping than it is to learn an unreferenced mapping [16]. In practice, this is implemented with residual and skip connections.

With this in mind, the input of a layer is not only formed by the output of the previous layers, but also of the residual values of previous upstream layers - obtained through a residual connection - a layer will always learn something different than what is already encoded in its input. This also prevents information loss during the data-processing flow[11].

In theory, given large enough memory, networks composed of infinite stacks of layers can be trained. On the original paper, ResNet architectures formed by up to 152 layers - almost an order of magnitude higher than the number of layers in VGG16.

The architecture can be arbitrarily long, usually being defined as a set of convolution groups - composed by a convolutional layer with ReLU, pooling and batch normalization- and residual blocks - formed with the same setup of a convolutional group, but with a skipped connection. In figure 3.10 the 34-layer configuration is shown.

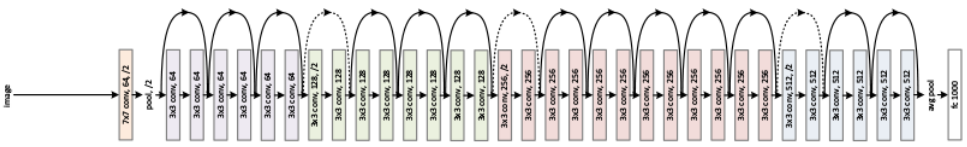


Figure 3.10: 34-layer ResNet configuration. Extracted from [16]

For performance considerations, the 50-layer architecture was used in this work, as provided by [10]. The parameters used for the convolutional layers are shown in table 3.10

3.2.2 CRF Architectures

3.2.2.1 CRF-XmasNet

A CRF performs a pixel-based prediction based on the class of neighboring pixels, and as discussed in 2.3, this approach provides significant performance in segmentation tasks when compared to traditional CNN networks because it promotes homogeneous regions of the same class.

The motivation of this work is to port this approach into the classification task.

A classification of whether a lesion is CS or not should not only have into consideration the majority prediction of each image's pixels but also if that prediction is cohesive in its region.

Therefore, instead of directly using each pixel's feature to perform a global classification, the CRF performs a feature extraction of the values of each pixel, and then passes that information to the NN that performs the final classification.

To execute this, XmasNet was chosen as the base architecture. It was chosen because of its relative simplicity and its previous success in this challenge. To adapt it to the the CRF format some changes in its architecture and configuration have been performed, shown in figure 3.11 and table 3.11, that can be compared to the original XmasNet in figure and table 3.9 and 3.9, respectively.

Even more emphasis on feature extraction was given, with the addition of three other conv layers. The original XmasNet had 4 conv layers, but with the increase in input complexity that the CRF can handle, this felt like a good compromise.

It is very important to note the inclusion of several skip connections in the architecture. In its adaptation of VGG16 to CRF-RNN, Liu et al [33] implemented it with good results, and several other architectures implemented it successfully. It has also been shown that skip connections smoothen the error landscapes [28] and allow the construction of deeper and more complex architectures [42] [16].

For refitting the CRF into a classification task, skip connections proved a necessary addition, to guarantee the network's training performance and reliability. Note that during the backpropagation process, the CRF-RNN expects to have an error value individually mapped to each input pixel it receives, but in the classification task this does not happen, as only one global classification label is possible. This is exacerbated by the presence of fully connected layers between the CRF and the final classification that further distort the error.

This was proven by empiric experience: the conv layers before the CRF would not experience any tuning because the error that would be propagated to would get diluted in the high number of parameters the CRF has. Early prototypes showed that the network final performance would be very similar to its initial - it would not improve.

Three skip connections were added, as shown in figure 3.11: The first connection injects some error information directly into the earlier conv layers, to prevent the

vanishing gradient problem. This also serves to directly transmit extracted features to the classifier. The second and third connections communicate high-level features to the CRF, to improve its feature extraction quality.

Resuming:

- XmasNet was used as the base architecture.
- A CRF-CNN layer [59] has been added between the convolutional layers and the fully connected ones.
- To ease the training procedure several skip connections were added.
- Due to performance considerations, the number of hidden units in the first FC layer and second FC layer were changed, from 1024 to 128 and 1024 to 256 respectively.
- A Dropout layer was added between the two fully connected layers.

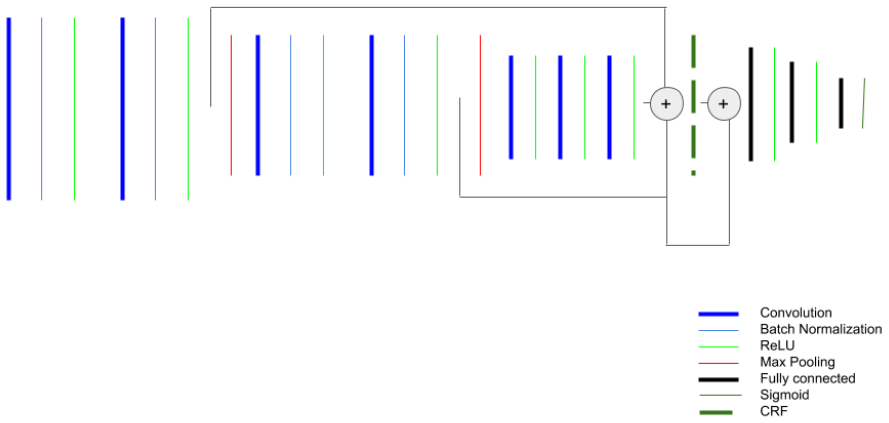


Figure 3.11: CRFXmasNet architecture.

Layers	Conv1	Conv2	Max Pooling1	Conv3	Conv4	Max Pooling2	CRF	FC1	Dropout	FC2	Softmax
Patch size / stride	3x3 / 1	3x3 / 1	2x2 / 2	3x3 / 1	3x3 / 1	2x2 / 2					
Output size	32x32 x32	32x32 x32	16x16 x64	16x16 x64	16x16 x64	8x8 x64	64x64	128x1	0.5	256x1	1x1

Table 3.11: CRFXmasNet parameters
f

CRFXmasNet uses the MFA in the CRF-RNN developed by [33], that defined a CRF model has a set of convolutional layers, shown in 2. For the MFA algorithm, 5

iterations are performed for each training batch, as suggested by the implementation authors [33].

A relevant constraint of the CRF-RNN implementation is that it only allows a batch size of 1.

Chapter 4

Results

Research question Can using a CRF as an intermediary step improve the performance of MRI classification tasks using CNNs?

4.1 Experimental setup

To answer this research question, an experimental framework was prepared with the aim of effectively and fairly testing each architecture against the dataset.

It was developed to tackle two known behaviors in CNN's:

- Hyperparameter sensitivity, meaning a small change in the network hyperparameters values can lead to large variations in its performance;
- Initialization - The network parameters (i.e. weights) are randomly initialized. Their capacity to converge largely depend on their initial values. Given the same hyperparameter configuration, convergence performance can vary, or it can even not be possible.

To address this behavior several configurations needed to be tested, and then trained several times, so that every model has a fair test, controlling for its initialization.

This is necessary, in order to achieve reliable and valid results from where one can draw conclusions.

The training framework was developed in Keras [10] and Tensorflow [1]; the training was performed in an Asus 550L laptop with 8GB of RAM, an Nvidia 840M GPU and an Intel i7-4510U 2.00GHz CPU; with a remote server using 1 Nvidia 1080ti GPU was also used. The training methodology was organized in the following manner:

1. Split the dataset into three partitions: training, validation and testing, each with 60%, 20% and 20% of the data, respectively.
2. For each architecture:
   a) Randomly select 20 random configurations (see 4.1.1).
   b) For each configuration, repeat 30 times:
      i. Train on the training and validation data, with Binary Cross Entropy as the loss function.
      ii. Train for 350 epochs OR until validation loss hasn't improved over 1×10^{-4} in the last 15 epochs.
      iii. Record the AUROC and Binary Cross Entropy metrics on the test data.
   c) Calculate the average performance for each configuration.
3. The configuration that has the lowest average test AUROC is selected as the best overall configuration for that architecture.

The training was performed in a batch size of 6 for the CNN architectures and of 1 for the CRF (reasons why explained in section 3.2.2.1).

4.1.1 Random search

To get a random hyperparameter configuration for each network, a random sample algorithm was employed, that generated a configuration out of the following possible values:

- Optimizer : $\{$SGD, Adam, RMSPROP$\}$
- Learning rate (lr): $\{1\times10^{-}1,\ 1\times10^{-}2,\ 1\times10^{-}3,\ 1\times10^{-}4,\ 1\times10^{-}5,\ 1\times10^{-}6,\ 1\times10^{-}7\}$
- Momentum (m): $\{0.0,\ 0.9,\ 0.99,\ 0.999\}$
- Decay (d): $\{0.0,\ 0.9,\ 0.99,\ 0.999\}$
- Nesterov (n): $\{0,\ 1\}$
- Amsgrad (a): $\{0,\ 1\}$
- CRF θ_α : $\{0.5,\ 1,\ 2,\ 3\}$
- CRF θ_β : $\{0.5,\ 1,\ 2,\ 3\}$
- CRF θ_γ : $\{0.5,\ 1,\ 2,\ 3\}$

It is important to note that not all parameters are relevant to every optimizer, as discussed in section 2.2.4 The parameters that each optimizer/layer can accept are shown in table 4.1.

Optimizer	lr	m	d	n	a	θ_α	θ_β	θ_γ
SGD	x	x	x	x				
Adam	x		x		x			
RMSPROPR	x		x					
CRFLayer						x	x	x

Table 4.1: Optimizer / hyperparameter compatibility

4.1.2 Metrics

To assess the quality of the models, a numerical value must be defined and calculated, to enable a quantitative comparison.

When dealing with an imbalanced problem traditional metrics fall short. An imbalanced classification problem is characterized by having one of the classes much more represented in the dataset than the other. Take for example am an extremely imbalanced problem, where only 1% of the observations are positive. If a model were to predict all observations presented to him as negative, he would achieve an accuracy of 99%.

If we think about certain applications such as this (i.e. predicting the presence of cancer): it is easy to see the problem of this approach: the cost of a false negative (e.g. predicting that a person does not have cancer when it has) is very high. On the other hand, if this behavior is taken to the extreme opposite and predicts every case as positive it is also far from the correct balance (e.g. cost to the medical system of doing unnecessary checkups, discomfort to the patient of being examined).

To take this into consideration, two metrics resistant to this problem have been chosen: Binary Cross Entropy and Area Under the ROC Curve (AUROC).

4.1.2.1 Binary Cross Entropy (BCE)

BCE was chosen as the training loss function when training the model because it promotes a balance between classifying every case as negative/positive. The role of the loss function is further discussed in 2.2.1.

As the name suggests, BCE uses the concept of entropy. In Information Theory, entropy in the measure of uncertainty associated with a given distribution $Q(y)$. In this case, $Q(y)$ n refers to the distribution of positive/negative cases.

As it is impossible to know this distribution - only a sample of it is available- the goals is to try to approximate a function $P(y)$ to it - created by our predictive models. BCE measures how closely this approximate distribution $P(y)$ follow the unknown original distribution $Q(y)$ - this measure is the cross-entropy of the two functions, as given by:

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))$$

Table 4.2 shows that BCE penalizes more predictions far from the true value, regardless of the true value (i.e. does not penalize false negatives more than false positives).

True	Predicted	Binary Cross Entropy
1	0.99	-0.010050
1	0.01	-4.605170
0	0.95	-2.995732
0	0.50	-0.693147
1	0.50	-0.693147
1	0.80	-0.223144
0	0.20	-0.223144

Table 4.2: Binary cross values for different theoretical real/predicted values.

4.1.2.2 Area Under ROC Curve (AUROC)

Area Under the ROC (Receiver Operating Characteristic) Curve promotes a balance, preventing the model from classifying everything as positive.

AUROC measures the relationship between True Positive Rate (TPR) (% of observations correctly predicted as positive) versus the False Positive Rate (FPR) (% of observations wrongly predicted as positive) as the classification threshold moves - the value over which a prediction probability is considered positive.

It provides a cost/benefit analysis on how strict should that threshold should be, ensuring a balance between false positives and false negatives.

The value of the AUROC is calculated by plotting the TPR on the y axis and the FPR x at various thresholds and calculating the area under that curve.

Ideally, that area is as big as possible (= 1), where all positive cases are correctly identified without wrongly identifying any negative cases.

For the calculation of this metric the implementation used was provided by the Scikit-learn package [43], written in Python.

4.1.3 Code implementation

To ease the prototyping capability, considerable time was spent in building a system written in Python that allows to easily add new modules and features. The source code is freely available in `https://github.com/plapa/prostatex-thesis`.

The code is separated into four major components:

1. *Data importation*, the data is imported to an SQLite database, registered and stored. There the lesions are linked to the corresponding images and all associated image metadata, including the history of performed preprocessing steps (i.e. image registration). This module is responsible by the patch retrieval and image merging, discussed in 3.1.2 .

2. *Feature creation* loads the data from the database and applies the necessary steps before creating the dataset, e.g. normalization/standardization and image augmentation if desired.

3. *Modelling*, arguably the most complex part, it orchestrates all the previous steps to guarantee that the results obtained are reproducible. It contains the models' definition, the random search, logging logic and is designed as barebone as possible to allow the code to be run on a shared server. To further ensure others can easily run it, the training is performed on a Docker container.

4. *Visualization and notebooks* Mostly a series of ad-hoc Jupyer notebooks to manually guarantee that the code logic is correct. Is also contains all the code that generates the images and tables used for this thesis and its related work.

4.2 Discussion

After training using the aforementioned methodology, the results were obtained and here discussed. The winning configuration for each architecture is shown in table 4.3, the performance results are shown in table 4.4, and the AUROC and BCE performance on the test set are shown in figures 4.2 and 4.1, respectively.

The best optimizer paints a very interesting picture: RMSPROP is chosen in the shallower architectures, like AlexNet and XmasNet, while Adam is chosen in deeper architectures like VGG16 and ResNet. This makes sense as Kinga et al [22] defend that Adam is an optimizer particularly well suited for large networks; and RMSPROP uses a moving average of the previous updates, leading to smaller updates, creating vanishing gradient problems in deeper networks.

The CRFXmasNet, the new architecture, appears in the third place when ordering by AUROC. Its potential is shown by seeing the improvement over the original XmasNet. It still lags behind VGG16 and ResNet, which are much more complex networks in terms of depth.

Regarding models performance, table 4.4 clearly shows that ResNet and VGG16 are the best performing architectures: they have the best values of the BCE and AUROC measures in the test data, respectively. It shows that the sheer complexity of these networks surpasses the marginal improvement of the addition of the CRF layer.

From these results, it is shown that AlexNet has a tendency to overfit in the training data, which leads to poor performance in unseen data.

The results also show that CRFXmasNet is very unreliable, being the architecture with the highest standard deviation across all metrics. The empirical experience shows that despite the addition of skip connections to facilitate backpropagation, the CRF is very dependent on its parameters initialization.

Almost all networks (i.e. AlexNet, ResNet, VGG16 and XmasNet) have very similar performance when comparing the BCE measure. AUROC shows a clearer distinction:

XmasNet and ResNet have similar performance; AlexNet and CRFXmasNet are somewhat better, but VGG16 is a clear outperformer.

Architecture	Optimizer	lr	m	d	n	a	θ_α	θ_β	θ_γ
AlexNet	RMSPROP	$1x10^{-5}$		0					
XmasNet	RMSPROP	$1x10^{-5}$		0					
VGG16	Adam	$1x10^{-4}$		0		0			
ResNet	Adam	$1x10^{-5}$		0		0			
CRFXmasNet	Adam	$1x10^{-4}$		0.99		0	3	0.5	3.0

Table 4.3: Best configuration of each architecture.

Architecture	Loss Train	Loss Val	Loss Test	AUROC Train	AUROC Val	AUROC Test
AlexNet	0.096 ± (0.032)	0.640 ± (0.034)	0.537 ± (0.023)	1.000 ± (0.000)	0.568 ± (0.044)	0.588 ± (0.051)
VGG16	0.474 ± (0.021)	0.512 ± (0.008)	0.481 ± (0.018)	0.729 ± (0.047)	0.553 ± (0.066)	0.707 ± (0.050)
CRFXmasNet	0.584 ± (0.100)	0.607 ± (0.125)	0.573 ± (0.102)	0.563 ± (0.156)	0.499 ± (0.190)	0.566 ± (0.186)
ResNet	0.496 ± (0.020)	0.549 ± (0.028)	0.528 ± (0.023)	0.658 ± (0.065)	0.471 ± (0.088)	0.520 ± (0.100)
XmasNet	0.500 ± (0.023)	0.545 ± (0.045)	0.540 ± (0.043)	0.622 ± (0.054)	0.508 ± (0.108)	0.517 ± (0.101)

Table 4.4: Best results for each architecture. Mean value with standard deviation in parenthesis, averaged over 30 iterations. BCE- lower is better, AUROC - higher is better.

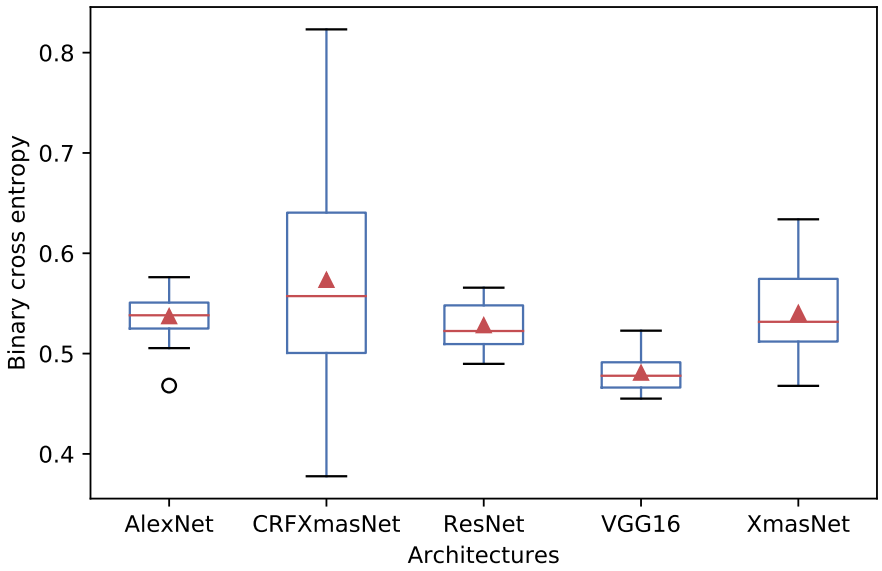


Figure 4.1: Binary cross entropy test set results.

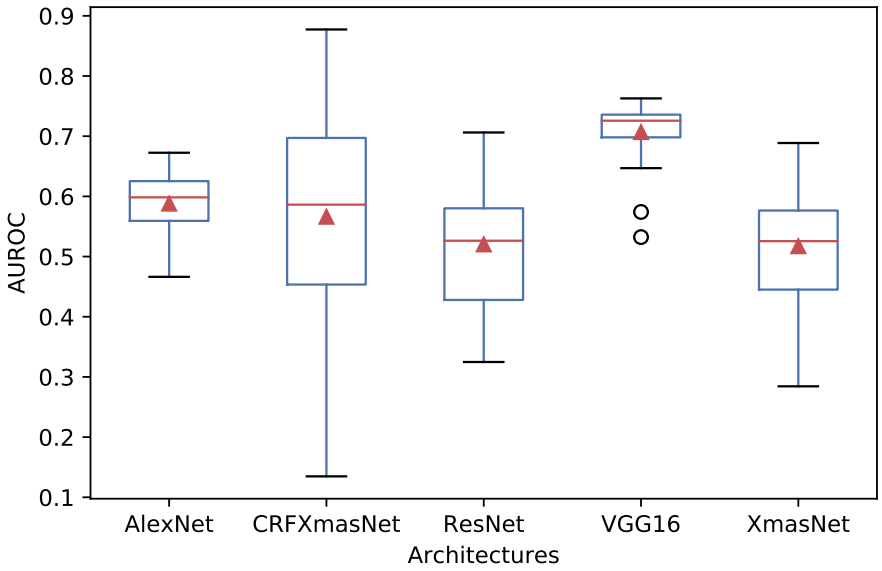


Figure 4.2: AUROC validation test set results.

4.3 Semantic Learning Machine Results

On the topic of CNN-based PCa classification, this work also explored a novel way to improve the model's performance, as introduced in section 2.4. This resulted in two contributions in academic venues: a poster in the Real World Applications (RWA) track [25] of the Genetic and Evolutionary Computation Conference (GECCO) and a paper in the Medical Applications of Genetic and Evolutionary Computation (MedGEC) workshop [26] of the same venue.

For our contribution, the SLM was used to build the neural network used to create a prediction, based on the features extracted by the CNN. To implement them a two-step method was used, as illustrated in Figure 4.3:

- The XmasNet network was trained end to end, to extract the features to be used and use the fully connected layers trained with backpropagation as a comparison.
- The fully connected layers were instead trained by a neuroevolutionary algorithm, namely the SLM.

The experimental results show that the SLM outperformed the XmasNet in terms of classification performance and training time, as the image 4.4 illustrates. The experimental results have been published to the academic community, as described earlier (see [26] and[25]).

Once again, to ensure the reliable and correct results, the training methodology was the same as described in section 4.1 of this chapter: each SLM architecture was trained 30 times.

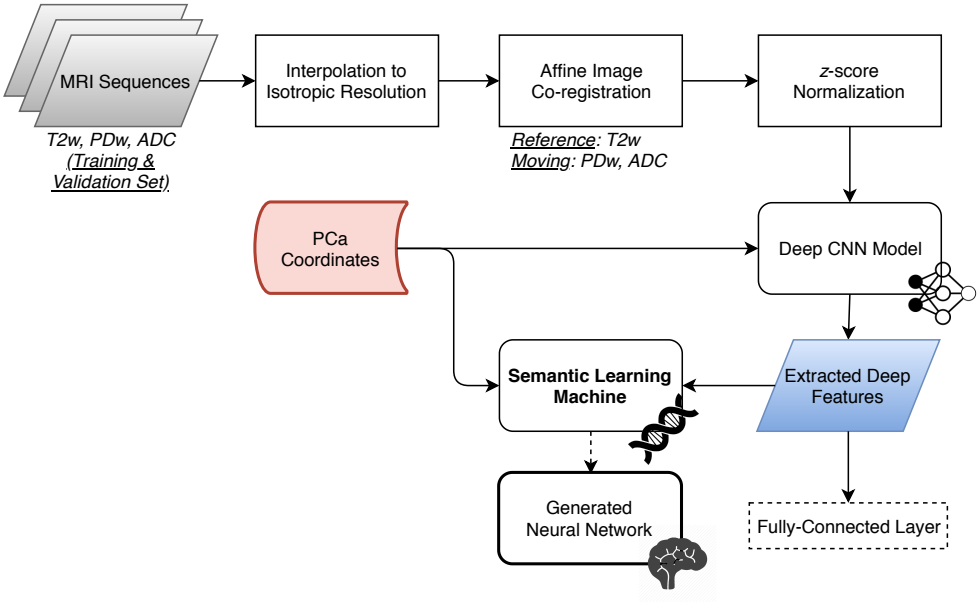


Figure 4.3: Workflow of the proposed neuroevolution approach based on the SLM

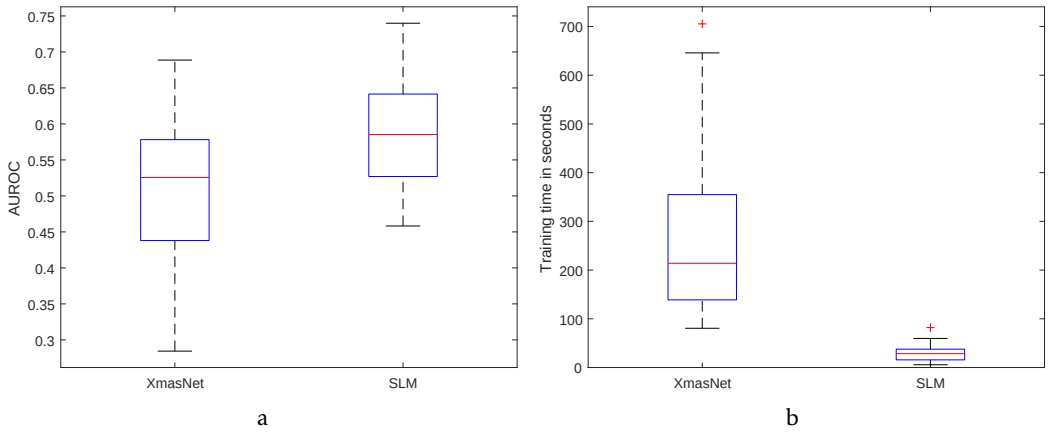


Figure 4.4: Performance comparison of SLM and XmasNet a AUROC achieved by SLM and XmasNet. b Training time required by XmasNet and SLM.

Chapter 5

Conclusions

Recently, computer systems have become an import aid in diagnosing cancer. In particular, deep learning techniques have shown to leverage relevant information extracted from Magnetic Resonance images [27].

The goal of this work was to exploit a method that has been used uniquely in the image segmentation task, i.e., Conditional Random Fields. Instead of their traditional role in image segmentation as a classifier, they were used as a feature extractor for the image classification task, added after the convolutional part of the neural network. Although traditionally very hard to train, CRFs can be redefined as a series of convolutional layers in a recurrent neural network [59]. This improvement significantly increases training speed and allows them to be integrated into a network and trained end-to-end.

Aiming at a quantitative quality measure, this work used a multi-parametric Magnetic Resonance Imaging dataset, provided for the PROSTATEx 2017 competition and collected by the Radboud University [29].

To assess the proposed design, the XmasNet architecture [32] was used, initially specifically tailored to this challenge. Starting from the initial XmasNet architecture, a CRF-RNN [59] module was integrated between the convolutional and fully connected portions.

A testing framework was employed, to evaluate the model's performance on the training data of the competition dataset. The CRFXmasNet results were compared against four state of the art architectures: AlexNet, VGG16, ResNet, and XmasNet.

The results are very promising, showing an increase in the network's classification performance when compared to the original architecture. Globally the new architecture ranked third when compared the AUROC (0.567) and BCE (0.53), even though it also showed unreliable performance and sensitivity to the initialization values.

Overall, as expected, the proposed solution is not yet at the level of the state of the art architectures, such as VGG16 and ResNet, since they are deeper. This could lead to an interesting future direction of this work, implementing the CRF feature extractor into these architectures.

The author would like to highlight two publications in peer-reviewed venues that have resulted of derivatives of this work (see [26] and[25]), which used the Semantic Learning Machine [20] to replace the Fully Connected Layers in the last part of the Convolutional Neural Networks.

Lastly, the author would like to highlight Mauro Castelli and Leonardo Rundo's supervision. Without their expertise, brains, ingenuity and most importantly, time and kindness, this work would not have been possible. Thank you.

Bibliography

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015.

[2] D Ampeliotis, A Antonakoudi, K Berberidis, E. Psarakis, and A Kounoudes. "A computer-aided system for the detection of prostate cancer based on magnetic resonance image analysis." In: *2008 3rd International Symposium on Communications, Control and Signal Processing*. IEEE. 2008, pp. 1372–1377.

[3] S. G. Armato, H. Huisman, K. Drukker, L. Hadjiiski, J. S. Kirby, N. Petrick, G. Redmond, M. L. Giger, K. Cha, A. Mamonov, et al. "PROSTATEx Challenges for computerized classification of prostate lesions from multiparametric magnetic resonance images." In: *J. Med. Imaging* 5.4 (2018), p. 044501. DOI: 10.1117/1.JMI.5.4.044501.

[4] A. Arnab, S. Zheng, S. Jayasumana, B. Romera-Paredes, M. Larsson, A. Kirillov, B. Savchynskyy, C. Rother, F. Kahl, and P. H. Torr. "Conditional random fields meet deep neural networks for semantic segmentation: Combining probabilistic graphical models with deep learning for structured prediction." In: *IEEE Signal Processing Magazine* 35.1 (2018), pp. 37–52.

[5] Y. Artan, D. L. Langer, M. A. Haider, T. H. Van der Kwast, A. J. Evans, M. N. Wernick, and I. S. Yetik. "Prostate cancer segmentation with multispectral MRI using cost-sensitive conditional random fields." In: *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. IEEE. 2009, pp. 278–281.

[6] J. Ashburner and K. J. Friston. "Rigid body registration." In: *Statistical parametric mapping: The analysis of functional brain images* (), pp. 49–62.

[7] B. B. Avants, N. Tustison, and G. Song. "Advanced normalization tools (ANTS)." In: (2009).

[8] F. Bray, J. Ferlay, I. Soerjomataram, R. L. Siegel, L. A. Torre, and A. Jemal. "Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries." In: *CA: a cancer journal for clinicians* 68.6 (2018), pp. 394–424.

[9] Y. J. Choi, J. K. Kim, N. Kim, K. W. Kim, E. K. Choi, and K. S. Cho. "Functional MR imaging of prostate cancer." In: *Radiographics* 27.1 (2007), pp. 63–75.

[10] F. Chollet et al. *Keras*. 2015.

[11] F. Chollet. *Deep Learning with Python*. 1st. Greenwich, CT, USA, 2017. ISBN: 1617294438, 9781617294433.

[12] B. J. Erickson, P. Korfiatis, T. L. Kline, Z. Akkus, K. Philbrick, and A. D. Weston. "Deep Learning in Radiology: Does One Size Fit All?" In: *Journal of the American College of Radiology* 15.3, Part B (2018). Data Science: Big Data Machine Learning and Artificial Intelligence, pp. 521 –526. ISSN: 1546-1440. DOI: https://doi.org/10.1016/j.jacr.2017.12.027.

[13] E. Garyfallidis, M. Brett, B. Amirbekian, A. Rokem, S. Van Der Walt, M. Descoteaux, and I. Nimmo-Smith. "Dipy, a library for the analysis of diffusion MRI data." In: *Frontiers in Neuroinformatics* 8 (2014), p. 8. ISSN: 1662-5196. DOI: 10.3389/fninf.2014.00008.

[14] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. 2016.

[15] V. Gulani, F. Calamante, F. G. Shellock, E. Kanal, S. B. Reeder, et al. "Gadolinium deposition in the brain: summary of evidence and recommendations." In: *Lancet Neurol.* 16.7 (2017), pp. 564–570. DOI: 10.1016/S1474-4422(17)30158-8.

[16] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[17] N. Ing, Z. Ma, J. Li, H. Salemi, C. Arnold, B. S. Knudsen, and A. Gertych. *Semantic segmentation for prostate cancer grading by convolutional neural networks*. 2018. DOI: 10.1117/12.2293000.

[18] S. Ioffe and C. Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." In: *CoRR* abs/1502.03167 (2015).

[19] J. G. Jacobs, E. Panagiotaki, and D. C. Alexander. "Gleason Grading of Prostate Tumours with Max-Margin Conditional Random Fields." In: *Machine Learning in Medical Imaging*. Ed. by G. Wu, D. Zhang, and L. Zhou. Cham, 2014, pp. 85–92.

[20] J.-B. Jagusch, I. Gonçalves, and M. Castelli. "Neuroevolution Under Unimodal Error Landscapes: An Exploration of the Semantic Learning Machine Algorithm." In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. GECCO '18. Kyoto, Japan, 2018, pp. 159–160. ISBN: 978-1-4503-5764-7. DOI: 10.1145/3205651.3205778.

[21] D. Junker, F. Steinkohl, V. Fritz, J. Bektic, T. Tokas, F. Aigner, T. R. Herrmann, M. Rieger, and U. Nagele. "Comparison of multiparametric and biparametric MRI of the prostate: are gadolinium-based contrast agents needed for routine examinations?" In: *World J. Urol.* (2018), pp. 1–9. DOI: 10.1007/s00345-018-2428-y.

[22] D. P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization*. 2014.

[23] P. Krähenbühl and V. Koltun. "Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials." In: (Oct. 2012).

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." In: *ImageNet Classification with Deep Convolutional Neural Networks*. 2012. ISBN: 9780415468442. DOI: 10.1061/(ASCE)GT.1943-5606.0001284.

[25] P. Lapa, I. Gonçalves, L. Rundo, and M. Castelli. "Enhancing Classification Performance of Convolutional Neural Networks for Prostate Cancer Detection on Magnetic Resonance Images: a Study with the Semantic Learning Machine." In: *Proceedings of the Genetic and Evolutionary Computation Conference 2019*. GECCO'19. New York, NY, USA, 2019.

[26] P. Lapa, I. Gonçalves, L. Rundo, and M. Castelli. "Semantic Learning Machine Improves the CNN-Based Detection of Prostate Cancer in Non-Contrast-Enhanced MRI." In: *Proceedings of the Genetic and Evolutionary Computation Conference 2019*. GECCO'19. New York, NY, USA, 2019. DOI: 10.1145/3319619.3326864.

[27] G. Lemaître, R. Martí, J. Freixenet, J. C. Vilanova, P. M. Walker, and F. Meriaudeau. "Computer-Aided Detection and diagnosis for prostate cancer based on mono and multi-parametric MRI: A review." In: *Computers in Biology and Medicine* 60 (2015), pp. 8–31. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2015.02.009.

[28] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein. "Visualizing the Loss Landscape of Neural Nets." In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. 2018, pp. 6389–6399.

[29] G. Litjens, O. Debats, J. Barentsz, N. Karssemeijer, and H. Huisman. "Computer-Aided Detection of Prostate Cancer in MRI." In: *IEEE Transactions on Medical Imaging* 33.5 (May 2014), pp. 1083–1092. ISSN: 0278-0062. DOI: 10.1109/TMI.2014.2303821.

[30] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sánchez. "A survey on deep learning in medical image analysis." In: *Medical Image Analysis* 42 (2017), pp. 60 –88. ISSN: 1361-8415. DOI: 10.1016/j.media.2017.07.005.

[31] G. Litjens, O. Debats, J. Barentsz, N. Karssemeijer, and H. Huisman. *"PROSTATEx Challenge data", The Cancer Imaging Archive*. Online; Accessed on January 25, 2019. 2017. DOI: 10.7937/K9TCIA.2017.MURS5CL.

[32] S. Liu, H. Zheng, Y. Feng, and W. Li. "Prostate Cancer Diagnosis using Deep Learning with 3D Multiparametric MRI." In: *CoRR* abs/1703.04078 (2017).

[33] T. Liu, X. Huang, and J. Ma. "Conditional Random Fields for Image Labeling." In: *Mathematical Problems in Engineering* 2016 (Apr. 2016), pp. 1–15. ISSN: 1024-123X. DOI: 10.1155/2016/3846125.

[34] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. "Multimodality image registration by maximization of mutual information." In: *IEEE Transactions on Medical Imaging* 16.2 (Apr. 1997), pp. 187–198. ISSN: 0278-0062. DOI: 10.1109/42.563664.

[35] W. Mangrum, K. Christianson, S. Duncan, P. Hoang, and A. S. " *Duke Review of MRI Principles*. first. Case Review Series. 2012.

[36] H. Masaoka, H. Ito, A. Yokomizo, M. Eto, and K. Matsuo. "Potential overtreatment among men aged 80 years and older with localized prostate cancer in Japan." In: *Cancer Science* 108.8 (2017), pp. 1673–1680. DOI: 10.1111/cas.13293.

[37] I. Mcmanus, K. Stöver, and D. Kim. "Arnheim's Gestalt Theory of Visual Balance: Examining the Compositional Structure of Art Photographs and Abstract Images." In: *i-Perception* 2 (Oct. 2011), pp. 615–47. DOI: 10.1068/i0445aap.

[38] D. McRobbie, E. Moore, M. Graves, and M. Prince. *MRI from Picture to Proton*. 2007. ISBN: 9780521683845.

[39] J Monaco, J Tomaszewski, M Feldman, M. Moradi, P. Mousavi, A. Boag, C. Davidson, P. Abolmaesumi, and A. Madabhushi. "Detection of prostate cancer from whole-mount histology images using Markov random fields." In: Citeseer.

[40] K. Möllenhoff, A.-M. Oros-Peusquens, and N Shah. "Introduction to the Basics of Magnetic Resonance Imaging." In: vol. 71. July 2011, pp. 75–98. ISBN: 978-1-61779-988-4. DOI: 10.1007/7657_2012_56.

[41] A. Y. Ng and M. I. Jordan. "On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes." In: *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*. NIPS'01. Vancouver, British Columbia, Canada, 2001, pp. 841–848.

[42] A. E. Orhan and X. Pitkow. *Skip Connections Eliminate Singularities*. 2017.

[43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. "Scikit-learn: Machine Learning in Python." In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[44] D. C. Preston. *Magnetic Resonance Imaging (MRI) of the Brain and Spine: Basics*. Nov. 2006.

[45] H. Sadeghi-Gandomani, M. Yousefi, S Rahimi, S. Yousefi, A Karimi-Rozveh, S Hosseini, A. Mahabadi, H. Abarqui, N. Borujeni, and H Salehiniya. "The Incidence, Risk Factors, and Knowledge About the Prostate Cancer through Worldwide and Iran." In: *World Cancer Research Journal* 4.4 (2017).

[46] D. Shen, G. Wu, and H.-I. Suk. "Deep Learning in Medical Image Analysis." In: *Annual Review of Biomedical Engineering* 19.1 (2017), pp. 221–248. DOI: 10.1146/annurev-bioeng-071516-044442.

[47] H. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. J. Mollura, and R. M. Summers. "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning." In: *CoRR* abs/1602.03409 (2016).

[48] K. Simonyan and A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition." In: *CoRR* abs/1409.1556 (2014).

[49] A. C. Society. "Prostate Cancer Early Detection, Diagnosis, and Staging." In: *American Cancer Society* (May 2016).

[50] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." In: *Journal of Machine Learning Research* 15 (2014), pp. 1929–1958.

[51] A. Stangelberger, M. Waldert, and B. Djavan. "Prostate Cancer in Elderly Men." In: *Reviews in Urology* 10.2 (2008), p. 111.

[52] C. Sutton, A. McCallum, et al. "An introduction to conditional random fields." In: *Foundations and Trends® in Machine Learning* 4.4 (2012), pp. 267–373.

[53] Z. Tian, L. Liu, Z. Zhang, and B. Fei. "PSNet: prostate segmentation on MRI based on a convolutional neural network." In: *J Med Imaging (Bellingham)* 5.2 (Apr. 2018). 29376105[pmid], pp. 021208–021208. ISSN: 2329-4302. DOI: 10.1117/1.JMI.5.2.021208.

[54] M. N. N. To, D. Q. Vu, B. Turkbey, P. L. Choyke, and J. T. Kwak. "Deep dense multi-path neural network for prostate segmentation in magnetic resonance imaging." In: *Int J Comput Assist Radiol Surg* 13.11 (Nov. 2018). 30088208[pmid], pp. 1687–1696. ISSN: 1861-6429. DOI: 10.1007/s11548-018-1841-4.

[55] P. S. Tofts. "T1-weighted DCE imaging concepts: modelling, acquisition and analysis." In: *signal* 500.450 (), p. 400.

[56] B. Turkbey, A. M. Brown, S. Sankineni, B. J. Wood, P. A. Pinto, and P. L. Choyke. "Multiparametric prostate magnetic resonance imaging in the evaluation of prostate cancer." In: *CA: A Cancer Journal for Clinicians* 66.4 (2016), pp. 326–336. DOI: 10.3322/caac.21333.

[57] P. C. Vos, J. O. Barentsz, N Karssemeijer, and H. J. Huisman. "Automatic computer-aided detection of prostate cancer based on multiparametric magnetic resonance image analysis." In: *Physics in Medicine and Biology* 57.6 (Mar. 2012), pp. 1527–1542. DOI: 10.1088/0031-9155/57/6/1527.

[58] X. Zhao, Y. Wu, G. Song, Z. Li, Y. Fan, and Y. Zhang. "Brain tumor segmentation using a fully convolutional neural network with conditional random fields." In: *International Workshop on Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Springer. 2016, pp. 75–87.

[59] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. "Conditional Random Fields as Recurrent Neural Networks." In: *The IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1529–1537.