UNIVERSITÀ
DEGLI STUDI
FIRENZE

PhD Program in Smart Computing
Dipartimento di Ingegneria dell'Informazione (DINFO)

Siamese and Recurrent neural networks for Medical Image Processing

Alberto Rossi

Dissertation presented in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Smart Computing

PhD Program in Smart Computing
University of Florence, University of Pisa, University of Siena

Siamese and Recurrent neural networks for Medical Image Processing

Alberto Rossi

Advisor:

Prof. Franco Scarselli
Prof.ssa Monica Bianchini

Head of the PhD Program:

Prof. Paolo Frasconi

Evaluation Committee:
Prof. Bram Van Ginneken, *University of Radboud*
Prof. Pietro Liò, *University of Cambridge*

XXXIII ciclo — October 2020

To my family

Acknowledgement

I would like to thank my supervisors Franco and Monica, for believing in me, supporting me in my studies for the past three years and, last but not least, for the time they have devoted to correcting my English writing style. I have to acknowledge the effort that my family made in supporting me, especially during my time abroad in the Netherlands, but also during the entire PhD period. I have worked very pleasantly with all my colleagues at SAILab and I hope they will remember me at least for some funny stories I told during lunches or breaks. It is a pleasure also to mention and thank Radboud UMC colleagues and supervisor Henkjan Huisman, in the Netherlands, for their hospitality and their ability to make me feel at home. I also like to say thanks to Markus and Ah Chung for guiding me on an intriguing project, always believing in me and my work. Finally, I have also to give credits to the supervisory committee for following me during the three year of the PhD, providing relevant feedback on my work. At last, I would like to acknowledge the Tuscany Region for the opportunity given to me to know this very interesting research field through the Pegaso scholarship they gave me.

Abstract

In recent years computer vision applications have been pervaded by deep convolutional neural networks (CNNs). These networks allow practitioners to achieve the state of the art performance at least for the segmentation and classification of images and in object localization, but in each of these cases the obtained results are directly correlated with the size of the training set, the quality of the annotations, the network depth and the power of modern GPUs.

The same rules apply to medical image analysis, although, in this case, collecting tagged images is more difficult than ever, due to the scarcity of data — because of privacy policies and acquisition difficulties — and to the need of experts in the field to make annotations.

Very recently, scientific interest in the study and application of CNNs to medical imaging has grown significantly, opening up to challenging new tasks but also raising fundamental issues that are still open. Is there a way to use deep networks for image retrieval in a database to compare and analyze a new image? Are CNNs robust enough to be trusted by doctors? How can small institutions, with limited funds, manage expensive equipments, such as modern GPUs, needed to train very deep neural networks?

This thesis investigates many of the issues described above, adopting two deep learning architectures, namely siamese networks and recurrent neural networks. We start with the use of siamese networks to build a Content–Based Image Retrieval system for prostate MRI, to provide radiologists with a tool for comparing multi–parametric MRI in order to facilitate a new diagnosis. Moreover, an investigation is proposed on the use of a composite loss classifier for prostate MRI, based on siamese networks, to increase robustness to random noise and adversarial attacks, yielding more reliable results. Finally, a new method for intra–procedural registration of prostatic MRIs based on siamese networks was developed.

The use of recurrent neural networks is then explored for skin lesion classification and age estimation based on brain MRI. In particular, a new devised recurrent architecture, called C–FRPN, is employed for classifying natural images of nevis and melanomas allowing good performance with a reduced computational load. Similar conclusion can be drawn for the case brain MRI, where 3D images can be sliced and processed by recurrent architectures in an efficient though reliable way.

Contents

Chapter 1

Introduction

Machine learning and, more recently, deep learning pervade lives of world citizens. The interest of the scientific community and companies in data science and automatic learning algorithms is largely increased aiming to speed up processes or to answer totally new questions. The reason of this interest is doubtless due to the incredible performance achieved by modern machine learning models in a huge variety of tasks, such as recommender systems, machine translation, speech recognition, image classification and segmentation, object detection and many others. Among all machine learning algorithms available in literature, the main trend is to use neural networks, due to their superior performance. This leadership was recently strengthened by their increased depth and complexity, in terms of network parameters. In the field of image analysis, it is emblematic how these models are grown in the last decade, starting from the first convolutional neural networks (CNNs), which had less than $100k$ parameters, to arrive at the actual state of the art models having millions of parameters.

To explain why such models are so powerful and trendy, we can identify at least three factors: the availability of huge data collections, the accessibility of very powerful computational resources as GPUs and TPUs, and the great advancement of research in machine learning.

In fact, machine learning models need data to learn a specific function (i.e. a task). The general idea is that the model gradually adapts itself by a learning mechanism that evaluates the error done by the model trying to predict the current pattern. It is easy to understand that a good model has to learn from a very large set of data to be sufficiently general to respond correctly to every input. This is one of the reasons why companies and researchers have begun to collect ever larger datasets. Nowadays very big data collections exist for a wide variety of tasks, and sometimes they are publicly available and used as benchmarks.

A great help to quickly process the vast amount of data required to have a proper learning comes from the increased level of computational power of modern GPUs

and TPUs. The parallel computing provided by these architectures drastically reduces the time required to train very complex models. Thanks to these two important assets, one of the main class of machine learning models nowadays is represented by deep learning algorithms, where the term deep denotes the high number of layers composing the network, resulting in a very relevant amount of parameters that must be trained (order of millions). Finally, advances in research go in the direction of having more effective learning methods, deviating from the first idea of just adding layers and parameters, which frequently lead to generalization problems.

The recurrent neural network model is particularly effective thanks to its property of sharing the same set of weights for all the elements composing the sequences. A similar behaviour is exploited by convolutional neural networks that, nowadays, compose most of the automatic systems for image analysis. In this case, a set of convolutional filters are learned and used in a sliding windows fashion to cover the entire image, extracting relevant features. But even with the reduction of weights guaranteed by the use of CNNs, by stacking several layers, the generalization power of the network can decrease unless we use some tricks to aid learning.

Unfortunately, the two fundamental assets described above (data and computational resources) could represent a big limitation for many deep learning enthusiasts. For some tasks, acquiring a relevant quantity of data is not easy. The main problem is due to the long time and the high cost required by this procedure, affecting mostly small companies with reduced budgets. Another major drawback arises from the need of annotating data to be used in the supervised learning framework. It may happen that experts are required to produce appropriate labels, increasing costs and time spent. In particular, the two described limitations fit the case of medical image analysis. In this field, acquiring a large set of images is problematic for multiple reasons. The first is related to the scarcity of data, because it is rare that a medical center processes thousands of patients for a pathology in a short period, while grouping images coming from different hospitals could be difficult due to different acquisition systems and methods. Secondly, there are often problems with privacy policies and it may happen that patients do not allow researchers to work on their images, reducing the possible amount of acquired images. Other problems can lie in the storage of high volumes of data in the hospital. Moreover, the needs for a precise data annotation can take time for the doctors, reducing their availability.

Problems with budget can arise also from the computational resources required to train large models. Not only the price but also the maintenance of such powerful systems can be problematic for small companies without a dedicated ICT team.

Another typical problem in automatic processing of medical images is related to the physician perception of such models. Indeed, physicians need to understand the support provided by the model: a black–box algorithm is not sufficient. Moreover, they would receive predictions from automatic tools in which they can highly trust,

whereas problems with the robustness are more than actual. The price to pay for an error can have dramatic effect on patients.

From an application point of view, this thesis focuses on tools that can limit the above mentioned problems related to the use of deep learning for medical image analysis. More precisely, the focus is on tools to support the decision process of clinicians, with a particular attention to improving the robustness of classifiers and increasing the efficiency of the models.

In fact, a way to support clinicians in the decision-making process is to provide a system for retrieving cases that are similar to the examined one. In this way, doctors can compare cases and directly assess the similarity with past exams. The exploitation of this comparison is particularly useful to calibrate diagnoses and treatments, moving toward a precision medicine approach. Such a framework can be implemented by a Content–Based Image Retrieval systems (CBIR), capable of retrieving the most similar images to a query one. In this thesis, we propose a novel CBIR designed to retrieve MR prostate images that look similar not just in appearance but also by the lesion severity point of view.

Moreover, the thesis studies techniques to improve the robustness of image classifiers. This can play an important role, increasing the doctors confidence in automatic models, and avoiding many misdiagnoses, with a wide saving in terms of complications for patients and money for the hospitals.

Finally, the thesis will study models able to work with reduced datasets and computing resources. This property can bring artificial intelligence to companies with reduced funds, opening new horizons for research.

From the machine learning point of view, the focus of the thesis is on two architectures, siamese and recurrent neural networks. These two architectures play an important role in modern deep learning. We carried out studies on them to understand their properties and derive specialized versions. Then, using these two models we obtained all the tools mentioned above for medical image analysis.

Actually, the siamese neural network is a particular architecture having two input streams designed to compare two patterns and assessing whether they are similar or not. By using siamese neural networks, we can implement CBIRs. Moreover, we also demonstrate how similarity learning can be used to improve the robustness of a classifier and, finally, it turn out to be particularly efficient for the case of small datasets. Instead, recurrent neural networks are designed to operate on sequences of data and adopt a mechanism, called weight sharing, to reduce the complexity of the model. We show how recurrent neural networks can implement alternative tools for learning with few computational resources and based on small datasets.

Thesis summary

The thesis is organized in two parts, collecting the results obtained with siamese and recurrent neural networks, respectively.

Initially, we describe the use of siamese neural networks for the implementation of a CBIR and, more precisely, we will apply the CBIR to the retrieval of prostate Magnetic Resonance Images (MRIs). Such a CBIR is able to consider similarities both with respect to the visual appearance and to the degree of severity of the represented lesion. In fact, this approach is new and can help radiologists compare cases and get a general picture of the case at hand. The siamese network is trained to understand the lesion severity correspondence expressed by the common radiological guideline (PIRADS score (Weinreb et al., 2016)), embracing the diagnostic similarity property.

The second result proposed regards a method to improve the robustness of a classifier based on convolutional neural networks. Such an approach uses siamese neural networks to make the deep classifier robust to random noise and to adversarial attacks. The network is trained based on combining the cross–entropy with the contrastive loss. The method has been successfully applied to the prostate lesion classification.

The third contribution is a new method by which siamese networks are used for implementing the registration of the intra–procedural prostate MR images. The goal of intra–procedural registration is that of aligning the image acquired during surgery to that acquired in a previous stage, allowing for an online targeting of the lesion. The proposed method consists of two main parts. First, intra–procedural MRIs are randomly augmented producing a set of candidates to be the best registered images with respect to the pre–procedural one. Then, a siamese network is trained to measure the similarity between pre– and intra–procedural candidates. Our method proves to be particularly efficient in the case of very small datasets.

The second part of the thesis is dedicated to recurrent neural networks. In particular, we studied a novel model that we call Convolutional Fully Recursive Perceptron (C–FRPN). The C–FRPN is a combination of recursive neural networks and convolutional networks, obtained by merging an architecture usually adopted for sequences with another popularly used for images. We show that C–FRPNs can outperform common deep learning networks having the same number of layers and weights, thus saving computational resources. Compared to similar architectures in literature, we proved that C–FRPN are more flexible and we deeply studied their properties. C–FRPN is applied on several benchmarks including the skin lesion classification task.

Second, we propose an approach to estimate age from brain MRI when the dataset is small and few computational resources are available. The method consists of extracting features from the slices composing a 3D MRI and then processing those

features by a recurrent neural network. The experimental results showed that such an approach can produce good performance, comparable to that achieved by a standard 3D convolutional neural network, while requiring less computational resources and making the training feasible even on small GPUs. The method is interesting in view of the fact that 3D MR images are usually large and require huge computational resources and datasets.

Major Contributions of the Thesis

The main contributions of the thesis are summarized below.

1. The proposal of a new CBIR system, based on siamese neural networks, for the retrieval of prostate multi–parametric MRI. *Based on* (*Rossi et al., 2020c*)

2. The design of a similarity learning schema realized by siamese neural networks to improve the robustness of a convolution neural network classifier with application to the classification of prostate lesion. *Based on* (*Rossi et al., 2020a*)

3. The proposal of an intra–procedural prostate MRI registration algorithm based on siamese neural networks. *Based on* (*Lyons and Rossi, 2020*)

4. The proposal of a new deep learning architecture, that mixes convolutional neural networks and recursive neural networks, aimed at producing high performance with few computational resources. A wide study of this architecture is presented together with its application to the classification of skin lesions. *Based on* (*See appendix A peer reviewed conference paper 3 and paper under review 1.*)

5. The proposal of a less demanding (in terms of resources) architecture, alternative to 3D convolutional neural networks, for the case of age prediction using brain MRI. *Based on* (*Rossi et al., 2019*)

Structure of the thesis

The thesis is organized as follows. Chapter 2 reports the state of the art in automatic medical image analysis along with a broad presentation of similarity learning and recurrent neural networks. Chapter 3 is devoted to the application of learning from similarity to the case of prostate MRI. In particular, Sec. 3.1 illustrates the new proposed CBIR for prostate MRI with the relative experimentation. Sec. 3.2 presents a method to increase the robustness of a prostate MRI lesion classifier using a siamese neural network. Sec. 3.3 proposes a new framework to register intra–procedural prostate MRI, which is based again on siamese neural networks.

Chapter 4 describes the application of recursive and recurrent models to medical images. In details, Sec. 4.1 presents the Convolutional Fully Recursive Perceptron Network. A large experimentation with the aim of discovering the properties of the model is included. Sec. 4.2 investigates alternative ways of processing brain MRI for the prediction of the patient age. The experimentation assesses that the proposed solution based on recurrent nets can be a valid alternative to common 3D convolutional neural networks. Finally, Chapter 5 briefly describes the other activities in which I was involved during my Phd, while Chapter 6 draws some conclusions on the presented research together with possible future perspectives.

Chapter 2

Machine learning in medical image analysis

Automatic medical image analysis appeared together with the availability of digital images, dating approximately back to 1970s. Early models were composed by low level pixel processing as edge detectors or region growing algorithms applied in a sequential fashion. The scheme behind these methods was extremely similar to common rule–based systems composing the first approaches to artificial intelligence (Haugeland, 1989). Two decades later, the use of machine learning start to become popular in medical imaging thanks to data storage facilities. The pioneering models were based on hand–crafted features extracted from the images, with the extraction process strictly dependent on the skills of the human expert. These features were provided to learning models that automatically adapt to a specific activity. The breakthrough towards more complex tools was the implementation of systems able to extract a relevant feature set without explicitly defining it, as in the Fukushima's neocognitron (Fukushima, 1980), the ancestor of the nowadays popular convolutional neural networks (CNNs). The first use of CNNs in medical image analysis was reported in (Lo et al., 1995), while the first relevant application to hand–written characters can be found in (LeCun et al., 1998). However, a great deal of attention from researchers was gained later, after the publication of (Krizhevsky et al., 2012) which achieves state–of–the–art performance in a challenge on classifying natural images into thousands classes (Deng et al., 2009). Since then, the same transition from hand–crafted features (Bengio et al., 2013) to CNNs has been gradually observed in the field of medical image analysis.

Deep CNN models

Given the importance of neural networks in the analysis of medical images, the main research contributions are reported below.

Image classification The older CNN models are very simple architectures (LeCun et al., 1998; Krizhevsky et al., 2012), since they comprise two and five convolutional layers, respectively, interleaved with pooling operations. Moreover, the kernel size was larger than in the modern models. A few years later (Simonyan and Zisserman, 2014) demonstrated that the use of smaller kernels allows to decrease the number of weights, making possible to increase the network depth. Anyway, this very deep network was not easy to train, and some problems, e.g., limited resources and vanishing gradients, induced research efforts in finding a workaround. The solution was found with the Inception model (Szegedy et al., 2015) that replaced a single convolutional layer with many convolutions working on the same input and having different kernel size, the output of whom are concatenated. The last very relevant step forward was achieved by (He et al., 2016), which proposed to learn the residual function by using skip connections. This produced the state–of–the–art in the IMAGE–NET challenge (Deng et al., 2009).

The massive use of transfer learning, either in the form of feature extraction or fine tuning, is particularly important in medical image processing. In the first case, a model trained on a task is used to extract discriminative features for a new problem, while, in the case of fine tuning, a pre–trained model is the base for a successive training phase involving all the network or just the final layers. The popularity of transfer learning specifically for medical image analysis is due to the general scarcity of data. In fact, training a very large model with few images could lead to poor performance. Using a pre–trained model, instead, allows to have a predefined set of kernels that can detect at least some general image properties, as corners and boundaries. While it is not clear which of the two processes is the best, many researchers successfully applied transfer learning to medical image tasks (Esteva et al., 2017; Gulshan et al., 2016; Antony et al., 2016; Kim et al., 2016).

In dealing with automatic image processing applications, especially involving large images, it could be useful to focus on small details while efficiently maintaining the context of the entire image. This is particularly true for the case of object classification, where a small part in a large context needs to be classified. Practitioners use to build multi–stream architectures feeding high resolution patches to one stream, while the other is fed with the low–resolution context (Farabet et al., 2012). Indeed, this principle is also useful for medical images (Kamnitsas et al., 2017; Moeskops et al., 2016; Song et al., 2015; Yang et al., 2017a), where the resolution can be really high, requiring a huge amount of computational resources, unless processing it in a multi–resolution way. Another challenge in medical images is the adaptation of CNNs to different inputs, as in the case of 3D images. In fact, by applying 3D convolutions, the model becomes really difficult to train. Researchers proposed workarounds based on the idea of providing slices or patches to different network streams (Prasoon et al., 2013; Roth et al., 2015; Setio et al., 2016).

Image segmentation and object detection A naive way to segment an image consists of applying a standard CNN using a sort of sliding window approach, i.e. by feeding the CNN with patches centered at the pixel to be classified. Unfortunately, this approach is very expensive from a computational point of view. A better solution consists of replacing the fully connected layer composing the head of many CNNs with a set of convolutions, with the last layer having a dimension equal to the input image, allowing to segment the entire image in a single step, as in (Long et al., 2015).

Of course, to obtain a label map of the same dimension of the input, an up–sample strategy is required, to compensate for the dimensionality reduction realized by pooling layers. A simple approach for up–sampling images involves employing nearest or bilinear interpolation, while, nowadays, a completely learnable up–sampling procedure is represented by the use of transposed convolutional or deconvolutional layers (Dumoulin and Visin, 2016).

One of the most famous models for medical image segmentation is the U–net, proposed by (Ronneberger et al., 2015), which uses skip connections to propagate the information from an encoder section to the corresponding layer in a decoder section of the network. This paper constitutes a reference for many other works, including the case of volumetric data processing, as in (Çiçek et al., 2016; Milletari et al., 2016).

Instead object or lesion detection/localization consists in the process of searching a sub–part of the image containing the object of interest. One of the first CNN model proposed in this ambit (Lo et al., 1995) was devoted to detecting nodules in lung X–ray images. A simple localization can be performed to directly predict the coordinates of a certain object in the image (Payer et al., 2016), or producing the coordinates of a bounding box containing the object (de Vos et al., 2016). Also in the detection case, the efficient introduction of contextual information is used in a similar way to the case of image classification, as in the multi–stream CNN reported by (Teramoto et al., 2016).

2.1 Learning by comparison

This section provides the principal findings in learning by comparison. Similarity has a central role in many cognitive processes. In fact, classification could be based on some relation with prototypes. Also in the case of problem solving, similarity is widely used by humans and can actually simplify the problem solutions. Finally, also learning new skills can be made easier by transferring some previous knowledge acquired in a similar framework. The first milestone in this field can be found in (Tversky, 1977), where a feature–based model is proposed, revealing how the perception of similarities increases with the matching of some features and

decreases in the case of dissimilarities. This presumes the existence of a universal feature set, while each object have its own subset of features, so that similarities can be defined by the intersection between the set describing two objects. However, successive works showed that similarities need to be expressed by hierarchical representations and not only by feature sets (Markman and Gentner, 1993). The benefits provided by the use of similarities are also exploited in the field of pedagogy (Rittle-Johnson and Star, 2011; Alfieri et al., 2013). For instance, analogies can help students to focus on the key point of some reasoning. Modern artificial neural network algorithms are mostly based on analogies with the human brain structure and functionalities, having their milestone in the work of (McCulloch and Pitts, 1943). Also the case of similarity learning has attracted attention allowing the development of new models approaching different aspects of the metric learning (Kulis et al., 2012). A straightforward way to use similarity is by comparing the embedding extracted from a pre–trained network or an autoencoder (Kramer, 1991). Anyway, this is an indirect comparison rather than a real learning from similarity. A simple and direct model that effectively learns similarities is the siamese neural network (Bromley et al., 1994). It is composed by two streams using the same set of weights. In the original implementation, the network received two inputs and the goal was to decide whether they belong to the same class or not. This was particularly effective in the few–shot learning environment as demonstrated in (Koch et al., 2015). Finally, by using a particular loss function (Hadsell et al., 2006) that try to penalize errors both for similar and dissimilar objects but in a different way, a sort of metric can be produced in output evaluating their "distance". The most interesting models proposing similarity learning based on siamese neural networks can be found in (Appalaraju and Chaoji, 2017; Wang et al., 2014; Zagoruyko and Komodakis, 2015; Zhang et al., 2016).

2.2 Recurrent and recursive networks

Another important application of artificial neural networks concerns the field of natural language processing and it originates from the intuition that the human brain has also recurrent connection and not just forward links (Dayan et al., 2003; Douglas and Martin, 2007). The first attempt to deploying recurrent connections in artificial neural networks, involving a time–lag feedback connection, may be attributed to (Elman, 1990; Williams and Zipser, 1989; Back and Tsoi, 1991), while (Pineda, 1987) introduced a generalization of the BackPropagation algorithm for training recurrent architectures.

In recurrent neural networks (RNNs), a set of shared weights is applied to each element composing a sequence. The main advantage is the huge saving in parameters together with a better generalization coming from the implicit presence of the

memory that allows a better contextualization. Unfortunately, in the original form, RNNs are prone to forget long–term dependencies. This drawback was overcome by a model called Long Short Term Memory (LSTM) network (Hochreiter and Schmidhuber, 1997), equipped with special gates allowing for the propagation of information only if necessary. Another step in this direction was done by the gated recurrent units presented in (Cho et al., 2014). For many years, RNNs have often achieved the state–of–the–art performance in many tasks involving sequences, such as speech recognition (Chorowski et al., 2015), machine translation (Bahdanau et al., 2014), text summarization (Rush et al., 2015) and many others. Even if feedback loops are present in both recurrent and recursive networks, in the former the input changes at every step, and the state naturally embeds information coming from the past input elements. Instead in the latter, the input is the same at every iteration and the state represents the knowledge "accumulated" by the network, which can be stopped after a predefined amount of iterations or when the state converges. A first attempt to use recursive connections involving a constant but unknown weighted feedback may be attributed to (Sperduti and Starita, 1997; Bianucci et al., 2000). While recurrent and recursive connections provide different behaviours in a multilayer perceptron (MLP) (Hagenbuchner et al., 2017), when a recursive MLP, e.g., (Hagenbuchner et al., 2017) is unfolded in time and truncated to a fixed number of stages, say K time steps, it resembles a recurrent neural network involving K time steps. In particular, (Hagenbuchner et al., 2017) proposed a simple MLP equipped with recursive links and reported better performance with respect to the corresponding shallow MLP with the same number of parameters. Another interesting approach, described in (Liang and Hu, 2015), generalizes the concept of recursive networks[1] to CNNs reporting high performance in a benchmark of natural image classification.

[1]Notice that the authors call their model recurrent convolutional neural network and not recursive, since they do not distinguish between recurrent and recursive networks.

Chapter 3

Siamese neural networks for prostate MRI

This chapter is entirely dedicated to the application of siamese neural networks and learning from similarity to the case of prostate MRI, given the relevant incidence of prostate cancer, and the promising screening performance reported by MRI.

In fact, prostate cancer is the most common cancer and the second most deadly cancer in men in the US (Siegel et al., 2019). Based on the GLOBOCAN 2018 estimates of cancer incidence and mortality, produced by the International Agency for Research on Cancer, with a focus on geographic variability across 20 world regions, prostate cancer is the second most commonly diagnosed cancer (7.1% of the total cases) and the fifth leading cause of cancer death in men (Bray et al., 2018). Relatively little is known about the prostate cancer etiology, apart from body fatness, for which there is a convincing evidence of an association, and the ethnic and genetic predisposition (f.i. for South–African and Caribbean population). Since the mid eighties, prostate cancer incidence has strongly risen by the diagnosis of latent cancers by PSA (Prostatic Specific Antigen) test for early detection. PSA and systematic biopsy, evaluated with the Gleason score, can reduce the mortality rate, but at the expense of a huge overtreatment (Schröder et al., 2009).

Multi–parametric Magnetic Resonance Imaging (mpMRI) can help avoiding unnecessary biopsies, reducing overtreatment (van der Leest et al., 2019; Rouvière et al., 2019) and improves early detection. However, interpreting prostate MRI is difficult, significantly dependent on the experience of the involved clinician and, therefore, diagnostic accuracy varies (Khanna and Crues, 2009; Rosenkrantz et al., 2016). One approach to reduce the evaluation variability is the definition of a reader guideline (PIRADS (Weinreb et al., 2016)). Nevertheless, also using PIRADS, the interpretation of mpMRI remains subject to some uncertainty (Smith et al., 2019; Barentsz et al., 2016; Muller et al., 2015).

Computer–Aided Diagnosis (CAD) tools can be helpful in reliable interpretation

of mpMRI. A common prostate CAD workflow presupposes to locate a suspicious lesion and compute the likelihood that it represents a significant prostate cancer (Litjens et al., 2014). In recent years, conventional feature–based CAD systems have been supplanted by deep convolutional neural networks (CNNs). Indeed, CNNs have been applied also to prostate mpMRI to detect the presence of clinically significant cancer (Song et al., 2018b; Yang et al., 2017b; Wang et al., 2018; Le et al., 2017; Schelb et al., 2019). Yet the performance is still sub–par with trained radiologists.

A siamese neural network is a connectionist architecture that allows to compare two input patterns, eventually assessing if they belong to the same category. Even if siamese networks can be used for any type of inputs (see (Bromley et al., 1994)), usually they are applied to image processing tasks, and they were shown to be particularly useful for image retrieval, verification, and few–shot learning (Chopra et al., 2005; Zagoruyko and Komodakis, 2015; Chung and Weng, 2017; Yi et al., 2014). The architecture of a siamese is constituted by a single convolutional neural network (CNN), which is used on both the input images in order to extract their features, and by a distance function, which measures their similarity (see Fig. 3.1).

Actually, in order to evaluate the similarity between two images, X_1 and X_2, a metric has to be defined in the embedding feature space, namely a parametric function G_W, realized by the CNN contained in the siamese. A common choice is the Euclidean distance, denoted as $D_W()$ in the following.

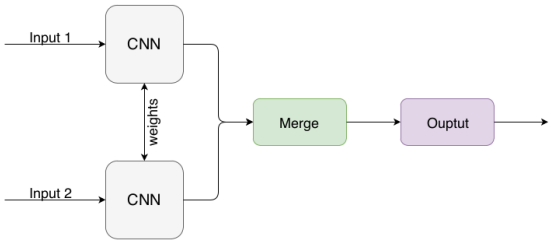


Figure 3.1: The siamese model.

However, the adoption of a contrastive loss function (Hadsell et al., 2006) was shown to be more effective to learn similarities. The contrastive loss presumes the availability of a supervised similarity label Y for each pair of images X_1, X_2, defined as

$$Y = \begin{cases} 0, & \text{if } X_1 \text{ similar to } X_2; \\ 1, & \text{otherwise.} \end{cases} \tag{3.1}$$

Formally, by re–writing $D_W(X_1, X_2)$ as D_W for brevity, the contrastive loss function L is:

$$L(W, Y, X_1, X_2) = (1 - Y)(D_W)^2 + Y[max(0, m - D_W)]^2 , \tag{3.2}$$

where $m > 0$ is a margin defined so that a pair contributes to the loss only if its distance D_W belongs to $(0, m)$. Intuitively, this loss makes the embeddings $G_W(X_1)$,

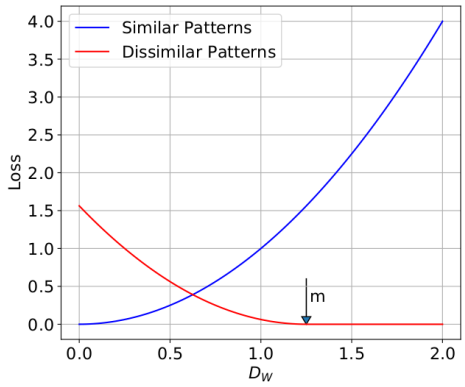


Figure 3.2: The contrastive loss function behavior. The blue line represents the function applied to similar patterns, while the red line is related to dissimilar samples.

$G_W(X_2)$ closer for similar inputs, and more distant for inputs that are different. A plot illustrating the loss is shown in Fig.3.2. Pairs of patterns that are close in terms of their embedding distance D_W produce a very low loss (blue line) if they are similar, whereas the loss is large if they are dissimilar (red line). A siamese network can be trained end–to–end using a common optimization method. More precisely, the learning procedure is similar to that used in standard CNNs, with few peculiarities. The training set consists of pairs of images (query–reference), to be constructed according to some predefined criterion. The contrastive loss allows to compute the gradient with respect to the siamese network parameters, namely the parameters of the embedded CNN, for each pair in the dataset. Finally, any common gradient–based optimization method can be applied.

In the following Sec. 3.1, a Content–Based Image Retrieval (CBIR) system is presented, based on siamese networks and diagnostic similarity. Then, in Sec. 3.2, an investigation on how to increase a classifier robustness to random noise and adversarial attacks is proposed, using again siamese networks and similarity learning. Finally, in Sec. 3.2, the same learning scheme is used to attack the image registration task.

3.1 A novel CBIR for prostate MRI

A CBIR can be employed for CAD, when used to retrieve similar cases to a given query image during the MRI reading and reporting. CBIR systems require a very rich data representation to identify similar cases in large databases, such as those that can be found in a medical center. A common approach for general applications and, in particular, for medical images is to extract some kind of features (e.g. SIFT, HOG) to obtain a compact representation, useful to measure the distance between the query image and the images in the entire dataset (Kumar et al., 2013; Müller

et al., 2004). The success of deep learning approaches in many image processing tasks is actually due to the possibility they offer to extract features automatically — for instance, based on the output of intermediate layers in a neural network —, simplifying the design of the system. Indeed, CBIRs based on CNNs often outperform classical CBIRs (Sun et al., 2017; Anavi et al., 2016). The most common neural network model used as CBIR is the autoencoder, which allows to encode images into a robust and lossless representation, that is usually preferable with respect to hand–crafted features of classical CBIRs (Liu et al., 2016; Zhang et al., 2015; Song et al., 2018a; Xu and Fang, 2016; Guo et al., 2015; Geng and Song, 2016; Knyaz et al., 2017). Despite their effectiveness in dealing with massive datasets, few works exist in the context of CBIRs for prostate cancer imaging.

A first attempt to develop a CBIR for medical image processing was proposed in (Wetzel et al., 1999), where microscopy images of prostate tissues from biopsies were analysed. The model retrieves the most similar images according to the Gleason score (Gleason, 1992), based on handcrafted features. In the field of magnetic resonance, in (Mitra et al., 2012), the retrieval is based on correspondences between 2D transrectal ultrasound images and MR slices, exploring the joint probability of shape and image similarities. A CBIR–like approach is also used to guide prostate segmentation in (Chandra et al., 2012), where a locally normalized mutual information metric is used to build a patient–specific atlas. More recently, in (Shah et al., 2016), the idea of employing a CNN for feature extraction from transversal T2W images — to be processed in a subsequent step by a CBIR system — is introduced. Then, the Euclidean distance is used for the association among the query and the stored images using the extracted features. In this way, only the visual similarity among images is considered, completely disregarding diagnostic information.

In this section, we propose a new CBIR for multiparametric prostate MRI in which similarity is defined considering the severity of the lesion expressed by the PIRADS score. Indeed, a retrieval system able to consider not only the appearance of the images but also their diagnostic meaning can help radiologists to understand a new case, assigning the right relevance score. The basic block of our model is the siamese neural network trained with the contrastive loss defined in Eq. (3.2). Another great novelty of this work is the integration in the CBIR of mpMRI (i.e. T2W and High–b value images estimated by the DWI sequences). Moreover, also the multi–view case is considered, using both axial and sagittal T2W images. A comparison with a CBIR based on a shallow autoencoder reveals how our model provides very good performance both in terms of diagnostic (ROC–AUC) and information retrieval (Precision–Recall, Discounted Cumulative Gain and Mean Average Precision) metrics. In particular, the multi–modal version improves over the single–mode approach, while the multi–modal multi–view siamese reports performance similar to the previous case, without further improvements.

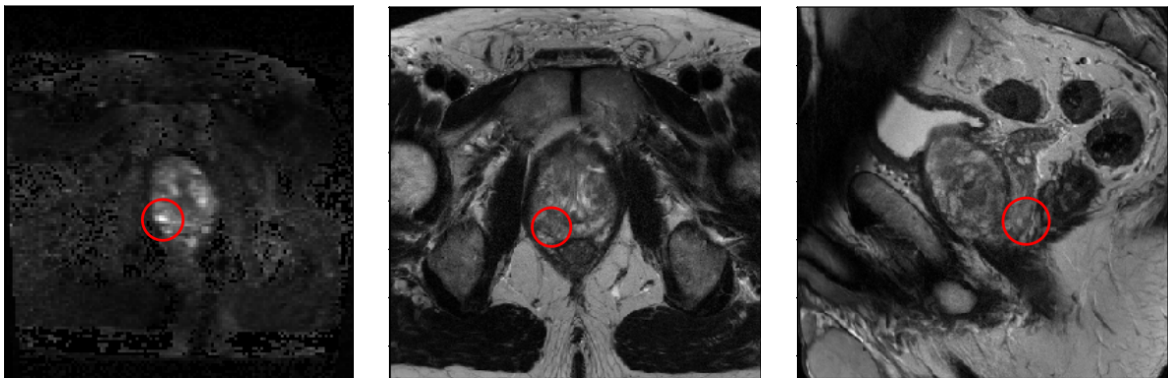


Figure 3.3: Different MR image views for the same lesion. Axial–HBV (left), axial–T2W (center), sagittal–T2W(right).

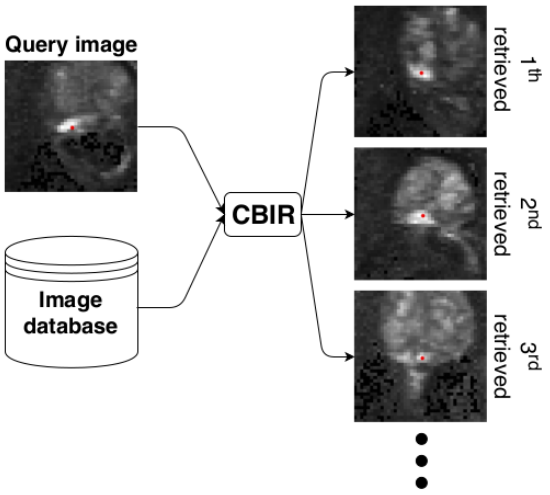


Figure 3.4: A CBIR example.

Background topics

A CBIR is an information retrieval system that, in response to a query image, searches for the most similar images in an archive. The images returned to the users are usually sorted according to the relevance with respect to the query (see Fig. 3.4). The component of the CBIR in charge for sorting the images, namely the ranker, exploits an image similarity measure that allows to compare the query with the images of the archive. In our approach, the similarity measure is implemented by a siamese neural network.

Autoencoders Autoencoders are artificial neural networks that allow to represent images with compressed feature vectors. An autoencoder network comprises two components: an encoder and a decoder. The encoder is fed with the input and produces a latent representation of it (namely, a feature vector); starting from the latent

representation, the decoder tries to reconstruct the original input. In image processing applications, the encoder is usually a convolutional neural network, while the decoder is a deconvolutional network. Autoencoders are usually trained based on the Mean Squared Error loss.

Thanks to their ability of compressing image information, autoencoders allow to implement CBIRs in which features are automatically extracted[1]. CBIRs based on autoencoders often outperform systems based on hand–crafted features, which have also the disadvantage of requiring a more complex design (see e.g., (Krizhevsky and Hinton, 2011; Sharma et al., 2016; Geng and Song, 2016; Knyaz et al., 2017; Zhang et al., 2015; Song et al., 2018a; Xu and Fang, 2016; Guo et al., 2015)).

Method

Data The available dataset is a private collection consisting of 601 consecutive multi–parametric prostate MRI of low risk lesions, acquired routinely at the Radboud University Medical Center in 2016 for the detection of clinically significant prostate cancer. The digital reporting of PIRADS and the position of each lesion was evaluated by one or more dedicated prostate radiologists. In total 890 candidate lesion were collected. For each patient, trans–axial MR images were available in T2W and Diffusion Weighted Imaging (DWI), accounting for both ADC and HBV modalities, while sagittal images were available only in T2W.

The resolution of trans–axial T2W images is $0.5 \times 0.5 \times 3.6$ mm, while trans–axial HBV images have a voxel spacing of $2.0 \times 2.0 \times 3.6$ mm. The spacing of the sagittal T2W images is $0.56 \times 0.56 \times 3.6$ mm.

PIRADS reporting software allowed to indicate and store the coordinates of each lesion (approximately the center). A $40 \times 40 \times 3$ voxel ROI, centered about at the lesion position was selected. Such a size value guarantees to cover all the surface of

[1]Notice that autoencoders can be more suitable for CBIRs than other deep learning models, such as ResNet or Inception. Actually, the autoencoder training aims at building features that contain a compressed representation of the input image, from which the same image can be reconstructed. Conversely, in common CNNs, the training focuses in deriving features that are useful to solve a given task, f.i. image classification. Thus, features produced by autoencoders are approximately lossless, whereas features produced by other models may lose most of the original information (f.i., in the very last layer of a ResNet, only the information needed to predict a class is available). Since a CBIR system has to compare a query with the database images, it is important that the encoding maintains all the relevant information on the input image. The disadvantage of CNN features may not be evident for CNNs trained on very huge and generic datasets of images, but they can become more important on small specialized datasets, such as the one encountered in our application. For example, in (Xu et al., 2015), the quality of the features extracted by an autoencoder are shown to outperform the features extracted by CNNs in the case of nuclei detection in breast cancer histopathology. To this aim, the autoencoder is trained on a dataset of images. Then, it is used to map each image into its latent feature space representation, which is stored in an archive where the search operations are carried out. More precisely, when a query is received, the query is transformed into its feature vector and compared with the feature vectors collected in the archive, in order to find the most similar images.

the lesions in the dataset. Fig. 3.3 shows an example of a prostate MRI with T2W (axial and sagittal) and HBV (axial) images containing a lesion.

CBIR model: Autoencoder This model is the simplest CBIR proposed in this work. It is composed of a standard autoencoder fed with trans–axial T2W or HBV images. Once the autoencoder is trained, the encoder part of the network is used to compute a feature vector for each image of the dataset. During the test, we compute the Euclidean distance between the encoded representation of the test image, namely the query, and all the encoded representations of the images in the training set. The system retrieves those images of the training set having the smallest distances from the query. To compare a query with the entire archived representations of the training set images takes only a few milliseconds on a normal laptop. Notice that, due to the very nature of autoencoders, there is no way to consider also diagnostic information in retrieving images. This CBIR can only be based on the visual similarity among images.

CBIR model: Single–view siamese (SiamHBV or SiamT2W) This CBIR model (and the models reported in the following) is based on siamese neural networks. The model is composed by the red parts in Fig. 3.5. During training, the two input images are considered similar if they have a similar diagnostic evaluation, i.e. they have close PIRADS scores. After the Siamese has been trained, it is possible to calculate the feature vectors of both the query and the reference image. The two embedding vectors are then compared by computing their Euclidean distance, which provides the similarity score. Repeating this procedure for all the images in the dataset allows to find the most similar images matching the query. As in the case of the autoencoder, this procedure takes a few milliseconds on a laptop with an Intel i3 CPU and 8 GB of RAM[2].
In the single–view siamese, the CBIR makes use of a single axial modality. Thus, the siamese includes a CNN that is fed with a unique type of images, namely T2W or HBV.

[2]In order to have a fast implementation of a siamese method, after the training, the database images are pre–processed to extract their features. Then, such features are stored in an archive and possibly indexed by an appropriate multi–indexing data structure, which allows a fast access (see e.g. (Böhm et al., 2001)). When a query is received by the CBIR, the features of the query are extracted and compared with those stored in the archive. The comparison is fast, since feature vectors are a compressed representation of the original images. Moreover, CBIRs return only few top images, i.e. those closer to the query, obtained by accessing the index and without comparing the query against every image in the database. For a multi–index based on a tree data structure, such an operation is very efficient: formally, the cost of retrieving k images in an index of N images is $O(k + \log_b N)$ on average, where b is a constant depending on the implemented data structure (Böhm et al., 2001) . In our experiments, we did not use any indexing mechanism, since it was not required.

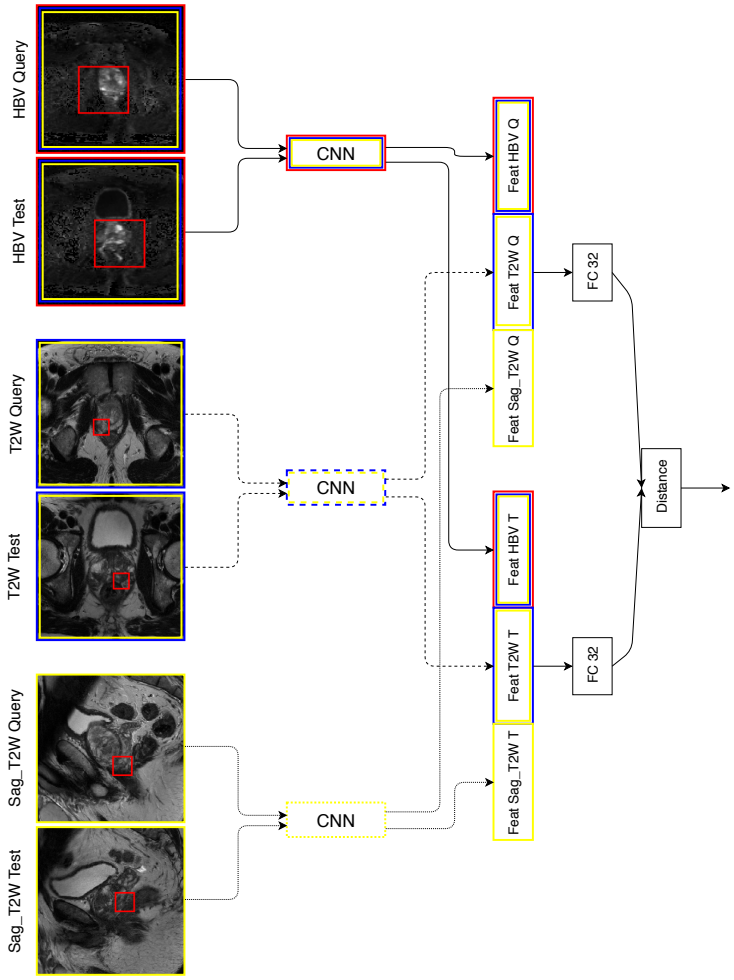


Figure 3.5: Multimodal siamese network.

CBIR model: Multi–modal parallel siamese (Double) This CBIR model makes use of two axial modalities, represented by the blue boxes in Fig. 3.5. In fact, in order to clearly understand the nature of a lesion, radiologists have to look at both T2W and DWI images, and possibly also at different views, as axial and sagittal views. Therefore, the siamese takes in input four images: two images (i.e., axial T2W and HBV) for the query and two images for the reference. These pairs of images come from two different lesions. The output of the siamese is the Euclidean distance between the embeddings of the query and the reference. In order to deal with the two image modalities, the network is composed of two different CNNs, having the same architecture but a different set of weights (see Fig. 3.5). Each of the two CNNs pro-

duces an embedding vector for each modality. After that, four embedding vectors are present, namely the embeddings of T2W and HBV for both the query and the reference image. We found that the concatenation of the two vectors results in an effective multi–modal embedding. In this way, the embeddings are combined by the fully connected (FC) layer, which can automatically produce different types of fusion, including f.i. summing or averaging. Moreover, there is no need to resample images coming from different modalities to the same resolution, since they are processed independently by two different branches. Finally, both the training and the inference processes are the same as in the previous case, based only on one image modality.

CBIR model: Multi–View, orthogonal siamese (Triple) This CBIR model is a further extension of the multi–modal parallel siamese and exploits also T2W sagittal images. The siamese is equipped with another stream, responsible for processing the added image modality (see the entire Fig. 3.5, fully enveloped by a yellow border). In this case, the system receives as input six images, accounting for two sets (query and test), each containing three image modalities (axial T2W and HBV, and sagittal T2W) that describe a lesion in two patients. Consequently, also the number of CNNs is increased to three. The corresponding embeddings are concatenated and used to compute the multi–view similarity.

Experimental setup

Siamese and CNN architectures In all the experiments, the proposed siamese models include a CNN, whose initial part has four pairs of convolutional–pooling layers. In the convolutional layers, the filter size is 3×3, the stride is 1, the padding is set to "same", the activation function is ReLu, and the number of feature maps is 8, 16, 32 and 64, respectively. In pooling layers, the pooling size is 2×2. After the last pooling, the CNN has another convolutional layer with 96 feature maps (the other hyper–parameters are as described above), a batch normalization layer and, finally, a dropout layer.

The CNN, used as the initial part of the siamese, produces a single embedded representation for each image modality of both the query and the reference image. These representations are then concatenated to obtain two vectors (see Fig. 3.5). A last FC layer having 32 units, with sigmoid activation functions, is responsible for the final embedding.

The training was carried out for 30 epochs using the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 10^{-5}, based on the contrastive loss function (Eq. (3.2)). During training, two lesions were considered similar if both their PIRADS scores belong to one of the three PIRADS sets: $\{1, 2\}$, $\{3\}$, or $\{4, 5\}$, representing un-

likely, equivocal, or likely significant lesions (Weinreb et al., 2016). This is in contrast with the standard use of PIRADS in which the scores are grouped in $\{1, 2, 3\}$, representing non–relevant cancer lesions, while $\{4, 5\}$ refer to relevant cases. Anyway, training the network following the above described policy is proved to be more efficient (based on the validation set), at least for some metrics as the DCG score, that considers ungrouped annotations. According to the reference document (Weinreb et al., 2016), differences among lesions are both in texture and size.

Autoencoder architecture The encoder stacks three pairs of convolutional–pooling layers. The number of feature maps in the convolutional layer is 90, 45, and 5, respectively. The decoder stacks three pairs of convolutional–upsampling layers. The number of feature maps in the convolutional layers is 5, 45, and 90, respectively. A final convolutional layer with 3 feature maps produces the output image. Kernel sizes, padding and activation functions of the convolutional layers are the same as in the siamese model. In order to have a more fair comparison between the proposed models, we designed the autoencoder to have a number of parameters closed to that of the siamese. All the models were implemented using the Keras library[3] and, performing 3 runs with different initializations.

The dataset For the experiments, the dataset has been randomly split into 672 lesions for the training set, 90 for the validation set, and 128 for the test set. The validation set was used to optimize the network hyper–parameters (size of the embedding and activation function) and for early–stopping. Each query image of each set is compared with all the images in the training set having a lesion in the same prostate zone. Notice that, in our experiments, we suppose that the CBIR archive is constituted by the training set. Since the siamese is fed with pairs of images, 213,990 pairs (query–reference) are used for training, 29,555 for validation, and 40,033 for testing. No image augmentation techniques have been used. The image pre–processing consisted in scaling the pixels in the range $[0, 1]$, dividing each pixel by the maximum value in the dataset (considering only the ROI, not the entire scan). Moreover, a mask is used in order to highlight the central part of the image, containing the lesion. This mask is obtained by applying a Gaussian function centered in the ROI center and having standard deviation of 5. A preliminary experimentation has shown that this pre–processing phase provides a benefit for the siamese but not for the autoencoder, so that it has not been used in the last case.

Since the model analyzes only a crop surrounding the lesion, common bias removal methods (Tustison et al., 2010) are unnecessary, as they tend to remove the texture characterizing a lesion, which is an important information for the considered task.

[3] https://keras.io/

Evaluation Metrics To measure the performance of the proposed methods, we use both standard information retrieval metrics and criteria able to evaluate their diagnostic performance. In all the cases, we consider only up to the first $R = 10$ images returned by the CBIR. Moreover, the retrieval is correct if both the query and the retrieved images belong to the same PIRADS set, namely $\{1,2,3\}$ or $\{4,5\}$. The two sets represent the presence of clinically significant cancer as unlikely or probable.

Precision/Recall @ K Precision and Recall measures are defined as:

$$precision = \frac{|\{relevant\ images\} \cap \{retrieved\ images\}|}{|\{retrieved\ images\}|}, \tag{3.3}$$

$$recall = \frac{|\{relevant\ images\} \cap \{retrieved\ images\}|}{|\{relevant\ images\}|}. \tag{3.4}$$

The precision accounts for the capability of a CBIR to reject not–relevant images from the retrieved set, while the recall stands for its capacity of finding the highest number of relevant images. Moreover, precision@K and recall@K differ from the previous definitions solely because only the top K–ranked images returned by the CBIR are considered.

Discounted Cumulative Gain (DCG) This metric allows to measure the ranking quality of the CBIR, namely its capability to correctly sort images with respect to their similarity to the query. Its use is common in information retrieval to evaluate search engine performance. DCG is particularly useful when there are more than two relevance values and the usefulness of an image in the retrieved ranked list decreases more than linearly. Let REL_i be the relevance associated to an image at rank i. In this case, $REL \in \{0,1,2,3\}$ is defined using the difference d between the PIRADS of the query and the retrieved image: in particular, $REL = 0$ if $d > 2$, $REL = 1$ if $d = 2$ with one element of PIRADS equal to 3, $REL = 2$ if $d = 1$, or $REL = 3$ in the case $d = 0$. Therefore, the DCG can be defined as:

$$DCG@K = \sum_{i=0}^{K} \frac{REL_i}{log_2(max(i,1))} \tag{3.5}$$

where $K = 10$ is the maximum number of elements retrieved for each query. The resulting plot is generally unbounded, with better results showing a higher curve.

Mean Average Precision MAP is another scoring method suited for information retrieval tasks. In particular, it takes into account also the order of the returned documents. For a given query $q \in Q$, where Q is the entire set of queries, the average precision is:

$$AveP(q) = \frac{\sum_{k=1}^{n} P(k) * rel(k)}{R} \tag{3.6}$$

where $P(k)$ is the precision at rank k, $rel(k)$ is an indicator function related to the relevance of the item at rank k, R is the number of relevant documents, and n is the highest rank considered. Given this definition, the MAP can be formulated as:

$$MAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q} \tag{3.7}$$

Note that we stop our precision evaluation at the $k < 10$ position, that is, $n = 10$, as only the ten matching best images need to be retrieved.

ROC–AUC This metric evaluates the diagnostic performance of a simple classifier based on the proposed CBIR. The CBIR classifier works as follows. The CBIR returns a ranked list of images with the corresponding PIRADS scores: a low rank means a large image embedding distance and a low similarity in diagnostic score with respect to the query image. By this ranked list, we can define a score accounting for the probability of a query image to represent a relevant cancer pathology as:

$$p(c) = \frac{1}{\sum_{i=0}^{N} i} \sum_{i=0}^{N} (N-i)d \quad d = \begin{cases} 1, & \text{csPCa,} \\ 0, & \text{non csPCa ,} \end{cases} \tag{3.8}$$

where N is the number of retrieved images, d is the diagnoses of the retrieved images, csPCa is a clinically significant prostate cancer, namely a lesion having PIRADS equal to 4 or 5, and non csPCa is a non clinically significant prostate cancer, having PIRADS equal to 1, 2 or 3. By returning the above score, the CBIR becomes a diagnostic tool[4], whose performance can be measured by the Area Under the Receiver Operating Characteristics (ROC–AUC). We assess the statistical significance of our results by using the two–sided T–test.

[4]Note that the way in which the score is generated is very simple and likely subject to improvements, but a complete investigation of this aspect is out of the scope for this research.

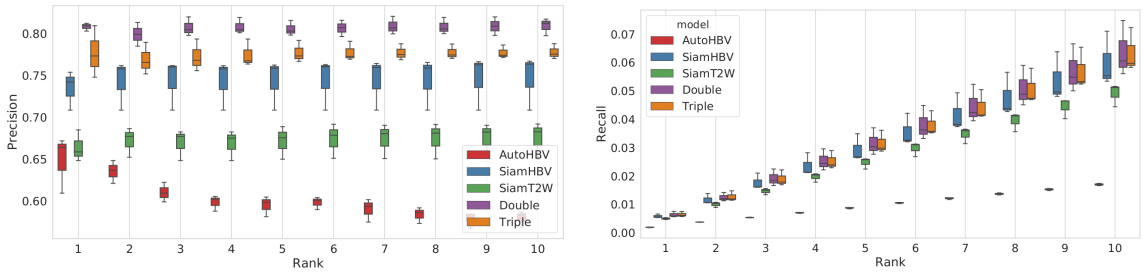


Figure 3.6: Precision@K (left), Recall@K (right).

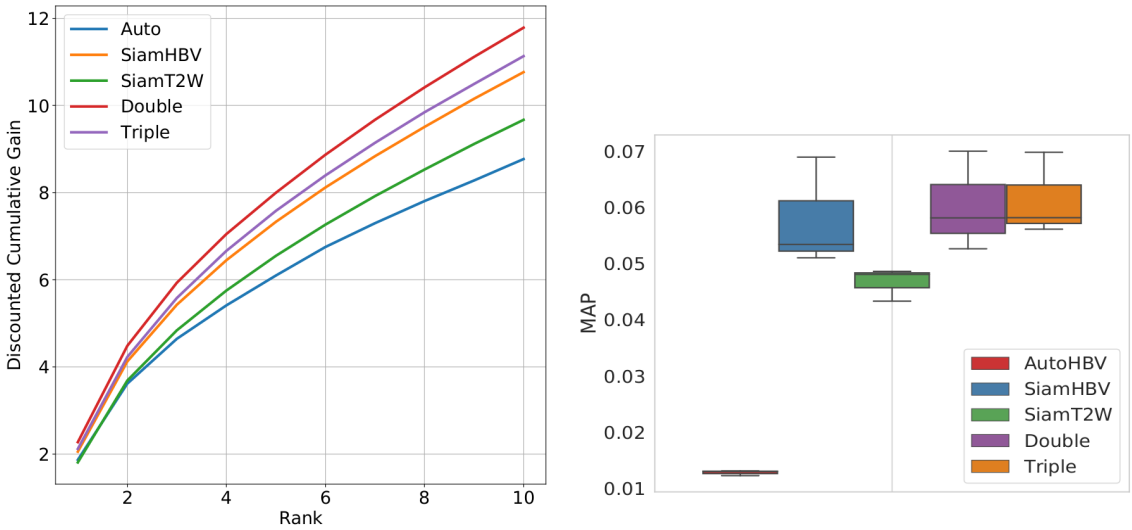


Figure 3.7: Discounted Cumulative Gain (left), Map Score (right).

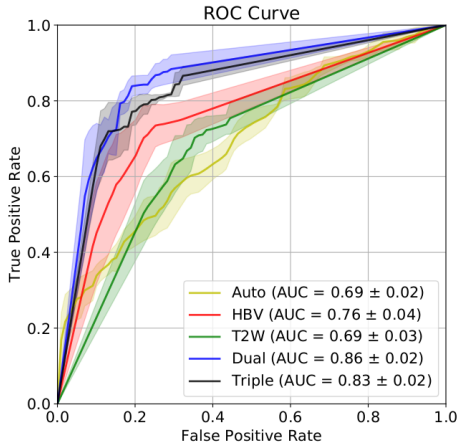


Figure 3.8: ROC curves for each experiment.

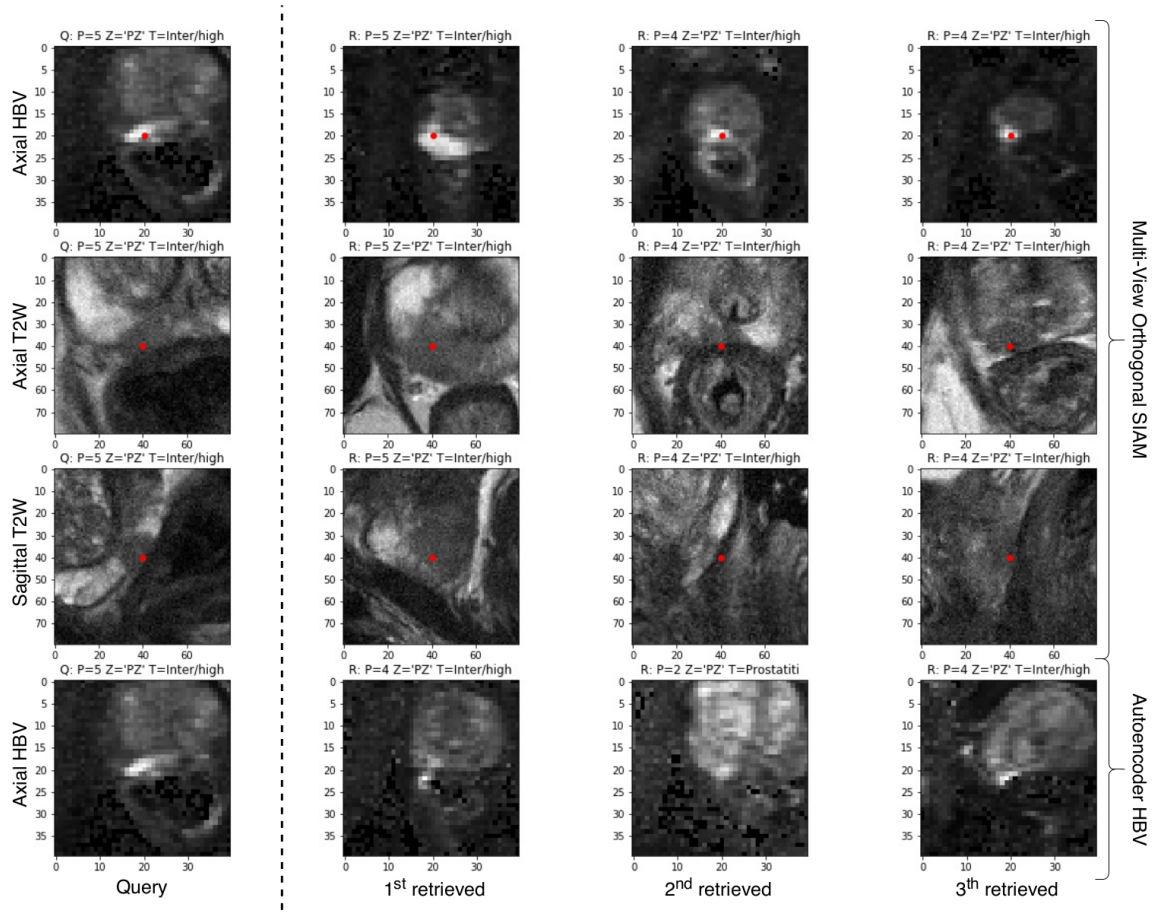


Figure 3.9: Qualitative results from the Multi–View, orthogonal siamese (Triple) experiment (rows 1,2,3) and autoencoder (row 4). The first column shows query images, while the other three columns contain the top three ranked results. (Q=query, R=retrieved, P=PIRADS, Z=Prostate Zone, T=Type of lesion)

Results

The results, reported on the test set, have been averaged over three different runs. Precision and recall at different ranks are reported in Fig. 3.6, while Fig. 3.7 shows the discounted cumulative gain and the MAP score. The boxplot represents the three quartile values of the obtained results together with extreme values. The whiskers extend to points that are within 1.5 in the interquartile range of the lower and upper quartiles. Samples that fall outside this range are displayed independently. Moreover, the ROC–AUC is shown in Fig. 3.8. Finally, Fig. 3.9 shows some examples of the images retrieved by the Multi–View, orthogonal siamese CBIR.

The proposed charts prove that all the siamese models outperform the autoencoder CBIR, suggesting that our siamese approach efficiently exploits the similarity based on clinical relevance. In fact, the siamese CBIR models achieve better results than the autoencoder CBIR with respect to all the considered measures. Similarly,

the single–view siamese works better with HBV images than with T2W. Moreover, adding more views results in a general increase in performance, although the results for the multi–view orthogonal model (3 views) are close to those of the multi–view parallel siamese (2 views). The qualitative results of the multi–modal multi–view siamese (Fig. 3.9) confirm the generally good performance of the model. Actually, for a given query, our CBIR returns lesions having the same PIRADS score not constrained to the same prostate region. Instead, the autoencoder output is reported in the last row of Fig. 3.9, showing how this model returns images with the same visual appearance, consisting in scans of lesions in the same region of the prostate, even if not necessarily sharing the PIRADS score.

Discussion and conclusions

We have shown that Content Based Image Retrieval (CBIR) systems based on neural networks can retrieve diagnostically similar images, being a powerful tool to assist radiologists. However, not all CBIR methods are suitable for prostate MRI analysis. Indeed, in this case, the similarity has to be based both on the closeness of the PIRADS scores and of the visual appearance of the lesions. This study shows that the autoencoder based CBIR, which is the common architecture for neural network CBIRs, focuses only on visual appearance. Conversely, siamese based CBIRs improve over autoencoders and can use both criteria. Fig. 3.9 shows how the siamese network can retrieve more appropriate images than the autoencoder. The difference between the two methods lies in the way in which learning is carried on. In fact, the siamese training explicitly exploits the diagnostic similarity, whereas the autoencoder is unable to easily use such an information (anyway implicitly codified in the images). Moreover, the difference in performance of the two methods is confirmed also when the CBIR output is used to implement a classifier to predict the diagnosis for the pathology represented in the image. Fig. 3.8 shows that the ROC–AUC of the diagnostic prediction increases from 0.52, for the autoencoder, to 0.69 for the T2W single–view siamese (p–value=0.028), and 0.76 for the HBV single–view siamese (p–value=0.018), respectively. We have also demonstrated how the siamese architecture can be extended to simultaneously process and integrate multiple views, showing a further increase in performance with respect to the single–view siamese. A significant difference was observed between a single–view siamese using only the HBV image and the multi–view parallel siamese, with the ROC–AUC that improves from 0.76 to 0.86 (p–value=0.03). Also, the multi–view orthogonal siamese performs better than the single–view siamese, but it does not provide the expected improvement with respect to the multi–view parallel network, reporting a ROC–AUC of 0.83. A possible motivation can be that, for the multi–view parallel siamese, both input images are acquired in the axial plane with different methods, while, in the multi–view orthogonal architecture, the third image has a totally different or-

thogonal prostate view appearance. Finally, a three–view network requires three different CNNs, yielding a higher number of parameters. This can be problematic when the number of training patterns is limited, as in the present case.

The integration of a CBIR into a CAD system is also possible. The classification of the retrieved images can be used as an additional, independent feature to support a classifier. The images can also be used to provide an explainable AI output.

To confirm the clinical value of this method, observer studies are needed, designed to investigate which benefits radiologists can achieve using the CBIR in clinical practice. This is a matter of future research.

Finally, the concept of retrieving similar objects could also be applied to assist medical image segmentation. CBIRs trained to retrieve other images with similar segmentation could serve as a sort of prior knowledge in a segmentation deep learning framework.

In conclusion, we have presented a novel siamese CBIR architecture that allows to integrate both clinical and visual information in a multi–parametric medical imaging task, to predict diagnostically similar images. This has been successfully demonstrated in the case of prostate MRI, though the described framework is general enough to be easily applicable to different image types.

3.2 Robust Prostate Cancer Classification with Siamese Neural Networks

In this section, we explain how to build a robust classifier using siamese neural networks (Bromley et al., 1994). In particular, we propose a hybrid siamese network equipped with a combined loss, including both a cross–entropy and contrastive term, to implement a robust prostate lesion classifier. The idea is similar to the one proposed by (Baddar et al., 2017), where a siamese network, trained with a cross–entropy loss plus a customized Euclidean distance, is used to improve the robustness of a facial expression recognition system.

To show the potential of our approach, a set of experiments was conducted, comparing the siamese network performance with a standard ResNet classifier. For the sake of fairness, the hybrid siamese network is made up of a ResNet backbone, so that only the loss function is changed. The two architectures have been evaluated on a validation set injected with random noise, calculating the corresponding decrease in performance in terms of AUC. Finally, to get a more comprehensive analysis, we also applied adversarial attacks to the models. The results clearly reveal a substantial increase in robustness for the hybrid network for both tests, along with better performance.

Materials & Methods

In this section, the hybrid siamese network is described in detail along with the data employed for the experiments. The data preprocessing procedure and the experimental setup are also explained.

The Dataset The images used for this study come from the ProstateX challenge (Litjens et al., 2014), hosted in the 2017 SPIE Medical Imaging Symposium. The dataset is composed by NMR images acquired in different modalities, namely T2–weighted (T2W), proton density–weighted (PD–W), dynamic contrast–enhanced (DCE), and diffusion–weighted (DWI), providing both High b–Value images and ADC maps. The training collection is composed by 330 lesions coming from 203 patients. For each lesion, the clinical significance (yes/no), the position of the lesion in physical coordinates and the voxel identifier are reported, together with the affected prostate zone, namely apical (AS), peripheral (PZ) and transition (TZ). The corresponding zonal distribution is reported in Table 3.1. Instead, the test set collects 208 images, related to 141 patients, with attached coordinates and prostate zone for each lesion, but without the target. Any classification method can be evaluated only by submitting the obtained results to the challenge web page[5].

[5]https://prostatex.grand-challenge.org/

Prostate Zone	Number	% not Significant	% Significant
Apical (*AS*)	55	44	56
Peripheral (*PZ*)	191	81	19
Transition (*TZ*)	82	89	11

Table 3.1: The dataset distribution with respect to the lesion prostate zone and clinical significance.

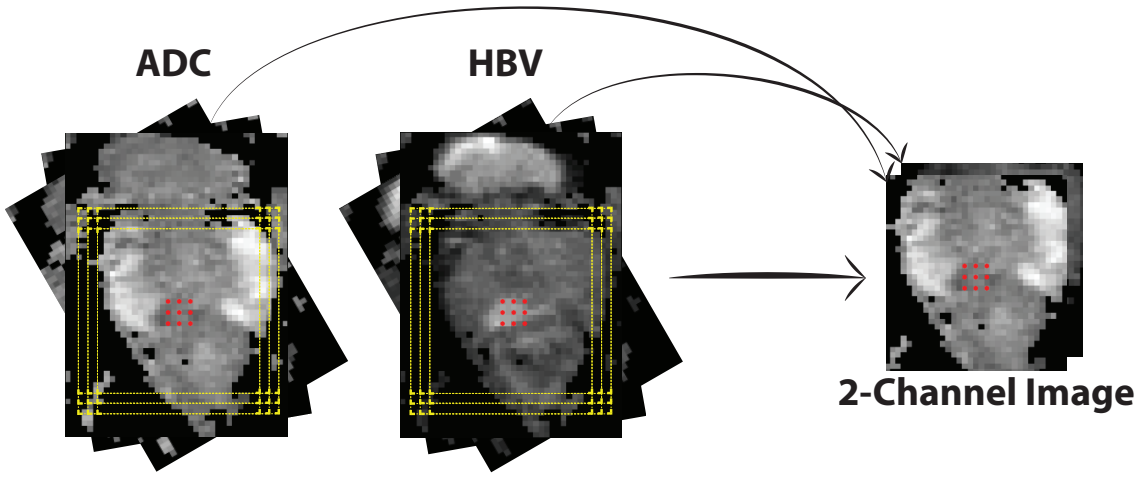


Figure 3.10: Image augmentation and composition.

Image Preprocessing The standard radiological procedure employs three different modalities to correctly evaluate a prostate lesion, namely T2W, ADC and High b–Value (Weinreb et al., 2016). Of all the models evaluated on ProstateX, one of the few with an associated paper is (Liu et al., 2017). Here, in fact, the maximum performance on the validation set was achieved by using a combination of T2W, ADC and High b–Value images. Instead, the best test AUC (equal to 0.84) was obtained from an ensemble model, where also other modalities, with a low score on the validation set, were combined. Still in (Liu et al., 2017), a time–consuming normalization and registration procedure was used to resize and align images correctly. In this study, for the sake of simplicity and to demonstrate how the improved robustness is due to our model and not to the parameter choice, we decided to use only DWI images, i.e. High b–Value and ADC formats. This set of data does not require any registration or normalization, since all images are captured in the same physical space. Finally, while in (Liu et al., 2017) images considered not adequate for a good training were discarded, we use the entire dataset.

From both the selected modalities, we crop a $32 \times 32 \times 1$ square ROI containing the lesion, according to the coordinate points available in the dataset. Since the dataset is very small, we extract 297 variants for each lesion, cropping the image

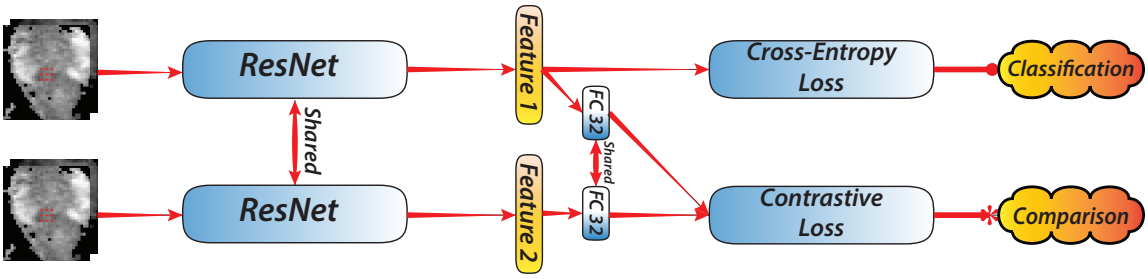


Figure 3.11: The proposed model. We train three different network instances, one for each of the AS, PZ and TZ lesion groups.

with 11 different rotation values $[0, \pm 10, \pm 15, \pm 20, \pm 30, \pm 40]$ and shifting its center point in the range $[-2, 0, 2]$ for each of the three axes, as shown in Fig. 3.10. ADC and High b–Value ROI are then concatenated to obtain a 2–channel image, on which per–channel normalization and mean subtraction are applied. Finally, an online image augmentation, consisting of random flip, shear and zoom, is used to further improve training performance. During the evaluation phase, the 297 variants are averaged to obtain the final prediction, as in (Liu et al., 2017).

The model Our proposed siamese network has a ResNet (He et al., 2016) backbone, which, thanks to the presence of skip connections, has proven to be one of the most effective models in image classification. Siamese networks can be trained based on different losses, depending on the problem to be solved. For example, in one–shot learning (Koch et al., 2015), the cross–entropy loss

$$L_e = \sum_{i}^{n} \sum_{k}^{K} -y_i^k log(\tilde{y_i^k}) \tag{3.9}$$

is used, where n is the dataset dimension, K the model classes, y_i the target, and $\tilde{y_i}$ the prediction. Instead, in metric learning (Appalaraju and Chaoji, 2017; Wang et al., 2014; Zagoruyko and Komodakis, 2015; Zhang et al., 2016), the contrastive loss (Hadsell et al., 2006) defined in Eq. (3.2) is employed to provide a real metric.

In our model, we combine the cross–entropy L_e and the contrastive loss L_c as follows:

$$L(W, X_1, X_2, C_1) = \alpha L_c(W, Y, X_1, X_2) + \beta L_e(W, X_1, C_1) \tag{3.10}$$

with α and β weighting the relative importance of the two losses. In Eq. (3.10), C_1 is the class of the query image and Y is the target for the pair (X_1, X_2) in the contrastive loss, defining the similarity between two images, based on the class correspondence.

As far as we know, there are no examples in the literature of the combined use of cross–entropy and contrastive loss, although a related approach is presented in (Baddar et al., 2017), where two loss functions — i.e. the standard cross–entropy

Prostate Zone	**ResNet Version**	**# of filter in the first block**
Aphical (AS)	8	12
Peripheral (PZ)	20	16
Transition (TZ)	14	12

Table 3.2: Details of the ResNet architectures used in the experiments.

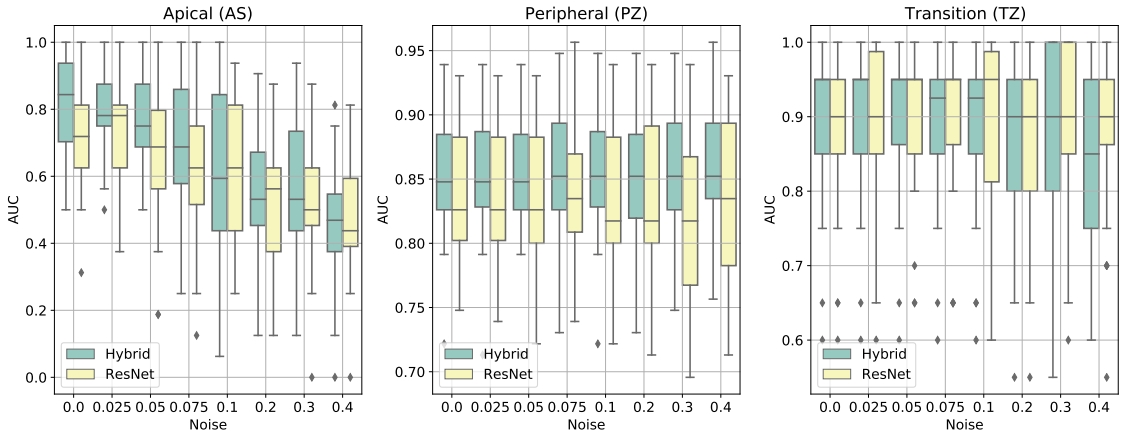
and a customized loss based on the Euclidean distance — are blended for training a siamese network for facial expression recognition.

In this research, we prove that the property of moving patterns across the feature space, making them closer or farther depending on their similarity, can improve the robustness of a prostate lesion classifier with respect to input perturbations. The proposed architecture is shown in Fig. 3.11.

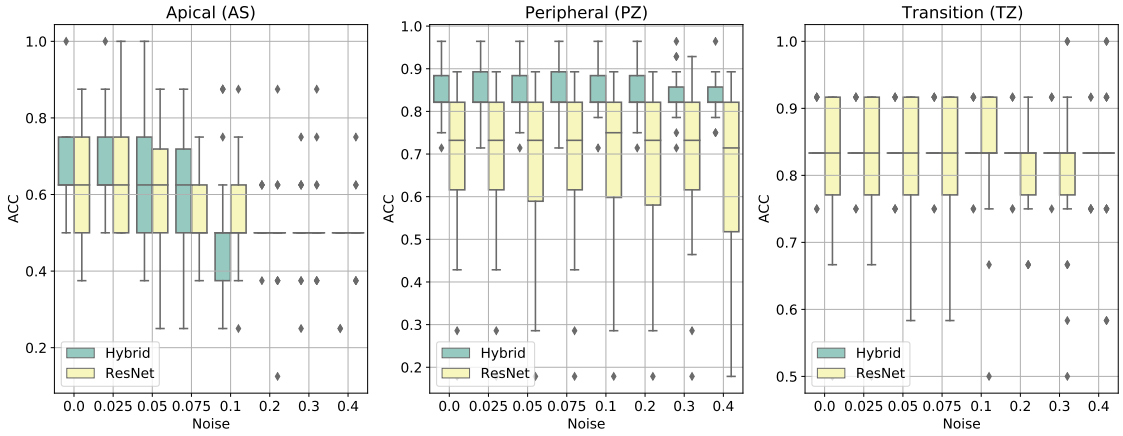
Experimental Setup In our experiments, which aim to asses the robustness of the proposed model, the siamese neural network trained on the combined loss function is compared with a standard ResNet (He et al., 2016) (able to process one input at a time), having the same architecture and exploiting the cross–entropy loss function. After training, we performed two different tests. The first experiment was based on applying a Gaussian noise, with zero mean and standard deviation varying in the range of $[0, 0.025, 0.05, 0.075, 0.1, 0.2]$, to the inputs. This allows to measure the drop in performance in order to evaluate the network robustness. Subsequently, a test was performed by applying an FGSM adversarial attack (Goodfellow et al., 2014) to the network and measuring how much the image has to be perturbed in order to change the predicted class. The results are evaluated measuring the $\|\cdot\|_\infty$ between the original image and the corresponding adversarial example.

Since lesions located in different positions of the prostate have different appearance and class distribution, we build three different models, one for each zone, i.e. AS, PZ and TZ, to better address the three cases. Relevant features of the three different ResNet architectures can be found in Table 3.2 (while more details about the ResNet can be derived from the original paper (He et al., 2016)).

All the hyperparameters were instantiated by splitting the validation set with respect to each of the three cases (AS, PZ and TZ), maintaining the same class distribution as in the training set. Learning is carried out with the Adam optimizer (Kingma and Ba, 2014), with an initial learning rate of 10^{-4}, decreased by a factor of 0.5 after 5 epochs with no improvement in the loss. The training is terminated based on an early stopping procedure, with a patience of 20 epochs, saving the best model according to the validation loss. Parameters α and β are set to 0.25 and 1, respectively, maintaining predominant the classification performance. We trained



(a)



(b)

Figure 3.12: Difference in AUC (a) and accuracy (b) as noise increases.

the networks on 25 instances for each of the three zones on an NVidia GTX 1080TI. The code was implemented in Keras.

Results

We have run 25 experiments for each of the three prostate areas, for a more reliable model assessment. For each zonal architecture, the corresponding performance reduction on the validation set is calculated, by varying the variance of the Gaussian noise added to the input. The obtained results, concerning the AUC and the accuracy, are plotted in Fig. 3.12, reporting the mean (horizontal line), the quartiles (boxes), the rest of the distribution (whiskers) and outliers (diamond) for the 25 run of each configuration. Fig. 3.12 shows how hybrid models achieve better performance with respect to the ResNet case, particularly for the AUC metric, independently from the noise injection level. Next, we tried to cheat our model (and

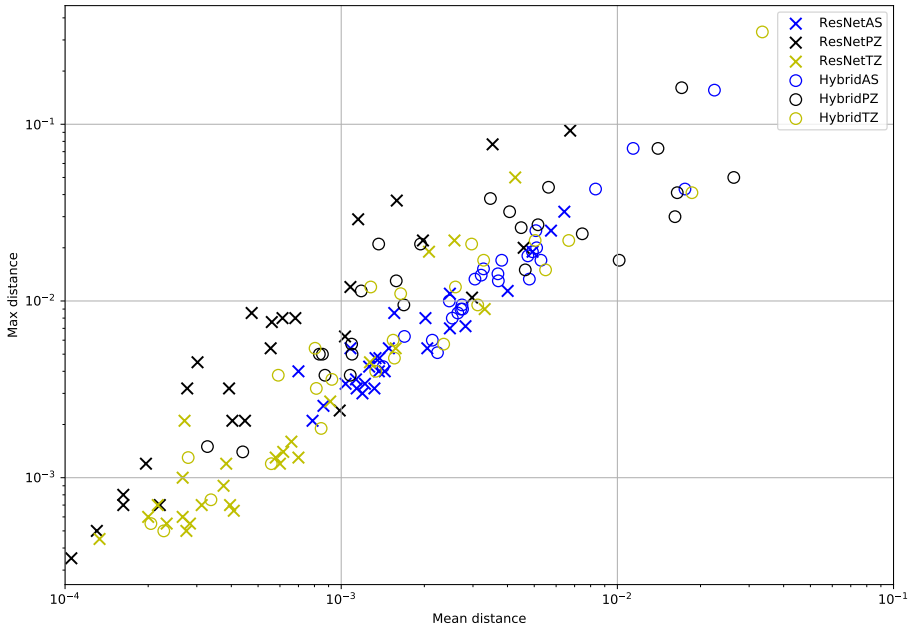


Figure 3.13: Average and maximum distance returned by the FGSM adversarial attack, for 25 training runs.

the baseline) using the FGSM (Goodfellow et al., 2014) adversarial attack, computing the ∞–norm between the original image and the one produced by FGSM. This gives us a measure of how easy is to make the classifier change the predicted class by modifying the input. The average of this distance, together with its maximum, is shown in Fig. 3.13 on a logarithmic scale. The high number of circles in the top–right part of the figure reveals how the hybrid model needs more FGSM iterations to be cheated than the corresponding ResNet, meaning that it reports greater average and maximum distances between the original image and the corrupted one.

Averaging the output of the 5 best models according to the validation set for each prostate zone, and letting the three ensembles respond for the corresponding test lesion zone, the hybrid siamese network got an AUC of 0.8, improving the ResNet (ensembled in the same way) by 0.05.

Conclusions

Results reported in Figs. 3.12–3.13 prove that our hybrid siamese network is more robust than the corresponding standard ResNet. This has been demonstrated in two completely different ways, i.e. by adding Gaussian random noise and fooling the network with adversarial attacks, and both tests confirm the benefits of *learning by comparison* to improve the model robustness. Indeed, having a more robust prediction could save a lot of time and could be very beneficial in medical practice,

avoiding the repetition of some scans, and also improving the confidence of radiologists in automatic diagnostic tools.

As a matter of future work, refining the pre–processing phase — possibly based on human experts' suggestions — is a fundamental issue for obtaining further results in terms of robustness and classification performance.

3.3 Prostate MRI registration using siamese metric learning

In this section, we propose a new method for intra–procedural prostate MRI registration based on siamese neural networks (Bromley et al., 1994). The first step in our research consists to randomly augment (i.e. modify) the original intra–procedural prostate MRIs, producing several variants of the same image. The siamese neural network equipped with the contrastive loss described in Eq. (3.2) is designed to choose the most similar pair between the reference image and all the images belonging to the set of augmented images.

Results prove that this simple scheme achieves a better performance than a registration process based on the SimpleITKv4. We also prove that this model performs better than an available deep CNN (Kuang and Schmah, 2019) for registering brain MRI images from the MindBoggle-101 dataset (Klein and Tourville, 2012)[6].

Three different ways of building couples for the siamese training are compared, based on the Intersection over Union (IoU), the Dice Score (DS), and the Mutual Information (MI) metrics, and we conclude that choosing similar and dissimilar samples for the training based on MI is the best option. This fact is important since it eliminates the need for segmentations, leading to an unsupervised method.

Furthermore, we show the effect of the size of the augmented set on performance, discovering that 18 variants is the optimal balance between numerosity and feasible complexity. Lastly, this research tests the effect of the number of similar and dissimilar samples per slice, determining that performance is not affected by this parameter and therefore suggesting the use of only the most similar and dissimilar images.

Dataset and image preprocessing

The dataset used is derived from (Fedorov et al., 2012). It is composed of anonymized pre– and intra–procedural MRIs from 10 patients. Three sets of prostate gland segmentations are manually prepared by the same number of raters, two of whom have more than ten years of experience in MRI rating. Moreover, also per–patient landmarks are present in the dataset. The images are acquired in the axial position with the standard T2–Weighted (T2W) modalities. The original pre–procedural MRIs had a size of $512 \times 512 \times 30$. The voxel spacings were 0.3125, 0.3125, and 3, respectively. The original intra–procedural MRIs had a size of $320 \times 320 \times 40$. The voxel spacings were 0.5, 0.5, and 3, respectively. To normalize the images while maintaining as much information as possible, they are resized to $128 \times 128 \times 128$, with voxel spacings of 1.10021, 1.0021, and 0.856299, respectively. Additionally, all voxel

[6]The proposed method focuses on optimizing a deep neural network that directly outputs displacement fields.

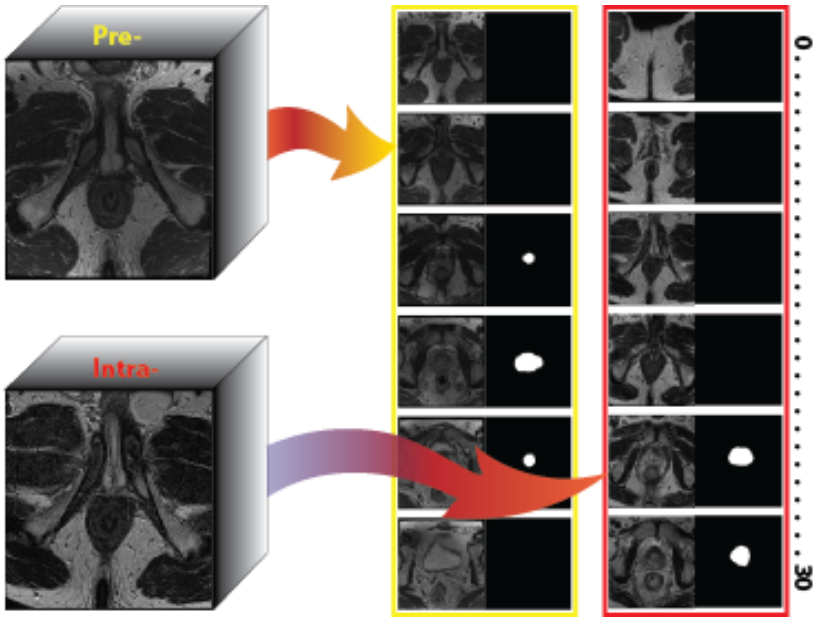


Figure 3.14: Example of P pre– (yellow) and intra–procedural (red) images.

values within each segmentation image are converted to 0 or 1. For training the network, pre–procedural and intra–procedural MRI slice pairs that do not contain any segmentation information (i.e. images containing only background) are not taken into account. The first seven patients are used for training, patient 8 for validation, and the remaining two for testing the model. Fig. 3.14 shows an example of pre–procedural and intra–procedural MRI images.

Experimental setup

In this section, we describe the convolutional siamese network, the overall framework, the baseline used, and finally the research question answered.

The siamese model The core of this model is represented by convolutional siamese neural networks (Bromley et al., 1994; Koch et al., 2015). The details of the network

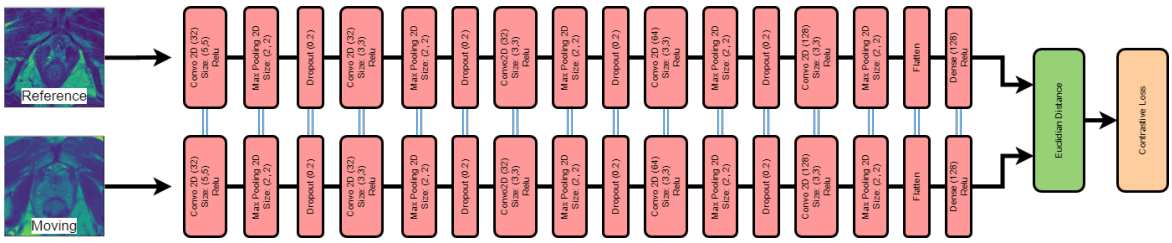


Figure 3.15: Structure of the siamese network. Blue lines mean that parameters are shared between the two branches.

are provided in Fig. 3.15. For each of the two inputs, an embedding is created, and then the euclidean distance between them is used to evaluate the contrastive loss defined in Eq. (3.2).

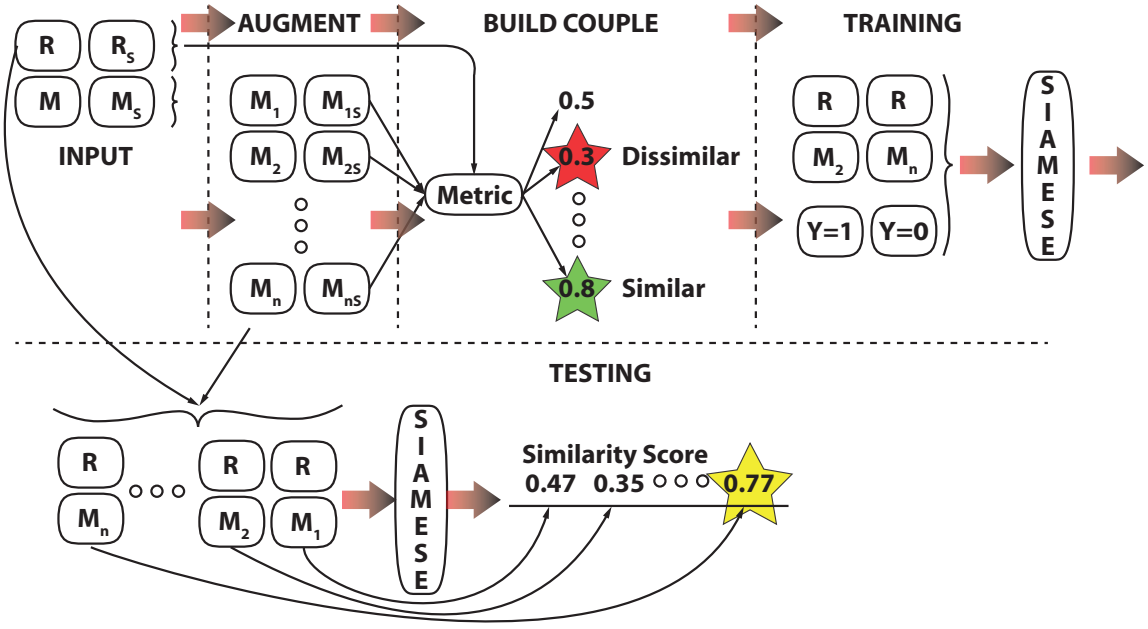


Figure 3.16: Illustration of the overall method. There are three distinguished training phases: the moving image (M) (and the corresponding segmentation Ms) augmentation, the couple set formation, i.e. finding similar and dissimilar samples based on a metric, and the training. During testing, the network is asked to choose the best registered candidates from the set M.

General framework The overall framework is composed of three different steps, and it is illustrated in Fig. 3.16. The first step consists of randomly producing augmented samples of the intra–procedural images. In particular, the influence of the number of augmented samples on the performance of the model is evaluated by varying this number per MRI slice in the set $(9, 18, 27, 36, 45)$. The applied augmentation consists of a combination of affine transformations, such as scaling, rotation, and translation along the x and y axes. Specifically, the values used for augmenting the slices are $(-0.1, -0.05, 0, 0.05, 0.1)$ for rotation, $(-0.25, 0, 0.25)$ for translation along the x–axis, $(-0.25, 0, 0.25)$ for translation along the y–axis, and $(0.8, 1, 1.2)$ for scaling along the y–axis. In addition, elastic distortion (Simard et al., 2003) is applied. The first step of this transformation is to produce a random displacement field $\Delta x(x, y) = rand(-1, 1)$ and $\Delta y(x, y) = rand(-1, 1)$, where $rand(-1, 1)$ means a random number in the range $(-1, 1)$. Those displacement fields are convolved with a gaussian of standard deviation σ and, finally, an intensity factor α is used to control the magnitude of the distortion. In particular, we used $\alpha = 1.2$ and $\sigma = 0.7$.

Subsets of size $(9, 18, 27, 36, 45)$ of these augmented slices are grouped together randomly, creating the selection of images the model can choose from. In the training procedure, the same set of parameters is used to augment the corresponding intra–procedural segmentations (necessary for building couples based on DS and IoU) exactly in the same way as for the images from which they originated.

The second step is only necessary for training and it is achieved by couple creation. Essentially, a set of informative pairs of samples are built to efficiently train the siamese network to distinguish between similar and dissimilar patterns. For this purpose, for each pre–procedural image, the most similar and dissimilar augmented intra–procedural image is selected according to three different metrics, two of them evaluated given the corresponding segmentation (i.e. DS and IoU), while the Mutual Information (MI) is evaluated directly from the image. In summary, this operation produces a set of similar and dissimilar pairs used for training. This set is perfectly balanced.

The third step is the model training, making the network able to choose the most similar intra–procedural image according to the corresponding pre–procedural image. For the training process, the model goes through 100 epochs with a batch size of 64. The optimization is done with Adam optimizer (Kingma and Ba, 2014) and with a learning rate of 0.0001. Out of the 10 cases of data used for this experiment, cases one through seven are assigned for training the model, and case eight is used for validating the model, while the remaining two are used for testing. All the hyperparameters are chosen according to the validation set.

During testing, for each of the two cases, individual slices are created, which are augmented by means of the same method as used for the training slices. For each unaugmented intra–procedural slice, each set of augmented slices is fed into the network, as well as its respective pre–procedural slice. The images reporting the highest similarity score are selected, and finally the model is evaluated based on different metrics comparing the pre–procedural slice segmentation and the segmentation of the slice the model deems as most accurately registered. Four different metrics are used for evaluating each pair of registered images: ROI DS and ROI IoU, as well as DS and IoU, that include every pixel within each slice. The ROI score only considers the overlap of the segmentation, while the other metric also considers the overlap of the background of each segmentation, not reducing it to a ROI.

Baselines The first baseline model, used to compare the siamese model results, comes from the SimpleITK python library, a toolkit commonly used for analyzing medical images, such as MR images and CT scans. In addition to image augmentation operations, which are used to augment the data, the SimpleITK toolkit includes a registration method for 3D images. The model takes in a fixed and a moving image, and interpolates each of the images so that it can accurately read the images

and transfer them to a virtual domain. As a result, the moving image can be manipulated in a domain separated from the image it came from. From this virtual domain, a transformation is applied to the moving image, creating a registered image. Then, the transform is optimized based on how well the image was registered. This process of updating the transform and analyzing the results is repeated until the final (most accurately registered) image is formed. The instance of this model, which is used on the prostate MRI data, was trained for 2000 epochs with a learning rate of 0.05.

The second baseline model comes from the research described in (Kuang and Schmah, 2019). Their FAIM is a deep learning model trained using the Mindboggle-101 dataset (Klein and Tourville, 2012), which contains 101 MR images. The focus of this research was to create a model that had a low level of "foldings" within the deformation, caused by the fact that Jacobian determinants used to create the deformations were negative. To accomplish this task, the model included a regularization method to increase the loss of displacement fields produced from the spatial deformation module, where the Jacobian determinant was negative. During training, a pair constituted by a moving and a reference image was fed into the model. The moving image was fed into the spatial deformation module, performing the transformations created by the network on a sampled moving image. Finally, the error was calculated using a cross correlation loss with two regularization methods, including the negative Jacobian determinant. To apply this model to the prostate MRI data, the siamese model is trained with the same number of epochs (10) and the same learning rate (10^{-4}). This model was chosen as a reference because it is one of the few with open access to the associated code. Also this model processes single slices and not the entire 3D volume.

Research questions The fundamental questions addressed in this research are the following:

- Can a siamese network, in a convenient to use environment, outperform other common and deep learning algorithms in the task of prostate MRI interventional registration?
- Which is the best strategy to build the training couple necessary for the siamese network? Three policies are tested, consisting of selecting the most similar and dissimilar samples using DS, IoU and MI. While the latter policy allows us to have a complete unsupervised model, the other two require prostate segmentation.
- Which is the best size of the augmented image set from which the siamese network can choose the best candidate to be the registered image? Each slice is augmented by means of five different values (9, 18, 27, 36, 45).

Model	***IoU* (*roi*)**	***DS* (*roi*)**	***IoU* (*all*)**	***DS* (*all*)**
SimpleItk	47.9	62.9	83.1	87.3
FAIM	37.5	50.8	92.1	92.1
Siamese	69.2	80.0	97.1	97.1

Table 3.3: Results from the siamese model and the two considered baselines.

- Is it sufficient for the training to pick just the most similar and dissimilar images or should an extended set be considered? The selection of up to 5 similar and dissimilar images is tested.

Results

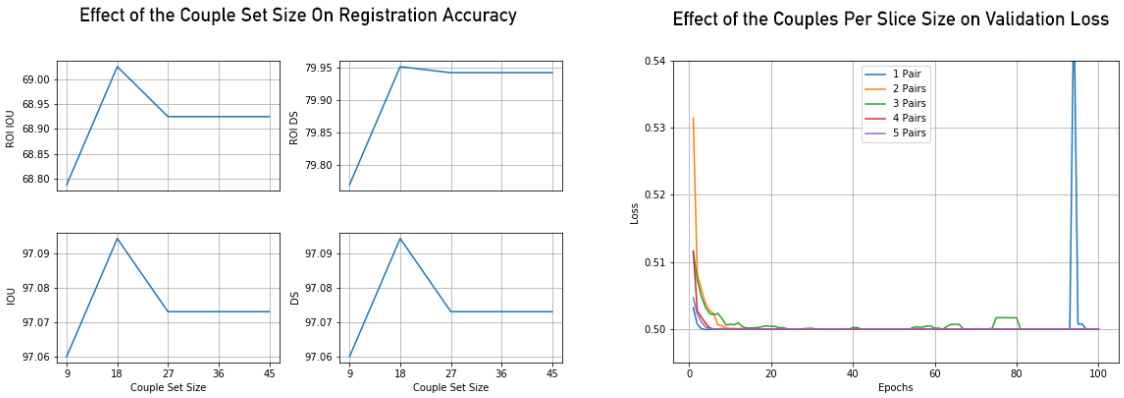


Figure 3.17: The effect of the size of the couple set, evaluated with all the four scores (left), and that of the pair per slice in the validation loss (right).

The results reported in Tab. 3.3 show that the siamese model, in the best performing setting, outperforms the two baselines by a large margin.

The following discussion addresses the best policy for building the couple set. All the policies result in almost equal outcomes. In particular DS and IoU produce exactly the same couple set, while MI differs from the other two in some of the choices. This suggests that using MI is more advantageous, as it eliminates the need for segmentation, yielding a completely unsupervised method.

Next, selecting from a total of 18 possible candidates for the best augmentation yields the best results. Increasing this number makes the decision too difficult for the network, reducing performance. The left part of Fig. 3.17 reports the score for all the metrics according to the size of the couple set. The results show that considering more couples from each slice to be registered has no effect on the accuracy of the model. Even though more data are being used for training, any added pairs of images beyond the most accurately registered pair reduces the decisiveness of the siamese model.

Therefore, when only one image is selected as the most accurately registered image during the evaluation process, the additional pairs for training do not have a significant impact on the model decision. The validation loss reported in the right part of Fig. 3.17 reveals how selecting just the most similar and dissimilar samples for training produces the best validation loss, even if the difference is very low. Lastly, Fig. 3.18 shows an example in which the siamese network selects the correct

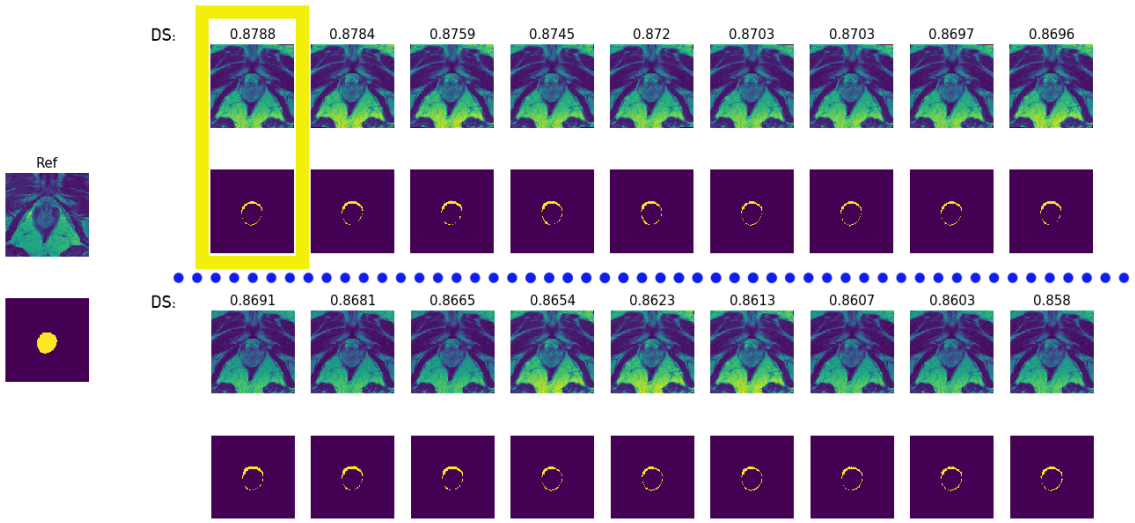


Figure 3.18: Qualitative results from the siamese model. The DS is reported for each candidate, while yellow rectangles depict the augmentation selected by the network. For each candidate, the difference between its segmentation and the reference segmentation is reported (even rows).

candidate among the set of 18 possible choices, together with the considered slice (odd rows); the difference between its segmentation and that of the reference image is also reported (even rows).

Conclusion

This study sheds light on the process of intra–procedural prostate MRI registration. This process is fundamental in clinical practice. Based on siamese neural networks, the model made in this research is able to outperform two competing baselines, demonstrating the power of siamese networks and metric learning for the case of a very restricted dataset as in prostate MRI registration. This research also reveals how informative training couples can be created just based on mutual information, avoiding the use of segmentation required by the intersection over union or Dice metric. This allows to have a completely unsupervised method. A large number of registration candidates is not necessary to achieve good performance. In fact, the best score was reached with 18 possible candidates. Finally, this experimentation

proves that increasing the number of pairs of pre–procedural and intra–procedural images gained from a set of augmentations does not have an effect on the model performance. Therefore, the new proposed model is innovative in the field of radiology. Its simplicity together with low training time and accurate performance could help in the process of guiding biopsy or surgical intervention in general. It would be useful to test the model in a real surgical scenario gathering the feedback of experienced radiologists, to further improve its effectiveness. Another interesting test would involve the use of a pre–trained neural network as a backbone to the siamese network, with further fine tuning to match the data particular use.

Chapter 4

Recurrent and recursive networks for medical image processing

This chapter is dedicated to the study of some applications of recurrent and recursive neural networks to medical images. We will show that these architectures are advantageous both in terms of computational resources and required training data. Indeed, in the next section, we will demonstrate that Convolutional Fully Recursive Perceptron Networks (C–FRPNs), which mix CNNs and recursive models, outperform a CNN having the same complexity — particularly when the dataset is small — on several benchmarks including the case of the detection of cutaneous melanoma. Finally, in the last section, we present the application of a recurrent neural network to age estimation from 3D brain MR images. Again, the reported results show better performance than 3D CNNs, which may suffer in the case of small datasets and limited computing power.

4.1 A study on the effect of recursive convolutional layers in CNNs

This section presents an empirical study on the effect of recursive layers applied to convolutional neural networks (CNNs).

The first attempt to insert recurrent connections into a CNN dates back to 2015 and was proposed in (Liang and Hu, 2015), which describes a CNN with an initial convolutional layer, followed by four recurrent layers in the intermediate stages, where the computation of the hidden states is iterated for a fixed number of times for each of the recurrent layers. The method showed remarkable performance on four different benchmark datasets, outperforming baseline CNNs that feature the same number of weights and layers. However, the study in (Liang and Hu, 2015) is limited by the fixed number of stages/iterations, which essentially prevents the

network from converging properly, due to the short convergence time; moreover, it does not investigate the possibility of having a mixture of fully recursive convolutional layers and ordinary convolutional layers in the CNN.

This study proposes a new model, called Convolutional neural network with Fully Recursive Perceptron Network (C–FRPN), in which some or all the convolutional layers in the CNN can be replaced by their fully recursive convolutional layers (RCLs) counterparts. A RCL is defined as a convolutional layer with its outputs connected directly with its inputs. The RCL can be considered as a generalization of the recursive connected hidden layer in a multilayer perceptron (Hagenbuchner et al., 2017) to a convolutional layer in the CNN. The effect of having a recursive layer instead of a standard hidden layer is that the recursive layer may be considered as the equivalent of a set of feedforward hidden layers with a data–dependent depth. In other words, a recursive layer represents an expansion of a set of feedforward hidden layers, whose depth depends on each incoming data. To make sense to this expansion, a constraint on having approximately the same number of parameters between the recursive layer and the set of feedforward hidden layers is imposed.

Since a CNN normally has a certain number of convolutional layers, the replacement of each convolutional layer by an RCL may be considered as allowing each input "pattern" to be processed by an equivalent set of feedforward convolutional layers, according to the complexity of the "pattern" itself; here, the term "pattern" is used to denote the input to a convolutional layer [1] and "complexity" refers to the need of an input pattern to be processed by a set of feedforward convolutional layers. Therefore, it is intuitive that if any of the convolutional layer in a CNN is replaced by an RCL, its performance would be at least as good as that obtained by the CNN with approximately the same number of weights, and it would be better if any of the input "patterns" requires a greater number of feedforward convolutional layers, due to its "complexity". However, a few questions arise: (1) how many RCLs should be used to achieve the best performance and (2) what could be the optimal positions of those RCLs in the CNN to achieve the best possible performance relative to the number of RCLs that can be placed?

It can be expected that the answer to these questions would be dependent on the learning problem. For investigating the effects of the RCLs in a CNN, this study will evaluate the algorithm on three different image datasets with significantly different properties, namely, CIFAR-10, a natural image recognition dataset, SVHN, a real–world dataset of house numbers, and ISIC (International Skin Imaging Collaboration), a skin lesion image dataset. By working through these three largely different datasets, some general conclusions could be drawn concerning the answers to the above mentioned questions.

[1] For all convolutional layers apart from the first one, the input is the vecotor collecting the features extracted from the previous convolutional layer.

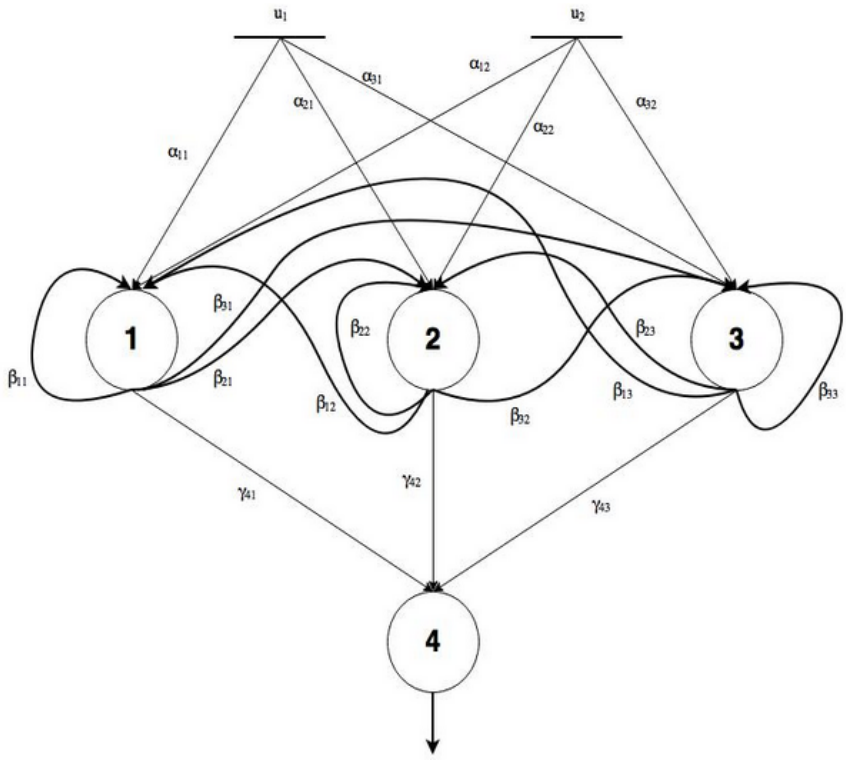


Figure 4.1: A simple example of a 3 neuron FRPN, showing the full connectivity of the network.

Fully Recursive Perceptron Networks

The first fully recursive neural network model was proposed in (Hagenbuchner et al., 2017). The model extends a standard MLP by adding self–connections and other feedback connections as shown in Figure 4.1.

Formally, given the inputs u_i, $i = 1, 2, \ldots, m$, $u_i \in \mathbb{R}$, a recursive layer having n neurons is described as follows:

$$x_i(t+1) = f\left(\sum_{j=1}^{m} \alpha_{ij} u_j + \sum_{k=1}^{n} \beta_{ik} x_k(t) + b_i\right), \tag{4.1}$$

where x_i, $i = 1, 2, \ldots, n$, is the output of neuron i, b_i is its bias, and α_{ij}, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$, and β_{ik}, $k = 1, 2, \ldots, n$ are its connection weights. The activation function $f(\cdot)$ can be one of the standard mapping used in neural networks, such as sigmoid, hyperbolic tangent, rectified linear unit (Goodfellow et al., 2016), scaled exponential linear unit (Klambauer et al., 2017), exponential linear unit (Clevert et al., 2015), or similar. If β_{ik}, $i = 1, 2, \ldots, n$, $k = 1, 2, \ldots, n$ are all 0, then Eq. (4.1) reverts back to be the classic MLP formulation, with a single hidden layer. The MLP with a single hidden layer being a recursive layer, defined in Eq. (4.1) is called a fully recursive perceptron network (FRPN) (Hagenbuchner et al., 2017).

Eq. (4.1) can be considered as a discretization or an unfolding in time of the continuous time equation $\frac{dx_i(t)}{dt} = f(\sum_{j=1}^{m} \alpha_{ij}^{'} u_j + \sum_{k=1}^{n} \beta_{ik}^{'} x_k(t) + b_i^{'})$, where $f(\cdot)$, $\alpha_{ij}^{'}$ and $\beta_{ik}^{'}$ have similar interpretations to those in Eq. (4.1). This can be shown by using a first order forward difference scheme to approximate the operator $\frac{dx(\zeta)}{dt} \approx \frac{x(\zeta+\Delta)-x(\zeta)}{\Delta}$, where Δ is a very small time step; after some simple algebraic manipulations, $x(\zeta)$ can be absorbed in the $\Delta f(\cdot)$ function, and with an abuse of the notation, $\Delta f(\cdot)$ is of a similar structure to $f(\cdot)$. The output of an FRPN network can be given as follows:

$$y_i = g(\sum_{j=1}^{n} \gamma_{ij} x_j + c_i), \tag{4.2}$$

where γ_{ij}, $i = 1, 2, \ldots, p$, $j = 1, 2, \ldots, n$, are the weights connecting the hidden and the output neurons, and $g(\cdot)$ is a function responsible for the output of the network, commonly a sigmoid or a softmax function for classification tasks, or a linear function for regression tasks.

The peculiarity of this network resides in the feedback connections realized by the weights β_{ik}, $i = 1, 2, \ldots, n$, $k = 1, 2, \ldots, n$. At first glance, this network appears similar to a recurrent neural network. A main difference is that, for the computation of x_i, the input $u(t)$ is always the same, whilst in a standard RNN, the input is time–dependent. Correspondingly, the maximum value of t in RNNs depends on the length of the input data sequence, whereas for FRPNs the maximum value of t is reached when $x_i(t+1) \approx x_i(t)$, $\forall i$. Thus, the FRPN iterates the computation of Eq. (4.1) until the representation of $x(t)$, called state, converges. Due to computational reasons, an upper bound on the number of iterations can be set, to stop the process when convergence does not occur within a given amount of time. If we linearize Eq. (4.1) around an operating point, we would be able to find the eigenvalues of the linearized equation. These eigenvalues will establish the instantaneous behaviour of Eq. (4.1). Therefore, the eigenvalues of the $n \times n$ matrix formed by the weights $\beta_{ik}, i = 1, \ldots, n$, $k = 1, \ldots, n$, while being stable, may have values close to the imaginary axis. Anyway, Eq. (4.1) may take longer time to converge. An upper limit must be imposed on the number of iterations to ensure that this process does not get out of hand under a particular operating condition, since the values of β_{ik} are changed during training and, instantaneously, they may have unstable eigenvalues or stable eigenvalues which are close to the imaginary axis. Moreover, iterating the evaluation of Eq. (4.1) corresponds to unfolding the network for a certain amount of time, realizing a deep network. On the other hand, the depth of the network is not fixed *a priori*, since convergence can occur at different times, and the rate of convergence depends both on the input, the size of the network weights, and on the activation function of the hidden neurons. The FRPN has, thus, a very desirable property of self–adjusting its depth for each input or, in other words, it is able to solve the hard problem of choosing a correct number of hidden layers for a particular learning

problem. The recursive computation of the FRPN is similar to a mechanism used by Graph Neural Networks (Scarselli et al., 2008). In the Graph Neural Network (GNN) model, an iterative mechanism is present in order to encode dependencies among the nodes in an input graph. The input of GNNs can also be considered static if the graph input does not change with time. Also in the case of GNNs, the computation is iterated for an indefinite number of times defined by the convergence of the state representation. In this case the input is not static and collects information from neighboring nodes allowing the information diffusion process specific of the GNN model, while in the case of FRPNs the input is static.

The C–FRPN model

The output of a convolutional window of dimension $N \times N$, with weights w_{ij}, on the input $u_{ij}, i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, N$, at the position (s_1, s_2) is given by:

$$x_{s_1,s_2} = \sum_{i=-N/2}^{N/2} \sum_{j=-N/2}^{N/2} w_{ij} u_{s_1+i,s_2+j}$$

Then, in analogy with Eq. (4.1), the Recursive Convolutional Layer (RCL) is obtained as follows:

$$\begin{aligned} x_{s_1,s_2}(t+1) = f\Bigg(& \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_{i,j} u_{s_1+1,s_2+j} + \\ & \sum_{k=1}^{N} \sum_{k=1}^{N} \beta_{i,j} x_{s_1+i,s_2+j}(t) + b \Bigg), \end{aligned} \tag{4.3}$$

where the indexes s_1 and s_2 range over the entire input image. Moreover, if m represents the number of feature maps associated with the RCL, then x_{s_1,s_2} is an m–dimensional vectors, α_{ij}, β_{ij} are $m \times m$ matrices of constant but unknown scalars and b is a m–dimensional vector of biases. Note that the recursion of x_{s_1,s_2} occurs within the $N \times N$ receptive field window, where x_{s_1,s_2} occupies the center; finally, N is an odd integer.

In this section, we propose to exploit the RCL in a network architecture called Convolutional FRPN (C–FRPN). In details, the proposed C–FRPN consists of an initial convolutional layer, followed by four intermediate stages, which can either be convolutional layers or RCLs. Then, a pooling layer is inserted after the second intermediate stage, followed by a global average pooling, which provides the inputs for a final softmax fully connected layer, producing the outputs (see Figure 4.2). Exploiting the C–FRPN, this study investigates the fundamental questions raised in Section 4.1.

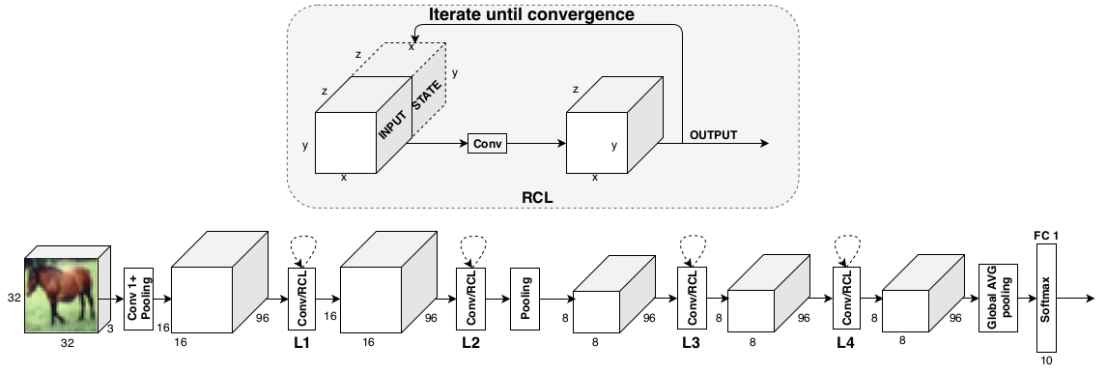


Figure 4.2: The C–FRPN network: each recursive convolutional layer unfolds until convergence of the state representation.

The C–FRPN is close to the architecture proposed in (Liang and Hu, 2015), but it is more flexible and differs for two reasons. First, the RCLs in the C–FRPN iterate until the state converges, whereas the "recurrent" convolutional layers in (Liang and Hu, 2015) iterate exactly three times. Moreover, our model allows to replace any convolutional intermediate stage with an RCL, whereas in (Liang and Hu, 2015), all the four convolutional layers in the intermediate stages must be replaced by "recurrent" layers.

Note that the RCLs in the C–FRPN do not have a fixed number of iterations, thus allowing the architecture to decide when the intermediate state has converged. This allows to skip the step of determining the optimal number of iterations, leaving the network to adjust its depth for each learning problem and for each input. The network can thus have different depths for different problems or even for different samples. The convergence is estimated by computing the mean squared error between the state representation at the current iteration and at the previous one. If the difference is below a prescribed small threshold, then we assume that the state has converged to a stable point.

Experiments

In order to discover the fundamental properties of RCLs when applied to practical problems, we follow two lines of investigations. First, the effectiveness of RCLs in networks of different size is investigated. This is performed via a set of experiments on several learning problems solved by using a variety of network sizes to compare performance of C–FRPNs and CNNs. The dimension of the network, i.e. the number of parameters, is adjusted to be approximately the same in both the C–FRPN and the CNN architectures. Secondly, we vary the number and the position of RCLs, taking care to maintain approximately constant the number of parameters and layers, for fair comparisons. This set of experiments is designed to sharpen our

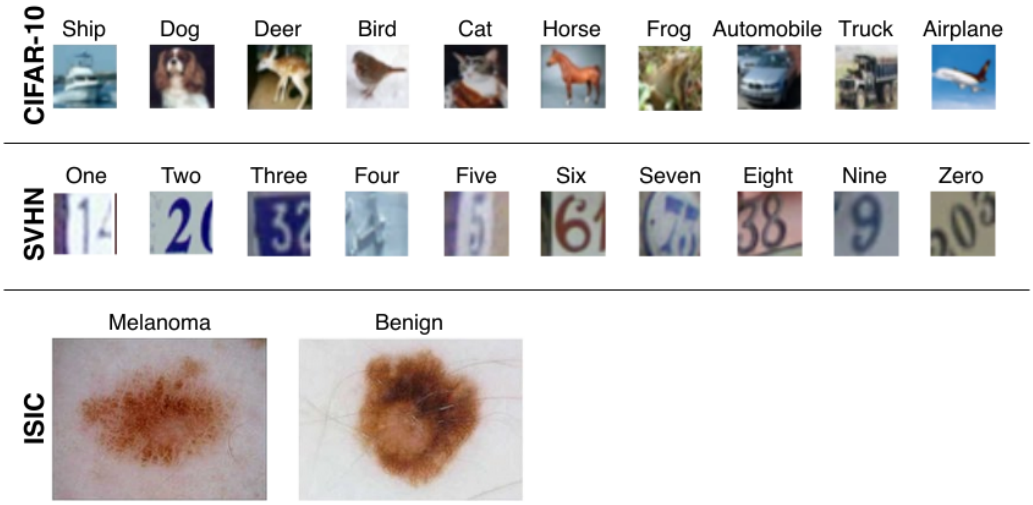


Figure 4.3: Sample images representing different classes for the three datasets.

understanding of where RCLs are most beneficial in terms of the generalization accuracy on the test set. The overall setting is exhaustive, as we investigate all possible permutations and combinations of RCLs inside the C–FRPN, to discover their effectiveness. The following sections offer a more complete description of the datasets used and of the setup of the experiments.

Datasets In order to obtain an assessment of the general characteristics of the C–FRPN — particularly the dependence of the convergence rate of RCLs on the data —, we tested the model on 3 datasets from different domains, having various peculiarities, and spanning a range of different properties. These datasets, namely CIFAR-10, SVHN, and ISIC, are described in the following.

CIFAR-10 (Krizhevsky et al., 2009): This dataset consists of 60,000 colored images of size 32×32 pixels, 50,000 of which were earmarked for training, while the remaining 10,000 were used for testing purposes. We selected 5,000 images from the training set to serve as a validation set, to determine the network hyperparameters, f.i., the learning rate. Once the best set of hyperparameters was determined, we used the entire training set for training the model, then evaluating its generalization capability on the test set. The procedure is the same as in (Liang and Hu, 2015). The images were categorized into 10 classes, namely *airplane*, *automobile*, *bird*, *cat*, *deer*, *dog*, *frog*, *horse*, *ship*, *truck*. This can be considered an example of a problem involving complex natural images. Some sample images are shown in Figure 4.3.

SVHN (Netzer et al., 2011): This is another well known benchmark for testing image classification models. It is composed of natural images showing the street number where a house is located. The task is to predict the central digit. All the RGB–colored images are of size 32×32 pixels. The set consists of 73,257 images

for training and 26,032 for testing, divided in 10 classes representing the ten digits. Some examples are shown in Figure 4.3. For simplicity, we reused the same set of hyperparameters used in the evaluation of the CIFAR-10 dataset so that we do not need to extract any validation set.

ISIC (Codella et al., 2018): The International Skin Imaging Collaboration (ISIC) defines a project involving academia and industry to encourage the developing of digital imaging algorithms to help diagnosing skin cancer (melanoma). ISIC maintains a publicly accessible archive of skin images to train and test new diagnostic models (Codella et al., 2018). The currently available dataset consists of more than 23,900 images showing skin lesions divided into benign (about 19,300 samples) and malignant (about 2,200 images, where most of the samples are melanomas and some of them are carcinomas). To facilitate learning, we balanced the dataset by undersampling, thus resulting in a dataset of only 4,100 images as in (Bonechi et al., 2019). The test set was composed of 20% of the dataset; 10% of the remaining images were used for the model validation. The images are in various formats, since they were acquired with different devices. To train and test our model, we resized each image to 224×224 pixels as in (Bonechi et al., 2019). This dimension is popular, since it is the one adopted by Image-Net Russakovsky et al. (2015). Images have always the same background (skin). Hence, the peculiarities of this dataset is to recognize very fine grained details of the images. Two examples from the dataset are shown in Figure 4.3.

Experimental settings To allow a fair comparison, the C–FRPN uses an architecture similar to that described in (Liang and Hu, 2015). The number of layers is thus 5, where the first layer is just a common convolutional layer, and the following 4 intermediate stages will be either convolutional layers, RCLs, or a mixture of them. In the first part of our experiments, we set the same number of feature maps for all layers, as in (Liang and Hu, 2015), so as to obtain networks where the intermediate stages have all the same width. To change the size of the network, we varied the number of feature maps in RCLs in the set of values [96, 85, 74, 60, 30, 15]. The corresponding baseline architecture is composed by 5 convolutional layers having the same number of weights: to achieve this, the feature maps were set to [135, 120, 104, 85, 42, 21], respectively. RCLs in a C–FRPN have more weights than their convolutional layer counterpart with the same number of neurons, due to the recursive nature of the RCL. Let ν be the number of feature maps in each convolutional layer or RCL. Actually, the total number of parameters in a common convolutional layer is $q \times \nu \times 3 \times 3$, where q denotes the number of inputs, and 3 is the receptive field size, whereas the parameters in an RCL are $(q + \nu) \times \nu \times 3 \times 3$, where the additional ν is due to the state dimension, which is fed back to the input. For example, a RCL with 96 feature maps has the same number of adjustable weights as a convolutional layer

Table 4.1: A table illustrating the detailed number of feature maps in each layer for all the possible combinations of recursive convolutional layers and convolutional layers in a C–FRPN.

Layer	**Combination**														
	L_1	L_2	L_3	L_4	L_1L_2	L_1L_3	L_1L_4	L_2L_3	L_2L_4	L_3L_4	$L_1L_2L_3$	$L_1L_2L_4$	$L_1L_3L_4$	$L_2L_3L_4$	$L_1L_2L_3L_4$
L_1	60	85	85	85	60	60	60	85	85	85	60	60	60	85	60
L_2	85	60	85	85	60	85	85	60	60	85	60	60	85	60	60
L_3	85	85	60	85	85	60	85	60	85	60	60	85	60	60	60
L_4	85	85	85	52	85	85	52	85	52	52	85	52	52	52	60

with 135 feature maps. For the sake of completeness, Table 4.1 describe a C–FRPN architecture having $264k$ parameters, considering x as the number of feature maps, y as the number of RCLs, with $4 - y$ number of convolutional layers, followed by a global average pooling and a softmax layer.

The detailed architecture of other networks resulting from various combinations and permutations of RCLs and convolutional layers can be computed using the guidelines leading to the construction of Table 4.1. Thus, this setting allows us to make a fair comparison of our C–FRPNs and the networks used in (Liang and Hu, 2015), since we will use the same number of adjustable parameters as well as the same number of layers.

In the second part of our experiments, we maintain a five layer network, using convolutional layers and RCLs in all possible permutations and combinations of their locations in the four intermediate stages. This was performed keeping the number of parameters and feature maps fixed to the values defined in the previous set of experiments.

In all the models, the kernel size of the first convolutional layer was 5×5 with a stride of 1, while all the other layers have a kernel with size 3×3 and a stride equal to 1. We used a max pooling, with a kernel size 2×2 and a stride of 1, after the first and the third layer, respectively, while a dropout, with a forget rate of 0.5, was deployed after each layer, but not in the output layer. Local response normalization (Krizhevsky et al., 2012) was used after each iteration of each RCL, and the state was considered stable if the Euclidean distance between its current representation and the previous one was less than 0.1, although we stopped the iterations after a maximum of 8 steps, as a fail–safe measure in case of very slow convergence and to control the turn–around time needed for the experiments.

We exploited image augmentation as in (Liang and Hu, 2015) for CIFAR-10 and SVHN, respectively, while we used random rotation together with horizontal and vertical flips for the ISIC dataset. We employed the Adam optimizer (Kingma and Ba, 2014), with a learning rate of 1×10^{-4} and a weight decay of 5×10^{-4}. The batch size was set to 128 for CIFAR-10 and SVHN, and to 24 for the ISIC dataset, respectively. The number of training epochs was set to a maximum of 500 for CIFAR-10 and

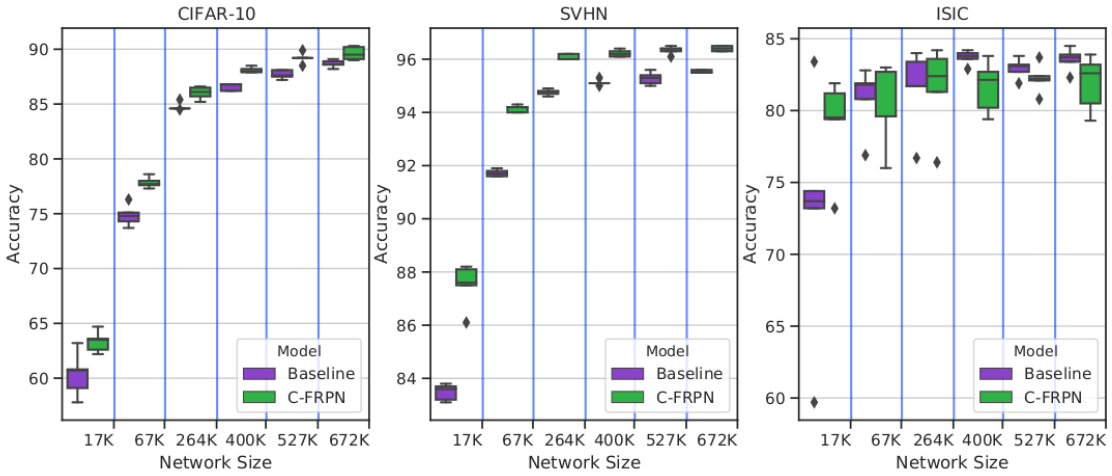


Figure 4.4: Results obtained for the three datasets by varying the size of the network. The box shows the quartiles of distribution over five runs, the whiskers report the rest of the distribution, while the points represent outlayers.

SVHN, while for ISIC we used early stopping with a patience factor of 20 epochs. The hyperparameters for CIFAR-10 and SVHN were defined using a validation set, which was later joined with the training set for the final training runs. In this manner, on CIFAR-10 and SVHN, we adopted the same learning procedure of (Liang and Hu, 2015), whereas for ISIC, which was not used in (Liang and Hu, 2015), we adopted a standard early stopping practice. Experiments, varying the size of the network, were repeated 5 times using different initial conditions and, in the following, we will report the computed average and standard deviations.

Performance comparisons between baseline CNNs and C–FRPNs Firstly, we explore how the behaviour of C–FRPNs varies with respect to the size of the network, i.e. the number of parameters. We report the results obtained using RCLs having a number of neurons chosen from the set $[96, 85, 74, 60, 30, 15]$. We compare these results with those achieved by a common network with standard convolutional layers and with the same number of parameters. Note that, for comparison, we are controlling the number of parameters and not the number of neurons. This is because RCLs use more parameters than a convolutional layers.

Figure 4.4 summarizes the results for this setting. The diagram comprises the information obtained from two sets of experiments comparing the two architectures, one with four intermediate convolutional layers, called "baseline" for convenience, and the other with four intermediate RCLs, called C–FRPN. The number of parameters in each intermediate stage is the same between the two sets of experiments. The number of parameters can be selected from the set $[17k, 67k, 264k, 400k, 527k, 672k]$, where $k = 1,000$. The horizontal axis in Figure 4.4 indicates the total number of

parameters, while the y–axis shows the generalization accuracy on the test set when each of the two architectures is applied to the dataset. For the same number of parameters, Figure 4.4 shows two boxes, one corresponding to the baseline, and the other corresponding to the C–FRPN. Thus, for example, in Figure 4.4 (left), there are two boxes shown in between 0 and 17k on the x–axis: the left one corresponds to the box and whiskers for the baseline architecture, while the right one corresponds to the box and whiskers for the C–FRPN, both when the total number of parameters is 17,000. There are a number of observations which can be made from Figure 4.4.

1. In general, the C–FRPN has better performance than the baseline architecture. An exception is observed for the ISIC dataset when the number of parameters is large, e.g., 527k, 672k. Moreover, also for the ISIC dataset, we can observe that the box and whisker plots for the C–FRPN architecture show much more variations (larger boxes, and longer whiskers) than for the corresponding baseline. For the other two datasets, the box and whisker plots corresponding to the C–FRPN, in general, are smaller than for the baseline architecture.

   This could be explained by considering the size of the three datasets. The ISIC dataset has the smallest number of training samples, while SVHN has the largest number of training samples. Therefore, Figure 4.4 (right) may be explained by the fact that the dataset is small, containing only about $5,000$ images. With a greater number of parameters, e.g. $672,000$, the network is showing telltale signs of overfitting. Conversely, when there are a large number of training data, e.g. in the SVHN dataset, for $672,000$ parameters, the C–FRPN shows better performance than the corresponding baseline. This suggests that the C–FRPN is more efficient in making use of data, because RCLs can process them repeatedly, until the state converges, while convolutional layers use the data only once in a forward fashion. Said in other words, a RCL can be unfolded for as many stages as required for a particular data item, until the state converges, while for a convolutional layer, the data item is processed only once. Thus for the four intermediate stages, in the baseline architecture, a data item is only processed four times, while in the C–FRPN, the processing steps are not known *a priori*, but, to reach convergence, they are definitely more than four. This is the reason why the C–FRPN outperforms the baseline when there are a large number of training data, while, if there is an insufficient number of training data, the C–FRPN tends to overfit, and therefore, its performance is similar to that of the baseline.

2. A side–effect of the above mentioned characteristic of C–FRPNs is that the training time is longer than that of the baseline. Figure 4.5 shows the average training time required for an epoch, one complete run through the training dataset, for the baseline and the C–FRPN, respectively, for the three bench-

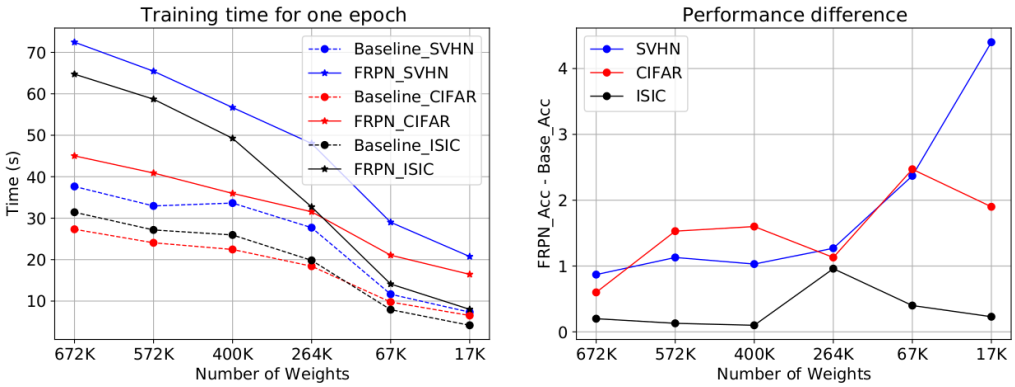


Figure 4.5: Average time required for one epoch of training (left). Performance difference between our C–FRPN and the baseline model for all the network sizes (right).

mark datasets (left), while, on the right side, it is reported the gap between the C–FRPN and the baseline model. The x–axis shows the six discrete values of the number of weights in the network, while the y–axis shows the average time for one epoch through the training data in seconds (left) and the differences in accuracy (right).

The diagram in Figure 4.5 (left) confirms that in all the three datasets, the C–FRPN takes, on an average, a longer time per epoch than its baseline counterpart. The diagram on the right presents the cost benefit ratio between the C–FRPN and the standard CNN. It can be observed that the cost–benefit is best for C–FRPNs of size 264k and larger.

3. It can be observed that the box and whisker plots of the C–FRPN, with respect to the two datasets which have a large number of training samples, are much narrower when the number of parameters is large. The converse of this observation could be made on the ISIC dataset. This can be explained by noting that the C–FRPN probably requires a large number of training data to converge and, therefore, when there are many, the model is well trained, and shows little variations when applied to the test set. On the other hand, for a relatively small number of training data, the C–FRPN does not generalize as well, producing a higher variation on the test set. For the ISIC dataset, since the number of training data is small, the trained model could have many levels of overfitting, and, therefore, the more parameters the greater the chance of overfitting, resulting in a greater variation in generalization performance.

4. For the SVHN and CIFAR-10 datasets, when the number of parameters is large, like 672k or 527k, the box and whisker plots of the baseline architecture tend to be narrower than for the C–FRPN. In some cases, e.g. when the number of

Dataset	C–FRPN	number of iterations fixed to 3
SVHN	96.4 ± 0.1	96.7 ± 0.1
CIFAR-10	89.5 ± 0.7	90.5 ± 0.1
ISIC	81.8 ± 0.6	80.8 ± 2.4

Table 4.2: Comparison between our method and the same model unfolded for three iterations, as in (Liang and Hu, 2015).

Dataset	C–FRPN	Liang et. al Liang and Hu (2015)
SVHN	$96.8 \pm < 0.1$	$96.6 \pm < 0.1$ (98.1)[2]
CIFAR-10	91.6 ± 0.1	92.5 ± 0.2 (92.6)
ISIC	81.8 ± 0.6	82.1 ± 1.7 (NA)

Table 4.3: Comparison between our best results and the best outcomes of the method in (Liang and Hu, 2015), as reproduced in our experiments. The value enclosed in brackets comes from (Liang and Hu, 2015).

parameters is $672k$, the baseline shows almost a unique solution, despite different initial conditions for which the box and whisker plots were obtained, while the C–FRPN shows multiple solutions. This can be due to the loss landscape of the C–FRPN, which is more complex than its baseline counterpart. Actually, there are fewer minima on the baseline loss surface, making learning easier and more "deterministic".

5. In (Liang and Hu, 2015), the number of iterations of the "recurrent" layer is fixed to three, whereas the number of iterations is not fixed in the learning algorithm of the C–FRPN. Table 4.2 compares the results obtained by our model without the constraint on the number of iterations, with the same model iterated for three times. From Table 4.2, it can be observed that, when the number of iterations is fixed at three, this could lead to improvements in results for large datasets, such as CIFAR-10 and SVHN. However, a fixed number of iterations appears to lead to a lower performance in the case of small datasets, such as ISIC. This result demonstrates that the performance of a method can be improved by hand–tuning the number of iterations, while our proposed method is able to adjust such a number by itself.

6. Table 4.3 compares our best model with that of (Liang and Hu, 2015). We can point out that our best accuracy is approximately 1% lower. Indeed, by using the same setup and parameters, we were able to produce results that were very similar to the original ones, with best accuracies of 96.8%, 90.6% and 83.2%, for SVHN, CIFAR-10, and ISIC, respectively. The small differences between the two approaches may be due to their implementation. In fact, while

[2] In this case, 128 neurons were used in (Liang and Hu, 2015) for each layer instead of 96.

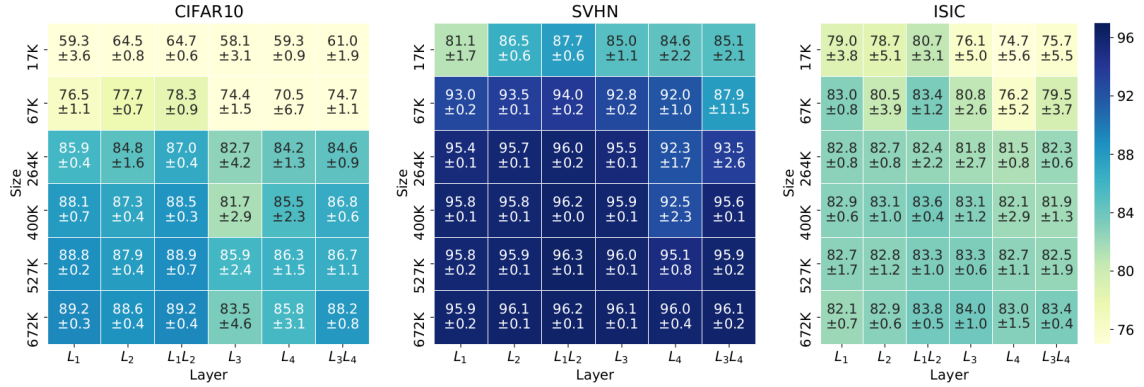


Figure 4.6: Results obtained by using only one or two subsequent RCLs.

we implemented our software in Tensorflow, in (Liang and Hu, 2015), Caffè and PyTorch were used. A second source of discrepancies is that (Liang and Hu, 2015) reported the parameter setting for the CIFAR-10 experiments, not explicitly indicating if the same setup is used also for the SVHN dataset. We have based our experiments on this assumption, though we have no way of ascertaining it. This speculation is based on the observation that, in Table 4.3, we produce very comparable results for CIFAR-10, but worse for SVHN. In Liang and Hu (2015), unfortunately, no experiments were carried out on ISIC, and therefore we are not able to compare our results on this dataset. A final source of discrepancy may be due to the different number of neurons used in each layer (128 vs. 96) for the SVHN dataset. Anyway, the main aim of our work is not to compare our method with state–of–the–art approaches, but rather exploring fundamental questions associated with such an architecture — containing a mixture of convolutional layers and RCLs —, while a comparison has been carried out only for the sake of completeness.

Investigations on how to locate RCLs We have then investigated the effectiveness of using RCLs in different positions inside the network. First, we used the six discrete network sizes as defined above, posing RCLs in one or two consecutive positions, within the four intermediate stages. The results are shown in Figure 4.6 and indicate that there is a relation between the architectural configuration and the number of unknown parameters.

To show all results against six different dimensions for the unknown parameter set would have produced many tables, too many to show here. Therefore, we decided to use a representative network to illustrate the general results of such experiments. Indeed, preliminary experiments, not shown here for the sake of brevity, suggested that the architecture with a total number of 264k unknown parameters appears to produce the most consistent results across all the three datasets. We thus

Figure 4.7: Results for all the possible combination of RCLs (network size fixed to 264K weights).

	L_1	L_2	L_3	L_4	L_1L_2	L_1L_3	L_1L_4	L_2L_3	L_2L_4	L_3L_4	$L_1L_2L_3$	$L_1L_2L_4$	$L_1L_3L_4$	$L_2L_3L_4$	$L_1L_2L_3L_4$
RCL in I.S. 1	✓				✓	✓	✓				✓	✓	✓		✓
RCL in I.S. 2		✓			✓			✓	✓		✓	✓		✓	✓
RCL in I.S. 3			✓			✓		✓		✓	✓		✓	✓	✓
RCL in I. S.4				✓			✓		✓	✓		✓	✓	✓	✓

Table 4.4: The 16 architectures obtained by inserting/permuting RCLs into the four intermediate stages. Labels L_i, $i = 1, \ldots, 4$, correspond to the location of the RCLs. A ✓indicates that a RCL is present in the correspong layer, while a blank indicates a convolutional layer.

selected only networks of size 264*k* for further investigations on all the possible combinations and permutations of convolutional layers and RCLs. In this case, we obtain 16 permutations as shown in Table 4.4. The corresponding results are summarized in Figure 4.7.

Figure 4.7 shows the generalization performance of various combinations and permutations of convolutional layers and RCLs in the intermediate stages of the C–FRPN for each of the three datasets, sorted by accuracy. Each box reports the accuracy and standard deviation achieved by the relative architecture. For ease of interpretation, each square is also coloured according to the accuracy value, with darker colors representing greater accuracy. Thus, the architecture with the lowest generalization accuracy is shown to the left, and that with the highest generalization accuracy is shown to the right. For example, for the first row, the leftmost box indicates that for an RCL located in the intermediate stage 3 — while the other layers are all convolutional — the generalization accuracy is 82.7%, with a standard deviation of $\pm$4.2%, while the rightmost box describes an architecture with RCLs in intermediate stages 1, 2, and 3, which achieves a generalization accuracy of 87.3%, with a standard deviation of $\pm$0.4%.

The following observations can be made on the basis of Figure 4.6 and Figure 4.7:

1. Even a unique RCL in the four intermediate stages allows improved results with respect to a standard CNN, constituted by four convolutional layers.

2. What if we had all four intermediate stages made through RCLs? The results

SVHN	$L_1L_2L_3$	$L_1L_2L_4$	$L_1L_3L_4$	$L_2L_3L_4$
CIFAR-10	$L_1L_2L_3$	$L_1L_2L_4$	$L_1L_3L_4$	$L_2L_3L_4$
ISIC	$L_1L_2L_3$	$L_1L_2L_4$	$L_1L_3L_4$	$L_2L_3L_4$

Table 4.5: All the combinations and permutations of three RCLs. The permutations are ranked from left to right according to the results shown in Figure 4.7, with the one in the rightmost position being that with the highest generalization accuracy.

observed support the following observation: if we have a large amount of training data, the best performance is guaranteed. This is evident from the middle row of Figure 4.7. If we have a small training set, such as in the ISIC dataset, the performance obtained by placing RCLs in all four intermediate stages produces mediocre performance (here it is ranked fourth out of 15). For the SVHN dataset, the architecture with $L_1L_2L_3L_4$ achieves the best rank over the 15 possible combinations and permutations. The number of training data of CIFAR-10 is in between that of SVHN and ISIC and, therefore, the performance of $L_1L_2L_3L_4$ is ranked eighth out of 15.

3. If we want to place a single RCL, what is the best position? It seems that if we have a huge training set, then placing it in the L_1 position is a good choice. This is certainly supported by CIFAR-10, with a rank 7 out of 15. For the SVHN dataset, this solution is ranked 4 out of 15, but the difference between choosing L_2 (ranked 9 out 15) is 0.3% only. For the ISIC dataset, the L_1 architecture is ranked 12 out of 15, with the best performance in case of a single RCL.

4. An inverse question is: where is the most damaging place for a single RCL? For SVHN, L_4 is ranked first out of 15, for CIFAR-10, L_3 is ranked first out of 15 and, for ISIC, L_4 is ranked third out of 15. We can also note that, for SVHN, the difference between L_1 (ranked fourth out of 15) and L_3 (ranked fifth out of 15) is 0.1%. Therefore, this appears to support the observation that placing a unique RCL at the fourth intermediate stage, i.e. closer to the output, appears to be the most damaging, regardless of the size of the training dataset.

5. If we employ three RCLs, what would be the optimal way to distribute them? According to table 4.5, the ranking for all three datasets is identical. This supports the idea that, in general, it is advantageous to place all RCLs in the first three intermediate stages, L_1, L_2, L_3 respectively.

6. What could be the effects of positioning only two RCLs? This is shown in Table 4.6, where the first two rows are virtually identical just swapping L_1L_3 with L_1L_4. The third row shows, instead, a different pattern with respect to the first two rows. This may be explained by the insufficient number of training data in ISIC, which requires to early stop the training — a procedure not required

SVHN	L_1L_2	L_1L_3	L_1L_4	L_2L_3	L_2L_4	L_3L_4
CIFAR-10	L_1L_2	L_1L_4	L_1L_3	L_2L_3	L_2L_4	L_3L_4
ISIC	L_1L_3	L_2L_3	L_2L_4	L_1L_2	L_3L_4	L_1L_4

Table 4.6: All the combinations and permutations of two RCLs. The permutations are ranked from left to right according to the results shown in Figure 4.7, with the one in the rightmost position being that with the highest generalization accuracy.

for CIFAR-10 and SVHN. From the patterns exhibited in the first two rows, it seems that the most beneficial configuration is to have the RCLs located close to the input, in position L_1 and L_2, respectively.

In summary, the results shown in table 4.6 support the following observations: (a) any inclusion of an RCL is advantageous from a performance point of view; (b) with a sufficient amount of data, it is useful to place two/three RCLs in the first intermediate stages; and (c) when sufficient training data are available, using four RCLs produces the best results. Having as much RCLs as possible is quite reasonable, considering that the convolutional layers in a CNN serves for feature extraction. Each convolutional layer extracts more and more abstract features than the previous layers. Therefore, the closer the RCL is placed to the input of a CNN, the better the feature extraction will be, while the further the RCL is far from the input, the less effective it will be at extracting the features, which are more abstract than to those further upstream. The placing of more than one RCL up to populating all intermediate stages with RCLs would be dependent on the training dataset dimension. If the training dataset is large enough, the more RCLs the better are the results.

Conclusions

This research shows new evidence of the benefits of having RCLs in CNNs for image processing applications. Indeed, the proposed C–FRPN model surpasses standard CNNs with the same number of levels and parameters. In general, RCLs should be placed in the lower layers of the network and, depending on the amount of training data available, the more RCLs, the better the generalization performance. This discovery could have a dramatic impact on future network architecture developments, enabling the creation of networks with data–dependent depth and a very high number of parameters, as weights are shared by the unfolding of an RCL.

The work presented here extensively extends both (Hagenbuchner et al., 2017) and (Liang and Hu, 2015), taking the best of both and analyzing the effects obtained in image analysis. Future work may consider applying this type of network to tasks such as image segmentation and object localization. Furthermore, a fascinating research area could be to find a theoretical reason that can formally explain the greater power of C–FRPN, compared to common CNN architectures.

4.2 Analysis of brain NMR images for age estimation with deep learning

In this section, we propose a new approach for age estimation based on 3D NMR brain images. Since training 3D convolutional neural networks is computationally expensive, we studied an alternative solution based on the combination of recurrent neural networks and 2D convolution neural networks. Another simpler method, which employs 2D convolutions, has also been proposed as a baseline for comparison purposes.

Analysis of brain NMR images for age estimation

The brain is the command center of the human nervous system and controls most of the body's activities, also supervising the reception and processing of sensory information. Furthermore, cognitive abilities, language, emotions, creativity and memory are governed by the brain. Unfortunately, like all other parts of the human body, the brain also suffers from aging.

Aging is not uniform between different people and causes changes in brain size, vascularization and cognition. The memory problems and cognitive impairment that may occur during aging would be more related to the loss of white matter, connective of the different regions of the brain, rather than the simple degeneration at the level of the cerebral cortex (grey matter), as revealed by a study of the Massachusetts Institute of Technology (Ziegler et al., 2010). The white matter consists of beams of neuronal axons that make connections between the neurons, allowing the brain regions to communicate with each other. The grey matter, on the other hand, is the place where neurons are found. In older subjects, a correlation between the decline in cognitive performance and the deterioration of the white matter of the frontal cerebellar regions, where the planning and execution functions are located, can be highlighted. Likewise, the deterioration of the white matter in the parietal and temporal lobes was associated with the weakening of memory.

The term *dementia* refers to the loss of cognitive functions, particularly memory, which is so serious as to interfere with everyday life. Alzheimer's disease is the most common form of dementia. In the brains of patients with Alzheimer's disease, the deposition of the amyloid protein and the death of neurons in the cortex is observed. Radiological examinations show, however, also a damage of the white matter, that part of the brain which is instead mainly constituted by myelin. The damage of the white matter seems to be a crucial element in the pathogenesis of Alzheimer's disease and the correlation between the levels of amyloid in the liquor and the lesion extension seems to suggest a direct link between the amyloid pathology and the damage of the cerebral white matter, which produces "premature aging"

(Pietroboni et al., 2018). This observation underlines the importance of the evaluation of the overall state of the brain (white–grey matter ratio) in a disease that has always been considered primarily linked to the degeneration of neurons, and opens the way to new techniques of early prognosis and to the identification of new therapeutic targets.

The premature aging of the brain is therefore an alarm for the onset of neurodegenerative diseases (Schnack et al., 2016; Cole et al., 2017b; Pardoe et al., 2017). Predictive neuroimaging models can be used to learn the brain age in healthy people. In the case of a suspected case of early dementia, a comparative assessment of the brain age can be made in relation to what is estimated for healthy peers. Moreover, if the estimated age based on brain Magnetic Resonance Imaging (MRI) is significantly greater than the actual age of an individual, this may reflect an unusual accumulation of age–related changes in the brain, an effect that can be quantified simply by comparing the actual with the predicted age. This approach has been adopted in several studies that correlate the presence of neurodegenerative pathologies with an increase in the expected brain age (Cole et al., 2015; Franke et al., 2013). Similar approaches have also been used to demonstrate the protective influence of meditation (Luders et al., 2016), physical activity and education (Steffener et al., 2016) on brain aging. Thus, it is easy to realize that the accurate prediction of the brain age can have a great clinical relevance (Cole et al., 2017a).

Because of the three–dimensional structure of the data, to analyze NMR brain images, 2D convolutions are commonly replaced with 3D convolutions. Unfortunately, the use of 3D convolutions introduces a significant increase in the computational load.

To address this problem, we propose to replace 3D with 2D convolutions, thereby substantially reducing memory and computational requirements. A wide range of experiments was conducted, testing two different approaches:

- The 3D image is decomposed based on axial, sagittal and coronal views, which are processed independently by three different CNNs. The encoded representations are then combined by two successive dense layers that produce the age prediction.

- All sections (slices) along the depth of the 3D image are fed into a pre–trained CNN that produces a coded representation of the slices. The CNN output is then concatenated and processed sequentially by a Bidirectional Long Short Term Memory (BLSTM) (Schuster and Paliwal, 1997; Baldi et al., 1999). Finally, the output of the BLSTM is the input for a dense level that generates the age prediction.

Our best–scoring architecture is compared to a state–of–the–art 3D–CNN, showing its ability to achieve competitive results in terms of accuracy, but requiring sig-

nificantly less computational resources. In particular, if the hardware configuration is limited to a single GPU, our model exceeds the 3D–CNN model.

Models

This section introduces all the models considered in this study, namely the 3D convolutional model, used as the baseline, the 3Way–Net, which considers separately the axial, sagittal and coronal views of 3D images, and the slice by slice approach, based on both CNNs and Long–Short–Term–Memories.

3D–CNN In this study, the 3D–CNN model for predicting age from NMR brain images proposed by (Cole et al., 2017a) is used as the baseline. The 3D–CNN is composed of five identical blocks, each consisting of a 3D convolution with kernel size of $3 \times 3 \times 3$ and stride 1, followed by a ReLu non–linearity. A second identical 3D convolutional layer is connected in cascade to the first, followed by a batch normalization operation to which a ReLu activation function is applied. Finally, a $2 \times 2 \times 2$ max–pooling operation is used to reduce the size of the feature maps. In particular, the first block has eight feature maps, a number that is doubled block by block (with the last one composed by 128 feature maps). Lastly, two fully connected layers, with ReLu activation functions, are used to perform age prediction. The entire network architecture is shown in Figure 4.8.

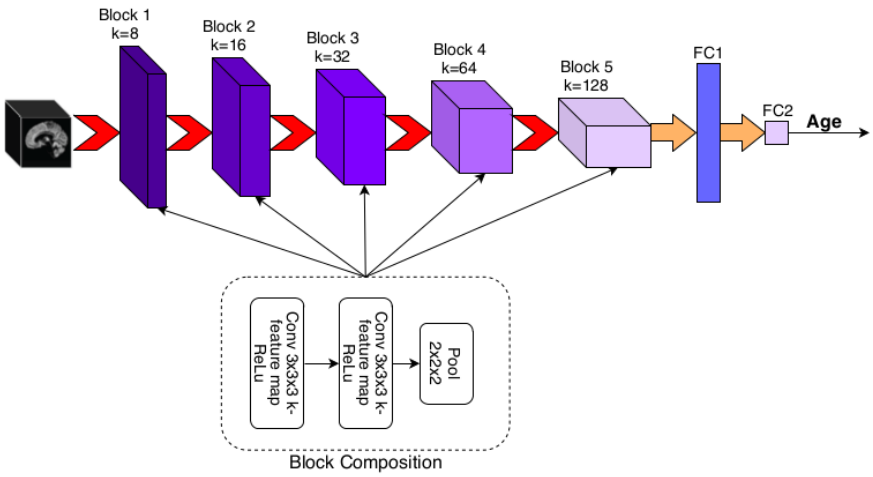


Figure 4.8: The 3D–CNN architecture proposed in (Cole et al., 2017a).

3Way–Net The logic behind the approach described below is inspired by the idea of obtaining a representation of a three–dimensional object through its orthogonal projections on the Cartesian planes. Therefore, from the 3D NMR images, three

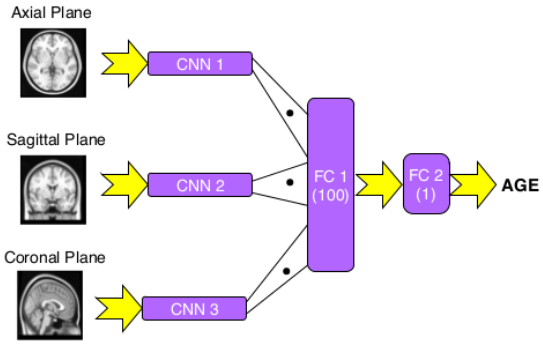


Figure 4.9: The architecture of the 3Way–Net.

views — axial, sagittal and coronal[3] — were extracted and processed independently, using three different 2D convolutional networks. Each view includes a sequence of 2D images with different shapes. The sequence of images is treated as a channel from the network. This means that, for the axial plane, we process images composed of $218 \times 182 \times 182$ pixels, for the sagittal plane of $182 \times 182 \times 218$ pixels, and for the coronal plane of size $182 \times 218 \times 182$. To produce the age prediction, the outputs of the three CNNs are concatenated and fed into two fully connected layers, each of which is followed by a ReLu activation function. The proposed architecture, called 3Way–Net, is represented in Figure 4.9.

To implement the 3Way–Net, two different types of convolutional neural networks were used.

1. *3Way–CNN* — The first architecture is a simple CNN with four identical blocks. Each block is composed of four 3×3 convolutions with stride 1, with batch normalization and ReLu activation functions. The number of feature maps doubles each time (16, 32, 64, 128), while a max–pooling operation, with kernel size 2×2, is used to reduce their size. A distinct CNN model is implemented for each plane extracted from the 3D NMR image.

2. *3Way–ResNetlike* — The second evaluated model is inspired by the ResNet architecture proposed by (He et al., 2016). In the ResNet, a skip connection across convolutional layers is used to combine different information and to better back–propagate the gradient. The proposed model employs three ResNet blocks with four basic units each. These units consists of two 3×3 convolutions with stride 1, having batch normalization and ReLu activation functions. A skip connection is used to sum the output of a unit with the output of the

[3]The axial plane (lateral, horizontal) divides the brain into the upper and lower sides, the last including the cerebellum. The sagittal plane (longitudinal, anteroposterior) is a plane parallel to the sagittal suture. It divides the brain into left and right. The coronal (vertical) plane divides the brain into front and back.

previous unit. To obtain dimensionality reduction in the first convolution of each block, a stride of 2 is employed. Moreover, in the first skip connection, to match the dimensionality between the input and the output of the unit, a 3×3 convolution with stride 2 is also applied. Finally, the last block is followed by a 2×2 pooling operation. A distinct architecture similar to the ResNet is employed for each plane extracted from the NMR image.

Slice–By–Slice analysis with CNNs and Bidirectional LSTMs The main idea behind this method is to consider the slices (or sections) of an NMR image as a sequence. The sagittal plane was used as the reference plane. A pre–trained CNN is employed to extract some features from each slice, constituting a sequence which is processed by a bidirectional LSTM (BLSTM). BLSTMs are a particular type of recursive networks based on the Long–Short–Term–Memory architecture (Hochreiter and Schmidhuber, 1997), which have been proven effective in addressing the problem of long–term dependencies (Graves et al., 2013; Bahdanau et al., 2014). Conventional LSTMs are able to use only the previous information, i.e. they process temporal (or sequential) data following their natural flow. BLSTMs, on the other hand, analyze the data in both directions (considering the context of each element within the sequence), using two separate hidden layers. Three models have been proposed, based on the use of different CNN architectures, pre–trained on the Image–Net dataset (Krizhevsky et al., 2012), to act as feature extractors, namely:

- VGG16 (Simonyan and Zisserman, 2014) for *SbS–R–VGG16*;
- ResNet (He et al., 2016) with 50 layers for *SbS–R–ResNet50*;
- DenseNet (Huang et al., 2017) for *SbS–R–DenseNet121*.

Three consecutive slices were concatenated to obtain 3–channel images to be fed into the pre–trained networks. The output of the CNNs are the features extracted from this group of three slices. All slices are processed in this way and the extracted features are analyzed as a sequence by a BLSTM with 50 internal units with ReLu activation. Finally, the age prediction task is performed by two fully connected layers, each of which followed by a ReLu activation function. The overall SliceBySlice (SbS–R–CNN) architecture is shown in Figure 4.10.

Experimental setup and results

The experiments conducted to validate the proposed approaches are described below. In particular, we first introduce both the dataset and the preprocessing phase, designed to obtain images of the brain without the skullcap and therefore containing only functional information for the problem to be solved (white and grey matter).

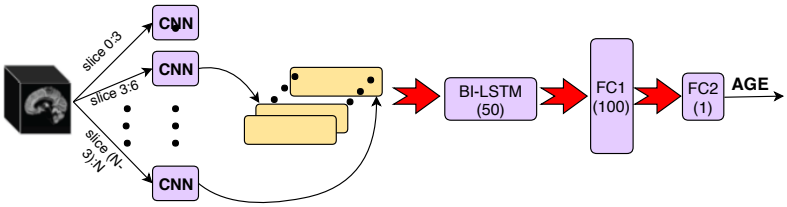


Figure 4.10: The architecture of the SbS–R–CNN.

Subsequently, the experimental setup is described, together with the obtained results, which demonstrate how our methods guarantee performances similar to 3D methods, with a lower training time and a modest memory occupation.

Dataset The IXI dataset (Information eXtraction of Images[4]) collects 600 images of healthy subjects from three major hospitals in London (the Hammersmith Hospital, the Guy's Hospital and the London Institute of Psychiatry), gathered with three different acquisition tools (Philips 3T, Philips 1.5T and GE). The dataset contains different types of NMR sequences (T1, T2, FLAIR, etc.) with the corresponding metadata (gender, age, etc.). In this study, we consider only the T1–weighted images as in (Franke et al., 2010), also reducing the dataset size to 561, due to the absence of the age annotation for the remaining 39 samples. Figure 4.11 reports the distribution by age of the IXI dataset, being the average age of patients 48.65, with a standard deviation of 16.45, and a range of variation from 20 to 86 years. All 3D NMRs are

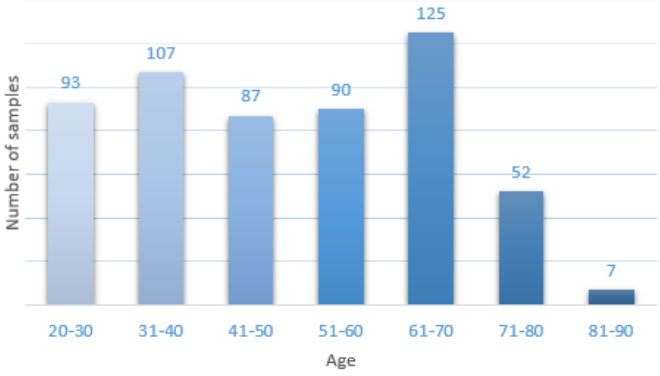


Figure 4.11: Distribution by age of the IXI dataset.

composed of a set of images, each of which represents a 1 mm portion of the brain and has a shape of 256×256 pixels. Depending on the acquisition system, each NMR has a different number of slices (130, 140 or 150).

[4]https://brain-development.org/ixi-dataset/

Pre–Processing The samples were first normalized using a co–registration operation, which is a standard pre–processing step in brain image analysis. This operation consists in maximizing the overlapping voxels between the current image and a reference template model, relying on the Normalized Correlation Coefficient (NCC). The overlapping region is defined as:

$$X_0 = \{x_0 : x_0 \in X \cap T(X')\} \tag{4.4}$$

where X is the template image, X' is the target image, and T is a rigid body transformation. The NCC of $F(X_0)$ and $G(X_0)$, representing the intensity set of the overlapping region w.r.t. the template and the target image, can be defined as:

$$NCC(F,G) = \frac{1}{N_0^2} \frac{\sum_{x_0 \in X_0} (f(x_0) - \bar{f})(g(x_0) - \bar{g})}{\sigma_f \sigma_g} \tag{4.5}$$

where $\bar{f}$, $\bar{g}$, σ_f and σ_g represent the mean and the standard deviation of the voxel intensity in $F(X_0)$ and $G(X_0)$, respectively, and $N_0 = |X_0|$. In this work, the MNI152–T1 Weighted (Evans et al., 2012) template has been used, which is the standard template for T1 NMR. After the pre–processing step, we obtain a set of images of size $182 \times 218 \times 182$.

In order to obtain NMR images without the skull, the Brain Extraction Tool (BET) has been used (Smith, 2002). This allows us to create two datasets, one containing the entire head (brain and skull) and the other containing only the brain (see Figure 4.12 for an example of the two types of images). Both datasets consist of 561 images with the same voxel level (between 0 and 1) and the same size. Each dataset (head and brain) is divided into a training, a validation and a test set, containing 447, 56 and 58 images, respectively (80% training, 10% validation and 10% test).

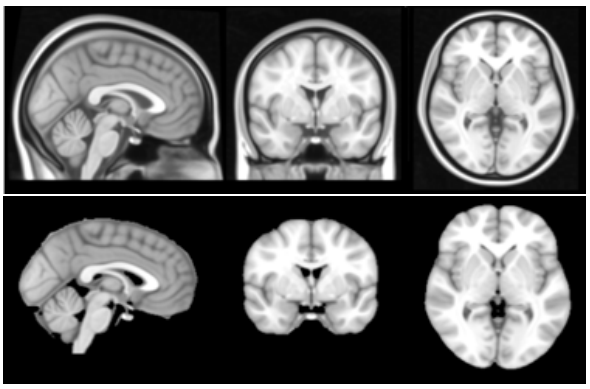


Figure 4.12: NMR images of the whole head (top) and of the brain (bottom); brain images are obtained after the skull–stripping process.

Experimental setup The analysis conducted in this section is aimed at comparing different CNN models, computationally cheaper than the algorithm based on 3D

convolutions proposed in (Cole et al., 2017a), to perform age prediction from NMR brain images. The comparison mainly focuses on performance, training time, and memory load. In fact, our goal is to find the best model that can be trained on a single Nvidia GTX 1080. Unfortunately, in (Cole et al., 2017a), a dataset that is not publicly available was used in the experiments and, for this reason, a fair comparison required to retrain and test the 3D–CNN on the IXI dataset. To satisfy the data memory constraints of the unique available GPU, we were forced to use a batch size of 2 instead of 32 when dealing with 3D convolutions. All models were trained based on the Adam optimizer (Kingma and Ba, 2014), with an initial learning rate of 0.001, using the Root Mean Square Error (RMSE) loss function (see Eq. (4.6)):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y} - y)^2}{N}} \tag{4.6}$$

where y and $\hat{y}$ are the target and the predicted value, respectively, and N is the number of samples. The training is stopped if the loss on the validation set does not decrease for at least 30 epochs or when the training reaches a predefined maximum number of epochs (200). The test evaluation has been performed using the Mean Absolute Error (MAE) metric (see Eq. (4.7)), as in (Cole et al., 2017a):

$$MAE = \frac{\sum_{i=1}^{N}|\hat{y} - y|}{N} \tag{4.7}$$

We repeated each experiment three times for all the compared models, in order to provide reliable results.

Experimental results This section reports the results obtained with our proposed architectures on the IXI dataset, for the age estimation based on brain NMR images. In Table 4.7, our approaches are compared with the state–of–the–art 3D–CNN presented in (Cole et al., 2017a).

When the hardware is tied to a single GPU configuration, our best scoring architecture (SbS–R–CNN) significantly outperforms that presented in (Cole et al., 2017a), with a 0.82 year improvement in terms of mean absolute error over the 3D–CNN. This suggests that when the hardware availability is limited, our method is a viable alternative for processing 3D NMR brain images. Furthermore, the training time is substantially reduced compared to the 3D–CNN. The comparison between the SbS–R–CNNs and the 3Way–Net adopted in this study shows that all SbS–R–CNNs provide better results with reduced training times. The 447 training images available, however, may not be sufficient to train a very deep network, such as 3Way–Net or 3D–CNN, due to the huge number of parameters. The performance of the different models used as feature extractors in the SbS–R–CNN architectures are also compared. It can be seen that VGG–16 provides better results than ResNet–50 and

Model	MAE Brain (Var)	MAE Head (Var)	Time per epoch (m)	Training time (h)
3D–CNN	6.76 (0.32)	6.89 (0.41)	20	70
3Way–CNN	6.99 (0.37)	7.93 (0.64)	4	4.5
3Way–ResNetlike	6.61 (0.27)	8.29 (0.65)	5	7
SbS–R–VGG16	5.94 (0.32)	6.15 (0.32)	1	1.66
SbS–R–ResNet50	6.86 (0.36)	7.15 (0.18)	1	1.58
SbS–R DenseNet121	6.17 (0.41)	6.32 (0.26)	1	1.5

Table 4.7: Results.

DenseNet–121. Finally, it is worth noting that even the best methods for assessing the age, starting from brain images, still produce significant errors (of the order of 6 years). In the specific case of the IXI dataset, this is due to the presence of a very significant percentage of samples relating to over 40 people. If, in fact, the age estimation can be performed with high precision (less than two years) in the case of young people (Franke et al., 2012), the difficulty of estimation increases exponentially with increasing age, since even in healthy individuals the aging patterns can be significantly differentiated, mainly due to lifestyle.

Conclusions

In this research, we have proposed some new approaches for age estimation, based on brain MRI using deep Convolutional Networks. Because of the structure of the data, MRI is normally performed on the basis of 3D convolutions, which implies a considerable memory load and takes a long time. To this end, some *ad hoc* methods have been examined based on 2D convolutions to optimize both memory consumption and the time required for training. The preliminary experimental results are really promising, showing how the SbS–R–CNN can outperform a state–of–the–art 3D–CNN in the case of a hardware configuration limited to a single GPU, opening the possibility for small health institutions to apply powerful methods for the early diagnosis of neurodegenerative diseases without huge investments. Finally, the proposed approaches can be easily generalized to different applications based on 3D images.

Chapter 5

Other works

This chapter is dedicated to briefly explaining other relevant works that I did during the PhD period and in which I was the principal investigator or simply a collaborator. They were born for different reasons and from different collaborations. In particular, the first research, described in Sec. 5.1, is related to the application of a hybrid inductive–transductive learning scheme to graph neural networks. Instead, a collaboration with a team of biologists brought to the realization of two journal papers, summarized in Sec. 5.2, describing the deployment of a new interactive tool for a genetic rare disease called Alkaptonuria. Finally, the last study regards the protein folding prediction and it is illustrated in Sec. 5.3.

5.1 On inductive-transductive learning with graph neural networks

This work is published in (Rossi et al., 2018) and its extended version is submitted to IEEE TPAMI (see the appendix A Paper under review 2). This research is related to learning in Graph Neural Networks (GNNs), which are a connectionist model suited to process graphs.

Graphs are made up of nodes, which denote entities, and arcs, which represent the relationships between them. Both entities and relationships can be endowed with features, which describe their nature. Currently, neural networks capable of processing data expressed in a non–Euclidean space, such as graphs, have become increasingly relevant, thanks to their ability to model relationships that allow us to tackle problems from social networks, cybersecurity, computational biology, etc.

By their nature, GNNs can be trained in both an inductive and a transductive framework. In inductive learning, model parameters are learned from the data in the training set. Then, during the test phase, the model can use the knowledge gathered in its parameters to predict new samples. Instead, with transductive learning,

training and test data are used together, as the decision is made by comparing their features and assigning objects with similar characteristics to the same class.

In this research activity, we studied a mixed inductive–transductive learning framework, applied to the GNN model. In particular, we have defined a subset of transductive nodes in the training set, having the target as an extra feature, which have been excluded from the weight updating process, typical of inductive learning. The goal was to observe how the inductive and transductive part of the model interact and contribute to its performance. We designed and conducted extensive experimentation that highlighted interesting properties of the new learning framework.

5.2 A new integrated and interactive tool applicable to inborn errors of metabolism: Application to alkaptonuria.

This research is published in (Spiga et al., 2018) and (Rossi et al., 2020b). Precision medicine (PM) is a groundbreaking approach to disease prevention, diagnosis and treatment, based on people's individual differences in genes, metabolomics, proteomics, environment and lifestyle. This does not necessarily mean to tailor a medical treatment to a unique patient, but rather to gain the ability to classify patients into subpopulations, according to their susceptibility to a particular disease or to their response to a specific treatment. Not all patients respond favorably to drugs or benefit from their use. An improved understanding of disease mechanisms, also through the identification of relevant biomarkers, is likely to lead to a more personalized medicine, allowing to match therapies to specific patients and thus maximizing the benefit–to–risk ratio. In the process of biomarker identification, access to biological and clinical data is a critical step, requiring careful processing, storage and organization of such data in an anonymous fashion. This is the core for the development of a "Precision Medicine Ecosystem" where resources are shared among researchers, clinicians and patients.

Alkaptonuria (AKU), the first genetic disorder described by Garrod in 1902, is a prototypical, rare inborn error of tyrosine and phenylalanine metabolism (MIM 203500) resulting from homogenized 1,2–dioxygenase (HGD) deficiency. AKU leads to the accumulation of homogentisic acid (HGA) and to a severe and crippling form of early–onset arthritis, presenting typical blue–black (ochronotic) deposits in connective tissues of joints and spine. Cardiac valves and other organs may be affected too. Moreover, AKU patients suffer from kidney stones. Consequently, in order to prevent or minimize the impact of the disease, a PM approach, achieved by adapting pharmacological, surgical and dietary treatment, could have a very significant

impact for the care of AKU patients. To overcome the limitations related to the lack of approved biomarkers to monitor the progression and severity of AKU, we have recently established a comprehensive database that offers a complete view of the different information layers and is likely to support doctors and researchers in a PM approach to AKU. In addition, we also collected a series of knee cartilage images from AKU and control people. These data reveal some interesting biomarkers for this disease. The proposed framework evaluates some statistics online, instantly exploiting the data added to the database. Finally, an automated image classification tool for biopsied knee cartilage is integrated into the web tool.

5.3 A deep attention network for predicting amino acid signals in the formation of α-helices.

This study produced the paper (Visibelli et al., 2020). The secondary and tertiary structure of a protein plays a primary role in determining its function. Many predictive models have been developed in recent decades, based on the primary structure of a protein, to predict its folding, sometimes reporting very high performance. The rationale behind our study is the search for some specific signals in the amino acid sequence that are able to detect the presence of $alpha$–helix motifs. To answer this question, we conducted an extensive statistical analysis which demonstrates the presence of special amino acid patterns that suggest helix formation. In addition, we have implemented several machine learning methods, equipped with attention modules, to confirm that even artificial models are able to exploit these particular patterns to produce relevant results. Indeed, the experiments showed that different models focus on the same subsequences, which can be viewed as biological signals that drive the formation of specific secondary structure motifs.

Chapter 6

Conclusions and future perspectives

In this thesis the analysis of medical images was approached from an unusual point of view. In the beginning, we studied learning by similarity. Indeed, similarity learning implies a very simple cognitive scheme, widely used also by humans and animals for its immediacy. In fact, it is one of the first learning processes acquired by children. This simplicity encourages the proposed line of research and finds an ideal candidate architecture in the siamese neural network, which can naturally compare patterns. All the methods proposed in Chapter 3 are dedicated to prostate magnetic resonance, which is important in medicine for the wide spread of prostate cancer and for the incredible help that can come from an automatic imaging system to diagnostics.

Initially, a CBIR system was developed, capable of recovering diagnostically similar MRIs of the prostate. The fact that our model is able to incorporate similarities in terms of clinical relevance of cases (based on the PIRADS score) is something other CBIRs cannot accomplish, as they usually limit their comparison to just the visual aspect. Furthermore, our model is able to learn and retrieve multi–parametric images, which are necessary for a very precise diagnosis. The benefits provided to radiologists are extensive given the ability to compare cases even with past diagnoses and prognoses. This fact is fundamental, considering the current trend towards precision medicine, in which patients are also treated based on the evidence of similar cases. The results obtained confirm that our multi–parametric siamese model is better at learning diagnostic similarity than a shallow autoencoder with the same complexity. In particular, the baseline model only returns visually similar patterns, even if the diagnosis is different, while the siamese model did not fall into this trap.

Similarity learning was also used to increase the robustness of a prostate MRI lesion classifier. The proposed architecture was built from a siamese network with two outputs: one of which is evaluated only for the query image and produces the output class, while the other calculates the similarity between the query and the

reference image. At the test time, only the query stream and class output are used. Experiments show how the combination of these two outputs and the use of a composite loss during training improve the robustness of the classifier to both Gaussian noise and adversarial attacks. The effect could be impressive as misdiagnosis could lead to a patient dying, so a robust classifier is mandatory. Additionally, radiologists' feel of a robust model can encourage them to trust and use the system. Finally, since Gaussian noise can be considered as a proxy of possible acquisition perturbations, a system with the advanced properties described can avoid the repetition of non–perfect examinations.

Finally, a method for intra–procedural registration of prostatic MRI is presented. The first step in this approach is to randomly augment the intra–procedural image to obtain a number of possible candidates for the registered image. Then, through a predefined similarity metric, a siamese network is trained to find the best candidate according to the pre–procedural image, which is fed into the second stream of the siamese. This model works better than a common CNN, at least on very small datasets. Furthermore, it is shown that the best performance is achieved by using the mutual information metrics, which makes the model completely unsupervised, as it does not require the support of any segmentation or landmark.

The second part of the thesis is dedicated to the application of recurrent and recursive neural networks to the analysis of medical images. First, we describe how to embed recursive connections in a convolutional layer, to obtain a so–called recursive convolutional layer (RCL). A thorough investigation was conducted to compare standard CNNs with deep architectures having the same complexity but using RCLs, varying the dataset properties, such as size and complexity. Furthermore, the position, number and combination of these recursive layers were analyzed looking for the best option. We applied the new network on several benchmarks, including a dataset for the classification of skin lesions, achieving better performance than the corresponding CNN. Furthermore, the gap is magnified for very small networks. From this we can deduce that the model is advantageous with few computational resources and when few data are available. A fascinating future task could be to find a theoretical explanation to justify these experimental results.

The last research line presented in this thesis is related to age estimation based on brain magnetic resonance. Indeed, predicting age from an MRI brain scan can potentially reveal some unexpected patterns induced by the onset of a neurodegenerative disease, such as Alzheimer's disease. Brain MRI analysis using 3D convolutional neural networks could be computationally challenging. For this reason, two alternatives are proposed and the best is a pre–trained 2D neural network used as a feature extractor for each of the sections that make up the 3D image. The extracted features are then recombined to form a sequence, which is finally processed by a BI–LSTM. Together with the performance, the power of this model also derives from

the possibility of working on cheap machines, limiting the cost of the necessary dedicated hardware, a crucial aspect for small medical centers.

Many of the models proposed in this thesis can be easily generalized to other tasks in medical imaging. Obviously, extensive testing by doctors is needed to understand the real benefits and report any limitations. Furthermore, at least for the case of C–FRPNs, a theoretical investigation could improve the understanding of the model, justifying some of the empirical observations. For the case of learning by similarity, on the other hand, future research could concern the image segmentation phase. A siamese network can in fact choose the most similar image available in the dataset, for which segmentation is present. This segmentation can be concatenated into one of the last layers of the network, making the task of segmenting the new image for the responsible network easier. In addition, other metric learning algorithms should be studied, not exclusively related to the use of siamese nets. Finally, bundling some of the proposed research into a single CAD system could be something really fascinating to close the circle.

Appendix A

Publications

Journal papers

1. **Alberto Rossi**, M. Hosseinzadeh, M. Bianchini, F. Scarselli, H. Huisman, "Multi-modal siamese network for diagnostically similar lesion retrieval in prostate", *IEEE Transaction on Medical Image Analysis*, Early Access. **Candidate's contributions**: designed algorithms, experimental setup, run experiments.

2. Anna Visibelli, P. Bongini, **A. Rossi**, N. Niccolai, M. Bianchini, "A deep attention network for predicting amino acid signals in the formation of α-helices." *Journal of Bioinformatics and Computational Biology*, 2050028-2050028, 2020. **Candidate's contributions**: designed algorithms.

3. **Alberto Rossi**, G. Giacomini, V. Cicaloni, S. Galderisi, MS Milella, A. Bernini, L. Millucci, O. Spiga, M. Bianchini, A. Santucci, "AKUImg: A database of cartilage images of Alkaptonuria patients", *Computers in biology and medicine*, vol. 122, pages:1–7, 2020. **Candidate's contributions**: designed algorithms, run experiments.

4. **Alberto Rossi**, G. Barlacchi, M. Bianchini, B. Lepri, "Modelling Taxi Drivers' Behaviour for the Next Destination Prediction", *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, issue 7, 2020. **Candidate's contributions**: designed algorithms, carried out theoretical analyses, experimental setup, run experiments.

5. Ottavia Spiga, V. Cicaloni, A. Zatkova, L. Millucci, G. Bernardini, A. Bernini, B. Marzocchi, M. Bianchini, A. Zugarini, **A. Rossi**, M. Zazzeri, A. Trezza, B. Frediani, L. Ranganath, D. Braconi, A. Santucci, "A new integrated and interactive tool applicable to inborn errors of metabolism: Application to alkaptonuria", *Computers in biology and medicine*, vol. 103, pages:1–7, 2018. **Candidate's contributions**: designed algorithms, run experiments.

Peer reviewed conference papers

1. **Alberto Rossi**, M. Bianchini, F. Scarselli, "Robust prostate cancer classification with siamese neural networks", *15th International Symposium on Visual Computing.*, 2020. **Candidate's contributions**: designed algorithms, experimental setup, run experiments.

2. Alexander Lyons, **A. Rossi**, "Prostate mri registration using siamese metric learning", *15th International Symposium on Visual Computing.*, 2020. **Candidate's contributions**: designed algorithms.

3. **Alberto Rossi**, M. Hagenbuchner, F. Scarselli, A. C. Tsoi, "Embedding of FRPN in CNN architecture", *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning.*, 2020. **Candidate's contributions**: designed algorithms, experimental setup, run experiments.

4. Niccolò Pancino, **A. Rossi**, G. Ciano, G. Giacomini, S. Bonechi, P. Andreini, F. Scarselli, M. Bianchini, P. Bongini, "Graph Neural Networks for the Prediction of Protein–Protein Interfaces" *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning.*, 2020. **Candidate's contributions**: designed algorithms.

5. M. Monaci, N. Pancino, P. Andreini, S. Bonechi, P. Bongini, **A. Rossi**, G. Ciano, G. Giacomini, F. Scarselli, M. Bianchini, "Deep Learning Techniques for Dragonfly Action Recognition", *Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods*, pages:562-569, 2020. **Candidate's contributions**: designed algorithms.

6. Simone Bonechi, M. Bianchini, P. Bongini, G. Ciano, G. Giacomini, R. Rosai, L. Tognetti, **Alberto Rossi**, L. Tognetti, ''Fusion of Visual and Anamnestic Data for the Classification of Skin Lesions with Deep Learning." *International Conference on Image Analysis and Processing*. 2019. **Candidate's contributions**: designed algorithms.

7. **Alberto Rossi**, G. Vannuccini, P. Andreini, S. Bonechi, G. Giacomini, F. Scarselli, M. Bianchini, "Analysis of brain NMR images for age estimation with deep learning", *23rd International Conference on Knowledge-Based and Intelligent Information and Engineering Systems* (*KES 2019*), *Procedia computer science*, vol. 159, pages:981–989, 2019. **Candidate's contributions**: designed algorithms, experimental setup.

8. **Alberto Rossi**, M. Tiezzi, G. M. Dimitri, M. Bianchini, M. Maggini, F. Scarselli, "Inductive–transductive learning with graph neural networks." *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*, pages:201–212), 2018.

Candidate's contributions: designed algorithms, carried out theoretical analyses, experimental setup, run experiments.

9. Gianni Barlacchi, **A. Rossi**, B. Lepri, A. Moschitti, "Structural semantic models for automatic analysis of urban areas", *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (*ECML 2017*), pages:279–291, 2017. **Candidate's contributions**: experimental setup.

Papers under review

1. **Alberto Rossi**, M. Hagenbuchner, F. Scarselli, A. C. Tsoi, "A Study on the Effects of Recursive Convolutional Layers in Convolutional Neural Networks", *Neurocomputing, Elsevier*, Manuscript sent for review. **Candidate's contributions**: carried out theoretical analyses, experimental setup, run experiments.

2. Giorgio Ciano, **A. Rossi**, M. Bianchini, F. Scarselli, "On Inductive-Transductive Learning with Graph Neural Networks", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, minor revisions pending. **Candidate's contributions**: carried out theoretical analyses, experimental setup, run experiments.

Bibliography

Alfieri, L., Nokes-Malach, T. J., and Schunn, C. D. (2013). Learning through case comparisons: A meta-analytic review. *Educational Psychologist*, 48(2):87–113.

Anavi, Y., Kogan, I., Gelbart, E., Geva, O., and Greenspan, H. (2016). Visualizing and enhancing a deep learning framework using patients age and gender for chest x-ray image retrieval. In *Medical Imaging 2016: Computer-Aided Diagnosis*, volume 9785, page 978510. International Society for Optics and Photonics.

Antony, J., McGuinness, K., O'Connor, N. E., and Moran, K. (2016). Quantifying radiographic knee osteoarthritis severity using deep convolutional neural networks. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 1195–1200. IEEE.

Appalaraju, S. and Chaoji, V. (2017). Image similarity using deep cnn and curriculum learning. *arXiv preprint arXiv:1709.08761*.

Back, A. D. and Tsoi, A. C. (1991). Fir and iir synapses, a new neural network architecture for time series modelling. *Neural Computation*, 3(3):375–385.

Baddar, W. J., Kim, D. H., and Ro, Y. M. (2017). Learning features robust to image variations with siamese networks for facial expression recognition. In *International Conference on Multimedia Modeling*, pages 189–200. Springer.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Baldi, P., Brunak, S., Frasconi, P., Soda, G., and Pollastri, G. (1999). Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics*, 15(11):937–946.

Barentsz, J. O., Weinreb, J. C., Verma, S., Thoeny, H. C., Tempany, C. M., Shtern, F., Padhani, A. R., Margolis, D., Macura, K. J., Haider, M. A., et al. (2016). Synopsis of the pi-rads v2 guidelines for multiparametric prostate magnetic resonance imaging and recommendations for use. *European urology*, 69(1):41.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

Bianucci, A. M., Micheli, A., Sperduti, A., and Starita, A. (2000). Application of cascade correlation networks for structures to chemistry. *Applied Intelligence*, 12(1-2):117–147.

Böhm, C., Berchtold, S., and Keim, D. A. (2001). Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Computing Surveys (CSUR)*, 33(3):322–373.

Bonechi, S., Bianchini, M., Bongini, P., Ciano, G., Giacomini, G., Rosai, R., Tognetti, L., Rossi, A., and Andreini, P. (2019). Fusion of visual and anamnestic data for the classification of skin lesions with deep learning. In *International Conference on Image Analysis and Processing*, pages 211–219. Springer.

Bray, F., Ferlay, J., Soerjomataram, I., Siegel, R. L., Torre, L. A., and Jemal, A. (2018). Global cancer statistics 2018: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: a cancer journal for clinicians*, 68(6):394–424.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. (1994). Signature verification using a" siamese" time delay neural network. In *Advances in neural information processing systems*, pages 737–744.

Chandra, S. S., Dowling, J. A., Shen, K.-K., Raniga, P., Pluim, J. P., Greer, P. B., Salvado, O., and Fripp, J. (2012). Patient specific prostate segmentation in 3-d magnetic resonance images. *IEEE transactions on medical imaging*, 31(10):1955–1964.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chopra, S., Hadsell, R., LeCun, Y., et al. (2005). Learning a similarity metric discriminatively, with application to face verification. In *CVPR (1)*, pages 539–546.

Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., and Bengio, Y. (2015). Attention-based models for speech recognition. *Advances in neural information processing systems*, 28:577–585.

Chung, Y.-A. and Weng, W.-H. (2017). Learning deep representations of medical images using siamese cnns with application to content-based image retrieval. *arXiv preprint arXiv:1711.08490*.

Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., and Ronneberger, O. (2016). 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer.

Clevert, D.-A., Unterthiner, T., and Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

Codella, N. C., Gutman, D., Celebi, M. E., Helba, B., Marchetti, M. A., Dusza, S. W., Kalloo, A., Liopyris, K., Mishra, N., Kittler, H., et al. (2018). Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 168–172. IEEE.

Cole, J. H., Leech, R., Sharp, D. J., and Initiative, A. D. N. (2015). Prediction of brain age suggests accelerated atrophy after traumatic brain injury. *Annals of neurology*, 77(4):571–581.

Cole, J. H., Poudel, R. P., Tsagkrasoulis, D., Caan, M. W., Steves, C., Spector, T. D., and Montana, G. (2017a). Predicting brain age with deep learning from raw imaging data results in a reliable and heritable biomarker. *NeuroImage*, 163:115–124.

Cole, J. H., Underwood, J., Caan, M. W., De Francesco, D., van Zoest, R. A., Leech, R., Wit, F. W., Portegies, P., Geurtsen, G. J., Schmand, B. A., et al. (2017b). Increased brain-predicted aging in treated hiv disease. *Neurology*, 88(14):1349–1357.

Dayan, P., Abbott, L. F., et al. (2003). Theoretical neuroscience: computational and mathematical modeling of neural systems. *Journal of Cognitive Neuroscience*, 15(1):154–155.

de Vos, B. D., Wolterink, J. M., de Jong, P. A., Viergever, M. A., and Išgum, I. (2016). 2d image classification for 3d anatomy localization: employing deep convolutional neural networks. In *Medical imaging 2016: Image processing*, volume 9784, page 97841Y. International Society for Optics and Photonics.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Douglas, R. J. and Martin, K. A. (2007). Recurrent neuronal circuits in the neocortex. *Current biology*, 17(13):R496–R500.

Dumoulin, V. and Visin, F. (2016). A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*.

Elman, J. (1990). Finding structure in time. *Cognitive Science*, 14:179–211.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115.

Evans, A. C., Janke, A. L., Collins, D. L., and Baillet, S. (2012). Brain templates and atlases. *Neuroimage*, 62(2):911–922.

Farabet, C., Couprie, C., Najman, L., and LeCun, Y. (2012). Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1915–1929.

Fedorov, A., Tuncali, K., Fennessy, F. M., Tokuda, J., Hata, N., Wells, W. M., Kikinis, R., and Tempany, C. M. (2012). Image registration for targeted mri-guided transperineal prostate biopsy. *Journal of Magnetic Resonance Imaging*, 36(4):987–992.

Franke, K., Gaser, C., Manor, B., and Novak, V. (2013). Advanced brainage in older adults with type 2 diabetes mellitus. *Frontiers in aging neuroscience*, 5:90.

Franke, K., Luders, E., May, A., Wilke, M., and Gaser, C. (2012). Brain maturation: predicting individual brainage in children and adolescents using structural mri. *Neuroimage*, 63(3):1305–1312.

Franke, K., Ziegler, G., Klöppel, S., Gaser, C., Initiative, A. D. N., et al. (2010). Estimating the age of healthy subjects from t1-weighted mri scans using kernel methods: exploring the influence of various parameters. *Neuroimage*, 50(3):883–892.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202.

Geng, C. and Song, J. (2016). Human action recognition based on convolutional neural networks with a convolutional auto-encoder. In *2015 5th International Conference on Computer Sciences and Automation Engineering (ICCSAE 2015)*. Atlantis Press.

Gleason, D. F. (1992). Histologic grading of prostate cancer: a perspective. *Human pathology*, 23(3):273–279.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE.

Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J., et al. (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Jama*, 316(22):2402–2410.

Guo, H., Wang, J., and Lu, H. (2015). Learning deep compact descriptor with bagging auto-encoders for object retrieval. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 3175–3179. IEEE.

Hadsell, R., Chopra, S., and LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

Hagenbuchner, M., Tsoi, A. C., Scarselli, F., and Zhang, S. J. (2017). A fully recursive perceptron network architecture. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE.

Haugeland, J. (1989). *Artificial intelligence: The very idea*. MIT press.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

Kamnitsas, K., Ledig, C., Newcombe, V. F., Simpson, J. P., Kane, A. D., Menon, D. K., Rueckert, D., and Glocker, B. (2017). Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical image analysis*, 36:61–78.

Khanna, S. and Crues, J. V. (2009). Complexities of mri and false positive findings. *Annals of the New York Academy of Sciences*, 1154(1):239–258.

Kim, E., Corte-Real, M., and Baloch, Z. (2016). A deep semantic mobile application for thyroid cytopathology. In *Medical Imaging 2016: PACS and Imaging Informatics: Next Generation and Innovations*, volume 9789, page 97890A. International Society for Optics and Photonics.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. (2017). Self-normalizing neural networks. In *Advances in neural information processing systems*, pages 971–980.

Klein, A. and Tourville, J. (2012). 101 labeled brain images and a consistent human cortical labeling protocol. *Frontiers in neuroscience*, 6:171.

Knyaz, V. A., Vygolov, O., Kniaz, V. V., Vizilter, Y., Gorbatsevich, V., Luhmann, T., and Conen, N. (2017). Deep learning of convolutional auto-encoder for image matching and 3d object reconstruction in the infrared range. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2155–2164.

Koch, G., Zemel, R., and Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2.

Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images. Technical report, Citeseer.

Krizhevsky, A. and Hinton, G. E. (2011). Using very deep autoencoders for content-based image retrieval. In *ESANN*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Kuang, D. and Schmah, T. (2019). Faim–a convnet method for unsupervised 3d medical image registration. In *International Workshop on Machine Learning in Medical Imaging*, pages 646–654. Springer.

Kulis, B. et al. (2012). Metric learning: A survey. *Foundations and trends in machine learning*, 5(4):287–364.

Kumar, A., Kim, J., Cai, W., Fulham, M., and Feng, D. (2013). Content-based medical image retrieval: a survey of applications to multidimensional and multimodality data. *Journal of digital imaging*, 26(6):1025–1039.

Le, M. H., Chen, J., Wang, L., Wang, Z., Liu, W., Cheng, K.-T. T., and Yang, X. (2017). Automated diagnosis of prostate cancer in multi-parametric mri based on multimodal convolutional neural networks. *Physics in Medicine & Biology*, 62(16):6497.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Liang, M. and Hu, X. (2015). Recurrent convolutional neural network for object recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3367–3375.

Litjens, G., Debats, O., Barentsz, J., Karssemeijer, N., and Huisman, H. (2014). Computer-aided detection of prostate cancer in mri. *IEEE transactions on medical imaging*, 33(5):1083–1092.

Liu, S., Zheng, H., Feng, Y., and Li, W. (2017). Prostate cancer diagnosis using deep learning with 3d multiparametric mri. In *Medical Imaging 2017: Computer-Aided Diagnosis*, volume 10134, page 1013428. International Society for Optics and Photonics.

Liu, X., Tizhoosh, H. R., and Kofman, J. (2016). Generating binary tags for fast medical image retrieval based on convolutional nets and radon transform. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 2872–2878. IEEE.

Lo, S.-C., Lou, S.-L., Lin, J.-S., Freedman, M. T., Chien, M. V., and Mun, S. K. (1995). Artificial convolution neural network techniques and applications for lung nodule detection. *IEEE transactions on medical imaging*, 14(4):711–718.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

Luders, E., Cherbuin, N., and Gaser, C. (2016). Estimating brain age using high-resolution pattern recognition: Younger brains in long-term meditation practitioners. *Neuroimage*, 134:508–513.

Lyons, A. and Rossi, A. (2020). Prostate mri registration using siamese metric learning. In *International Symposium on Visual Computing*, pages 593–603. Springer.

Markman, A. B. and Gentner, D. (1993). Structural alignment during similarity comparisons. *Cognitive psychology*, 25(4):431–467.

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

Milletari, F., Navab, N., and Ahmadi, S.-A. (2016). V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. IEEE.

Mitra, J., Ghose, S., Sidibé, D., Marti, R., Oliver, A., Llado, X., Vilanova, J. C., Comet, J., and Mériaudeau, F. (2012). Joint probability of shape and image similarities to retrieve 2d trus-mr slice correspondence for prostate biopsy. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5416–5419. IEEE.

Moeskops, P., Viergever, M. A., Mendrik, A. M., De Vries, L. S., Benders, M. J., and Išgum, I. (2016). Automatic segmentation of mr brain images with a convolutional neural network. *IEEE transactions on medical imaging*, 35(5):1252–1261.

Muller, B. G., Shih, J. H., Sankineni, S., Marko, J., Rais-Bahrami, S., George, A. K., de la Rosette, J. J., Merino, M. J., Wood, B. J., Pinto, P., et al. (2015). Prostate cancer: interobserver agreement and accuracy with the revised prostate imaging reporting and data system at multiparametric mr imaging. *Radiology*, 277(3):741–750.

Müller, H., Michoux, N., Bandon, D., and Geissbuhler, A. (2004). A review of content-based image retrieval systems in medical applications—clinical benefits and future directions. *International journal of medical informatics*, 73(1):1–23.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning.

Pardoe, H. R., Cole, J. H., Blackmon, K., Thesen, T., Kuzniecky, R., Investigators, H. E. P., et al. (2017). Structural brain changes in medically refractory focal epilepsy resemble premature brain aging. *Epilepsy research*, 133:28–32.

Payer, C., Štern, D., Bischof, H., and Urschler, M. (2016). Regressing heatmaps for multiple landmark localization using cnns. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 230–238. Springer.

Pietroboni, A. M., Scarioni, M., Carandini, T., Basilico, P., Cadioli, M., Giulietti, G., Arighi, A., Caprioli, M., Serra, L., Sina, C., et al. (2018). Csf β-amyloid and white matter damage: a new perspective on alzheimer's disease. *Journal of Neurology, Neurosurgery & Psychiatry*, 89(4):352–357.

Pineda, F. J. (1987). Generalization of back-propagation to recurrent neural networks. *Physical Review Letters*, 59:2229–2232.

Prasoon, A., Petersen, K., Igel, C., Lauze, F., Dam, E., and Nielsen, M. (2013). Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network. In *International conference on medical image computing and computer-assisted intervention*, pages 246–253. Springer.

Rittle-Johnson, B. and Star, J. R. (2011). The power of comparison in learning and instruction: Learning outcomes supported by different types of comparisons. In *Psychology of learning and motivation*, volume 55, pages 199–225. Elsevier.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Rosenkrantz, A. B., Ginocchio, L. A., Cornfeld, D., Froemming, A. T., Gupta, R. T., Turkbey, B., Westphalen, A. C., Babb, J. S., and Margolis, D. J. (2016). Interobserver reproducibility of the pi-rads version 2 lexicon: a multicenter study of six experienced prostate radiologists. *Radiology*, 280(3):793–804.

Rossi, A., Bianchini, M., and Scarselli, F. (2020a). Robust prostate cancer classification with siamese neural networks. In *International Symposium on Visual Computing*, pages 180–189. Springer.

Rossi, A., Giacomini, G., Cicaloni, V., Galderisi, S., Milella, M. S., Bernini, A., Millucci, L., Spiga, O., Bianchini, M., and Santucci, A. (2020b). Akuimg: A database of cartilage images of alkaptonuria patients. *Computers in Biology and Medicine*, 122:103863.

Rossi, A., Hosseinzadeh, M., Bianchini, M., Scarselli, F., and Huisman, H. (2020c). Multi-modal siamese network for diagnostically similar lesion retrieval in prostate mri. *IEEE Transactions on Medical Imaging*.

Rossi, A., Tiezzi, M., Dimitri, G. M., Bianchini, M., Maggini, M., and Scarselli, F. (2018). Inductive–transductive learning with graph neural networks. In *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*, pages 201–212. Springer.

Rossi, A., Vannuccini, G., Andreini, P., Bonechi, S., Giacomini, G., Scarselli, F., and Bianchini, M. (2019). Analysis of brain nmr images for age estimation with deep learning. *Procedia Computer Science*, 159:981–989.

Roth, H. R., Lu, L., Liu, J., Yao, J., Seff, A., Cherry, K., Kim, L., and Summers, R. M. (2015). Improving computer-aided detection using convolutional neural networks and random view aggregation. *IEEE transactions on medical imaging*, 35(5):1170–1181.

Rouvière, O., Puech, P., Renard-Penna, R., Claudon, M., Roy, C., Mège-Lechevallier, F., Decaussin-Petrucci, M., Dubreuil-Chambardel, M., Magaud, L., Remontet, L., et al. (2019). Use of prostate systematic and targeted biopsy on the basis of multiparametric mri in biopsy-naive patients (mri-first): a prospective, multicentre, paired diagnostic study. *The Lancet Oncology*, 20(1):100–109.

Rush, A. M., Chopra, S., and Weston, J. (2015). A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2008). The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

Schelb, P., Kohl, S., Radtke, J. P., Wiesenfarth, M., Kickingereder, P., Bickelhaupt, S., Kuder, T. A., Stenzinger, A., Hohenfellner, M., Schlemmer, H.-P., et al. (2019). Classification of cancer at prostate mri: Deep learning versus clinical pi-rads assessment. *Radiology*, page 190938.

Schnack, H. G., Van Haren, N. E., Nieuwenhuis, M., Hulshoff Pol, H. E., Cahn, W., and Kahn, R. S. (2016). Accelerated brain aging in schizophrenia: a longitudinal pattern recognition study. *American Journal of Psychiatry*, 173(6):607–616.

Schröder, F. H., Hugosson, J., Roobol, M. J., Tammela, T. L., Ciatto, S., Nelen, V., Kwiatkowski, M., Lujan, M., Lilja, H., Zappa, M., et al. (2009). Screening and prostate-cancer mortality in a randomized european study. *New England journal of medicine*, 360(13):1320–1328.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.

Setio, A. A. A., Ciompi, F., Litjens, G., Gerke, P., Jacobs, C., Van Riel, S. J., Wille, M. M. W., Naqibullah, M., Sánchez, C. I., and van Ginneken, B. (2016). Pulmonary nodule detection in ct images: false positive reduction using multi-view convolutional networks. *IEEE transactions on medical imaging*, 35(5):1160–1169.

Shah, A., Conjeti, S., Navab, N., and Katouzian, A. (2016). Deeply learnt hashing forests for content based image retrieval in prostate mr images. In *Medical Imaging 2016: Image Processing*, volume 9784, page 978414. International Society for Optics and Photonics.

Sharma, S., Umar, I., Ospina, L., Wong, D., and Tizhoosh, H. R. (2016). Stacked autoencoders for medical image search. In *International Symposium on Visual Computing*, pages 45–54. Springer.

Siegel, R. L., Miller, K. D., and Jemal, A. (2019). Cancer statistics, 2019. *CA: a cancer journal for clinicians*, 69(1):7–34.

Simard, P. Y., Steinkraus, D., Platt, J. C., et al. (2003). Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*, volume 3.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Smith, C. P., Harmon, S. A., Barrett, T., Bittencourt, L. K., Law, Y. M., Shebel, H., An, J. Y., Czarniecki, M., Mehralivand, S., Coskun, M., et al. (2019). Intra-and interreader reproducibility of pi-radsv2: A multireader study. *Journal of Magnetic Resonance Imaging*, 49(6):1694–1703.

Smith, S. M. (2002). Fast robust automated brain extraction. *Human brain mapping*, 17(3):143–155.

Song, J., Zhang, H., Li, X., Gao, L., Wang, M., and Hong, R. (2018a). Self-supervised video hashing with hierarchical binary auto-encoder. *IEEE Transactions on Image Processing*, 27(7):3210–3221.

Song, Y., Zhang, L., Chen, S., Ni, D., Lei, B., and Wang, T. (2015). Accurate segmentation of cervical cytoplasm and nuclei based on multiscale convolutional network and graph partitioning. *IEEE Transactions on Biomedical Engineering*, 62(10):2421–2433.

Song, Y., Zhang, Y.-D., Yan, X., Liu, H., Zhou, M., Hu, B., and Yang, G. (2018b). Computer-aided diagnosis of prostate cancer using a deep convolutional neural network from multiparametric mri. *Journal of Magnetic Resonance Imaging*, 48(6):1570–1577.

Sperduti, A. and Starita, A. (1997). Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3):714–735.

Spiga, O., Cicaloni, V., Zatkova, A., Millucci, L., Bernardini, G., Bernini, A., Marzocchi, B., Bianchini, M., Zugarini, A., Rossi, A., et al. (2018). A new integrated and interactive tool applicable to inborn errors of metabolism: Application to alkaptonuria. *Computers in biology and medicine*, 103:1–7.

Steffener, J., Habeck, C., O'Shea, D., Razlighi, Q., Bherer, L., and Stern, Y. (2016). Differences between chronological and brain age are related to education and self-reported physical activity. *Neurobiology of aging*, 40:138–144.

Sun, Q., Yang, Y., Sun, J., Yang, Z., and Zhang, J. (2017). Using deep learning for content-based medical image retrieval. In *Medical Imaging 2017: Imaging Informatics for Healthcare, Research, and Applications*, volume 10138, page 1013812. International Society for Optics and Photonics.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

Teramoto, A., Fujita, H., Yamamuro, O., and Tamaki, T. (2016). Automated detection of pulmonary nodules in pet/ct images: Ensemble false-positive reduction using a convolutional neural network technique. *Medical physics*, 43(6Part1):2821–2827.

Tustison, N. J., Avants, B. B., Cook, P. A., Zheng, Y., Egan, A., Yushkevich, P. A., and Gee, J. C. (2010). N4itk: improved n3 bias correction. *IEEE transactions on medical imaging*, 29(6):1310.

Tversky, A. (1977). Features of similarity. *Psychological review*, 84(4):327.

van der Leest, M., Cornel, E., Israel, B., Hendriks, R., Padhani, A. R., Hoogenboom, M., Zamecnik, P., Bakker, D., Setiasti, A. Y., Veltman, J., et al. (2019). Head-to-head comparison of transrectal ultrasound-guided prostate biopsy versus multiparametric prostate resonance imaging with subsequent magnetic resonance-guided biopsy in biopsy-naïve men with elevated prostate-specific antigen: a large prospective multicenter clinical study. *European urology*, 75(4):570–578.

Visibelli, A., Bongini, P., Rossi, A., Niccolai, N., and Bianchini, M. (2020). A deep attention network for predicting amino acid signals in the formation of [formula: see text]-helices. *Journal of Bioinformatics and Computational Biology*, pages 2050028–2050028.

Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., and Wu, Y. (2014). Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1386–1393.

Wang, Z., Liu, C., Cheng, D., Wang, L., Yang, X., and Cheng, K.-T. (2018). Automated detection of clinically significant prostate cancer in mp-mri images based on an end-to-end deep neural network. *IEEE transactions on medical imaging*, 37(5):1127–1139.

Weinreb, J. C., Barentsz, J. O., Choyke, P. L., Cornud, F., Haider, M. A., Macura, K. J., Margolis, D., Schnall, M. D., Shtern, F., Tempany, C. M., et al. (2016). Pi-rads prostate imaging–reporting and data system: 2015, version 2. *European urology*, 69(1):16–40.

Wetzel, A. W., Crowley, R., Kim, S., Dawson, R., Zheng, L., Joo, Y., Yagi, Y., Gilbertson, J., Gadd, C., Deerfield, D., et al. (1999). Evaluation of prostate tumor grades by content-based image retrieval. In *27th AIPR Workshop: Advances in Computer-Assisted Recognition*, volume 3584, pages 244–253. International Society for Optics and Photonics.

Williams, R. J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280.

Xu, G. and Fang, W. (2016). Shape retrieval using deep autoencoder learning representation. In *2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pages 227–230. IEEE.

Xu, J., Xiang, L., Liu, Q., Gilmore, H., Wu, J., Tang, J., and Madabhushi, A. (2015). Stacked sparse autoencoder (ssae) for nuclei detection on breast cancer histopathology images. *IEEE transactions on medical imaging*, 35(1):119–130.

Yang, W., Chen, Y., Liu, Y., Zhong, L., Qin, G., Lu, Z., Feng, Q., and Chen, W. (2017a). Cascade of multi-scale convolutional neural networks for bone suppression of chest radiographs in gradient domain. *Medical image analysis*, 35:421–433.

Yang, X., Liu, C., Wang, Z., Yang, J., Le Min, H., Wang, L., and Cheng, K.-T. T. (2017b). Co-trained convolutional neural networks for automated detection of prostate cancer in multi-parametric mri. *Medical image analysis*, 42:212–227.

Yi, D., Lei, Z., Liao, S., and Li, S. Z. (2014). Deep metric learning for person re-identification. In *2014 22nd International Conference on Pattern Recognition*, pages 34–39. IEEE.

Zagoruyko, S. and Komodakis, N. (2015). Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4353–4361.

Zhang, C., Liu, W., Ma, H., and Fu, H. (2016). Siamese neural network based gait recognition for human identification. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2832–2836. IEEE.

Zhang, X., Dou, H., Ju, T., Xu, J., and Zhang, S. (2015). Fusing heterogeneous features from stacked sparse autoencoder for histopathological image analysis. *IEEE journal of biomedical and health informatics*, 20(5):1377–1383.

Ziegler, D. A., Piguet, O., Salat, D. H., Prince, K., Connally, E., and Corkin, S. (2010). Cognition in healthy aging is related to regional white matter integrity, but not cortical thickness. *Neurobiology of aging*, 31(11):1912–1926.