TÉCNICO LISBOA

Convolutional Neural Networks for the Classification of 3D Medical Images

Luís Henrique Vieira Pereira

Thesis to obtain the Master of Science Degree in

Biomedical Engineering

Supervisor: Prof. Bruno Emanuel Da Graça Martins

Examination Committee

Chairperson: Prof. Maria Margarida Campos da Silveira
Supervisor: Prof. Bruno Emanuel Da Graça Martins
Member of the Committee: Prof. David Manuel Martins de Matos

January 2021

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Preface

The work presented in this thesis was performed at the Instituto de Engenharia de Sistemas e Computadores, Investigação e Desenvolvimento of Instituto Superior Técnico (Lisbon, Portugal), during the period February 2020-January 2021, under the supervision of Prof. Bruno Martins.

Acknowledgments

I would like to thank my thesis advisor, Prof. Bruno Martins, as the successful completion of my dissertation would not have been possible without his constant advice, support and knowledge.

I also express my gratitude to INESC-ID for providing the conditions necessary to conduct my thesis. Additionally, I would also like to thank Instituto Superior Técnico for the contribution to my personal and academic growth.

To all my friends, who always cheered me on, I am truly thankful to have you by my side. These five years would have not been the same without you.

To my girlfriend, for her patience, friendship and support, and for keeping me motivated during hard times, I am very grateful.

Lastly, I would like to thank my family, especially my parents and sister who believed in me and gave me all the conditions to be successful, even far away from home.

Dedicated to my mother, who is gone but always present. You made me who I am.

Abstract

Medical imaging is a fundamental screening and diagnostic tool. Healthcare professionals can nowadays rely on various types of image modalities of the human body, including three-dimensional images such as magnetic resonance images. However, with increasingly more information and image complexity, the pressure that radiologists are subjected to is ever increasing, and the resulting fatigue can lead to diagnostic errors. Machine learning mechanisms have been proposed for the analysis of medical images, although most previous work has dealt with inputs involving two dimensions. This work proposes an approach based on convolutional neural networks, combined with recurrent neural networks, for the classification of three-dimensional medical images. The proposed architecture aims to extract features from the individual slices of the three-dimensional image, using a convolutional network, and correlate them with the three-dimensional nature of the original images using a recurrent neuronal network. Experiments were carried out with different architectures to classify three-dimensional images of the knee, leveraging a publicly available data-set. The results show that the main model presented in this work, based on the ResNet architecture and LSTM units, is efficient for the classification of this type of images, despite being relatively simple.

Keywords

Artificial Intelligence, 3D Image Classification, Deep Learning, Convolutional Neural Networks, Recurrent Neural Networks

Resumo

Imagiologia médica é nos dias de hoje uma ferramenta fundamental de triagem e diagnóstico, que fornece vários tipos de imagem do corpo humano, inclusivamente imagens em três dimensões, como por exemplo a modalidade ressonância magnética. Porém com o aumento de informação e da complexidade das imagens, a pressão à qual os radiologistas estão sujeitos é cada vez maior, pelo que a fadiga pode levar a erros de diagnóstico. Mecanismos de aprendizagem automática têm sido propostos para a análise destas imagens, embora em duas dimensões. Esta dissertação propõe uma abordagem baseada em redes neuronais convolucionais pré treinadas, combinadas com redes neuronais recorrentes para a classificação de imagens médicas tridimensionais. Esta arquitectura proposta pretende extrair características dos cortes individuais da imagem tridimensional, com a rede convolucional, e relacionar as características extraídas com a natureza tridimensional da imagem original usando redes neuronais recorrente. Foram realizadas experiências com diversas arquitecturas para classificar imagens tridimensionais do joelho. Os resultados mostram que o modelo apresentado nesta dissertação, baseado na arquitectura ResNet e em unidades LSTM, é eficiente para a classificação deste tipo de imagens, apesar de ser relativamente simples.

Palavras Chave

Classificação de Imagens 3D, Aprendizagem com Redes Neuronais Profundas, Redes Neuronais Convolucionais, Inteligência Artificial, Redes Neuronais Recorrentes

Contents

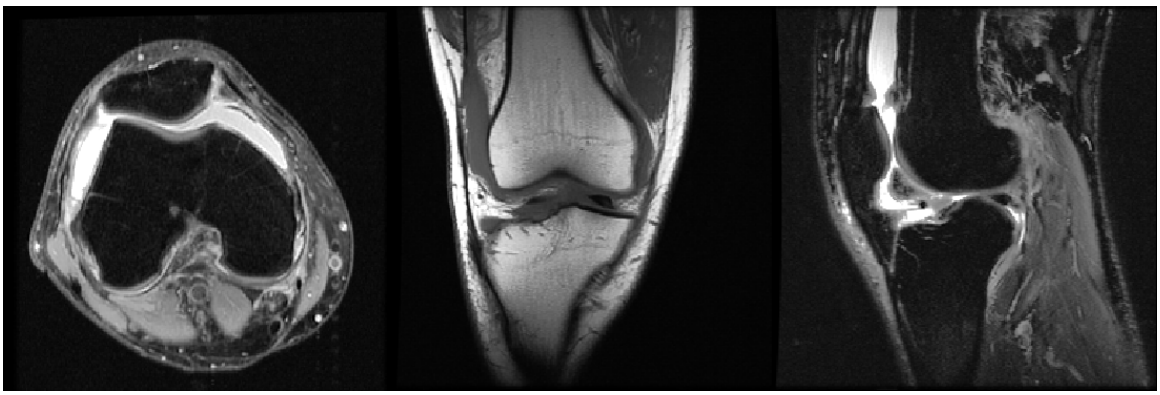
List of Figures

List of Tables

1

Introduction

Contents



(a) Axial Plane (b) Coronal Plane (c) Sagittal Plane

Figure 1.1: Slice of a multi-planar knee MRI from MRNet [1] data set.

Medical imaging is a fundamental screening and diagnosis tool with widespread use in modern medicine. Common types of medical imaging include computed tomography (CT), positron emission tomography (PET), and magnetic resonance imaging (MRI) (Figure 1.1). These scans give detailed three-dimensional (3D) images of human organs and can be used to detect infection, cancers, traumatic injuries, and abnormalities in blood vessels and organs. With the advances of modern medicine and imaging, the modalities of medical imaging that use more than a single image for diagnosis, have started to become more readily available, with better image quality. Yet, despite having increasingly better imaging instruments, every medical image still requires a specialized medical professional for analysis and diagnostic. As information in these scans grows and becomes more complex, these tasks handed to radiologists become more fatiguing and time-consuming, which can lead to misdiagnosis and medical error, as well as increasing costs per exam.

With increasingly more databases of medical images and the advent of high performance computers, machine learning methods can now play a crucial role in assisting clinicians, providing numerous benefits, from improving workflow to supporting clinical decisions. For instance, several supervised deep learning approaches to classify X-ray images [2] have been successfully tried and implemented. However, in medical data that comprises multidimensional and multi-planar images, traditional approaches often fail due to the complexity of the data being handled. As a consequence, implementations related to the analysis of three dimensional medical images, e.g. MR imaging, remain not well explored, though it must be said that in recent years work in this domain, i.e., deep learning publications on the analysis 3D images, has been growing rapidly.

Existing approaches based on 3D deep neural networks have indeed been tried, and successfully gained on traditional image analysis methods, enabling significant progress in medical imaging tasks [3, 4]. Most of these methods rely on convolutional neural networks (CNNs), extended to 3D data,

which have proven to be dependable on the classification and segmentation of two dimensional (2D) images. More recently, frameworks that combine CNNs and recurrent neural networks (RNNs) have been pushed forward [5], looking at the classification problem from a different perspective. Instead of relying on three-dimensional layers, these networks decompose the original sample into slices that can be fed into two-dimensional layers.

This dissertation intends to expand the previously developed methodologies, by studying current and novel deep learning methods for image classification, implementing a combination of state-of-the-art methods in a three-dimensional deep convolutional neural network as a way to improve classification results.

1.1 Objectives

Classification of three-dimensional medical images by means of deep learning is still on the early stages of development and implementations. This dissertation intends to explore novel approaches while studying previously developed methodologies to address how deep neural networks can be used to classify three-dimensional medical images.

The proposed task also seeks to explore: (1) Application of transfer learning using feature extraction on a classification problem; (2) Combined use of Long Short Term Memory units (LSTMs) together with convolutional neural networks and its impact in multidimensional images; (3) Development of an end-to-end model capable of classifying multiple lesions.

1.2 Methodology

To better understand the complexity of the problem in hand, the first step was to perform a related work revision focused on 2D and 3D image classification, especially on areas were medical image classification was the main focal point. From that, the jumping-off point was the work of Bien et al. [1] on 3D deep learning CNNs for medical image classification.

Discovering what approach would fare best in the task of classifying medical images, occurred through the comparison of several models created to meet this end. These models evolved from different pre-existing architectures to process image data, such as VGG architecture [6], ResNet architecture [7] and DenseNet architecture [8], all pre-trained on the ImageNet [9] data set. A main architecture was then defined, with an alternative approach also being suggested.

The predictive capability made by the different architectures was evaluated using the performance metrics of accuracy, precision, recall, and area under ROC curve (AUC).

Testing our different models relied on the publicly available MRNet [1] data set that contains knee

magnetic resonance (MR) images from three different planes (i.e., axial plane, coronal plane, sagittal plane), labeled according to 3 observational classes. This dataset had a total of 1250 exams and was divided into two different subsets, one for training (90%) and one for testing (10%). Training models with the knee MR images was made with batches of 32 instances, leveraging back-propagation together with the Adam optimization method [10].

Python was utilized to program our architectures, due to abundance of machine learning methods and support resources available. Keras[1], using TensorFlow[2] as a computational backend, was the library chosen to be the main workhorse of this dissertation. A small number of other libraries used to support our work and architecture were numpy[3], scikit-image[4] for image augmentation, scipy[5], and scikit-learn[6].

1.3 Contributions

The contributions made in this M.Sc. research project are:

- A novel CNN-LSTM deep learning approach to classify 3D medical images. The neural network combines a time distributed feature extractor CNN based on ResNet50 architecture with LSTM units to produce a prediction.
- The introduction of logistic regression to weight and aggregate the several predictions from the different planes and generate a single output per label.
- Comparison of different methods and network configurations concerning the importance of the spatial correlation of 3D image slices. Using LSTM units after feature extraction contributed to a significantly better performance of the model.
- The introduction of a single multi-label CNN-LSTM model capable of classifying several lesions in a single 3D image plane.

1.4 Structure of the Document

The present dissertation is organized as follows. Chapter 2 touches important concepts of machine learning applied to image classification and related work that are the underpinning of the dissertation. Chapter 3 details the proposed methodology, presenting a deep neural network architecture developed

[1] https://keras.io
[2] https://www.tensorflow.org
[3] https://numpy.org
[4] https://scikit-image.org
[5] https://www.scipy.org
[6] https://scikit-learn.org

for this work. Then Chapter 4 specifies the data set, evaluation procedure and obtained results. Finally, Chapter 5 compiles the conclusions of this document and presents directions for future work.

2

Concepts and Related Work

Contents

This chapter elucidates on fundamental concepts needed to understand the work developed on this dissertation. It also presents previous studies related to image classification methods and ,subsequently, reports on works pertinent to medical image classification. Section 2.1 overviews the main concepts of artificial neural networks applied in this work, defining the notion of perceptrons, convolutional neural networks, and recurrent neural networks. Section 2.2 reviews previous studies conducted in the field of image classification tasks. Then, Section 2.3 presents an overview of previous work focusing on medical image classification. To conclude the chapter, Section 2.4 presents a summary of the related work here described.

2.1 Introduction to Neural Networks

This Section introduces essential concepts of neural networks (i.e., perceptrons, convolutional neural networks and recurrent neural networks) to better understand the proposed solution to classify 3D medical images.

2.1.1 Single Layer and Multi-Layer Perceptron

Artificial Neural networks (ANNs) are machine learning methods that take inspiration from our own biological being, i.e. the biological neural networks that form animal brains, allowing them to possess learning capabilities.

Neural networks are, at the most basic level, composed of perceptrons, which are connect to each other in layers in order to map the inputs into the targeted outputs. These layers can be seen as nested functions whose parameters can be trained directly to minimize a given loss function computed over the outputs and the expected results.

In its simplest form, a single-node neural network as shown in Figure 2.1, is used for supervised learning of binary classifiers. The perceptron computes a single output by using a function to linearly combine the multiple real-valued inputs with a set of input weights. These weights are an important part of the learning component, as varying weights change the function that the node calculates. To avoid dependability of any input value a bias is used to shift the decision boundary away from the origin. An activation function is used to produce the final output classification, given the previous calculate linear combination and a threshold.

In a mathematical form, Equation 2.1 shows how the single-node neural network can be written, where *y* refers to the output prediction, $\mathbf{x} = (x_1, \cdots, x_n)$ is the vector of inputs, **w** denotes the vector of weights, *b* is a bias term, and ϕ is an activation function.

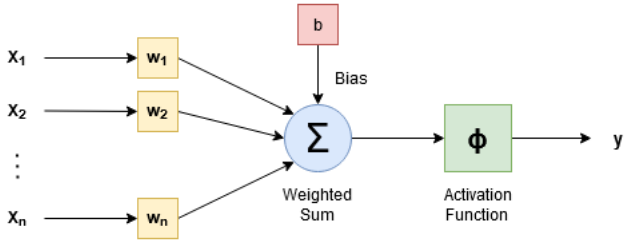


Figure 2.1: The single perceptron classification model.

$$y = f(x) = \phi\left(\sum_{i=1}^{n} w_i \times x_i + b\right) = \phi(\mathbf{w} \cdot \mathbf{x} + b) \tag{2.1}$$

Training the perceptron model requires a training set of inputs **x** together with the corresponding labels. With this data, at each training instance an error is calculated for the given output. Then the calculated error *e*, bias *b* and the weight vector **w** are updated according to a learning rate *r* (see Equation 2.2). The learning progress stops when, after iterating through all the data several times, a convergence criteria is met.

$$\begin{aligned} b &= b + r \times e \\ w_i &= w_i + r \times e \times x_i \end{aligned} \tag{2.2}$$

Granting that a single-node neural network has limited mapping ability, the conjunction of several of these nodes into blocks can be used to build a more complex model. A Multi-Layer Perceptron (MLP) builds on this idea, as it consists of a set of nodes forming the input layer, one or more hidden layers of computation nodes, and an output layer of nodes. The information flows from the input layer through the network layer-by-layer, until it reaches the output. MLPs are also typically refered to as Feed-forward networks and in the case of a single hidden layer, MLP can be mathematically interpreted as:

$$y = f(x) = \phi(\mathbf{B} \times \phi'(\mathbf{A} \cdot \mathbf{x} + \mathbf{a}) + \mathbf{b}) \tag{2.3}$$

In Equation 2.3, **x** is a vector of inputs and *y* a vector of outputs. The matrix **A** represents the weights of the first layer and **a** is the bias vector of the first layer, while **B** and **b** are, respectively, the weight matrix and the bias vector of the second layer. The functions $\phi'(.)$ and $\phi(.)$ both stand for an element-wise non-linearity, as result of activation functions respectively associated to nodes in the hidden layer, and in the output layer.

As in the perceptron model, an MLP neural network is trained by adapting weights and bias to optimal values, so that the output of the model *y* matches with the real label. The back-propagation algorithm [11]

is often used to train this type of networks. This learning technique consists of a forward pass, where the output values of the model are evaluated and the cost function determined, and of a backward pass, where the partial derivatives of a given cost function, corresponding to the different parameters, are propagated backwards throughout the layers, assigning each layer's weight responsibility for a portion of the error.

To minimize cost function, i.e., the difference between predicted values and actual values, a gradient descent algorithm is used to adjust weights and biases of the network in the opposite direction of the gradient. The algorithm has different versions that vary by the amount of data that there is to compute the gradient of the cost function.

Batch gradient descent injects all data at once, computing the gradient of all the data set in one update. This approach has a straight trajectory towards the minimum and it is guaranteed to converge in theory to the global minimum, if the cost function is convex, and to a local minimum, if the cost function is not convex. However, batch gradient descent can be time consuming and the size of the data set must be taken into account, as it is only possible to use this version if the data can fit in memory available.

Another version is stochastic gradient descent, which, instead of going through all examples, performs the parameters update on each example x_i, y_i. Therefore, learning happens on every example. This can lead the cost function to not converge at all and to progress very slowly. On the other hand, it has a high variation as the gradient is calculated for a specific sample only at a time, helping to improve the generalization error.

The final variant, mini-batch gradient descent updates for every mini-batch of *n* training examples. This version has the advantages of both anterior versions and trains faster due to the parallelization of operations.

Improvements to these methods can be made, and one of the most impacting can be on the learning rate parameter. Adaptive Moment Estimation (Adam) [10] is a gradient descent optimization algorithms extensively used for this purpose, computing parameter updates leveraging an exponentially decaying average of past gradients, together with adaptive learning rates for each parameter.

2.1.2 Convolutional Neural Networks

For more complex applications, such as image processing, MLPs have limitations due to the number of parameters associated with images. MLP networks use dense interactions between every input and output unit making their use prohibitive. Convolutional Neural Networks (CNNs) tackle this problem by having the neurons within a layer only connecting to a small region of the layer preceding it, and thus using common parameters to process all these small regions.

In more detail, CNNs are typically comprised of three types of layers (see Figure 2.2). Convolutional layers determine the output of neurons connected to local regions of the input, through the calculation

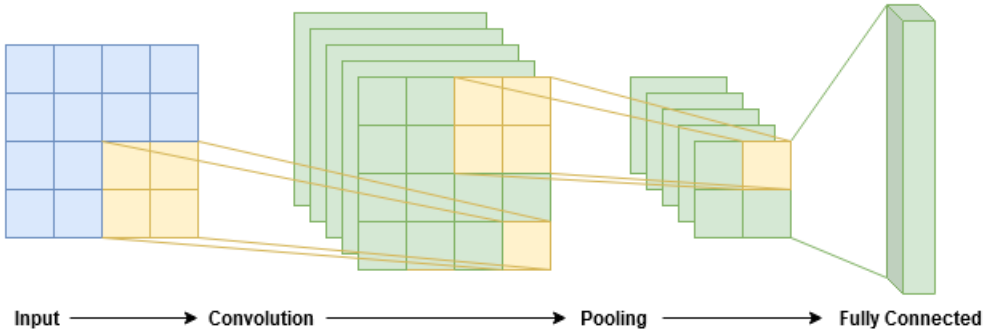


Figure 2.2: A simple CNN architecture.

of the scalar product between the layer's weights and the region connected to the input volume, followed by an activation function. These regions are decided by specifying the size and number of kernels, also known as filters, to be applied. The number of filters correspond to the depth of the output produced by a convolutional layer. Each filter is convolved across the spatial dimensionality of the input (1D, 2D, 3D), producing a feature map. These maps are stacked along the depth dimension to form the output volume from the convolutional layer. Padding in convolutional layers is generally used, as it prevents the feature maps from shrinking by surrounding the input with zeros. Mathematically, a feature map of a convolution layer can be represented by the following Equation.

$$G[m,n] = (f * h)[m,n] = \sum_{j} \sum_{k} h[j,k] \times f[m-j, n-k] \tag{2.4}$$

In Equation 2.4, *G* corresponds to the final feature map, produced by the convolution of kernel *h* with the input picture denoted by *f*. The indexes of rows and columns of the feature matrix are represented with *m* and *n* respectively.

Convolutional layers are often used together with the rectified linear unit (ReLU) [12] activation function. This function applies $\sigma(x) = \max(0, x)$, removing negative values form the activation map so that the final feature map values is the result of the ReLU function applied to the summed inputs. It increases the nonlinear properties of the decision function and of the overall network without affecting the receptive fields of the convolution layer.

Pooling layers follow convolution layers, and use an aggregation operation to down-sample each feature map independently, reducing both the height and width so that the number of parameters is reduce, while persevering the depth intact. This type of layers is frequently used between consecutive convolutional layers. Common types of pooling layers include, max-pooling layers, which returns the maximum value in the pooling window, and average pooling layers that return the average value in the pooling window.

Finally, the last type of layers are fully-connected layers. These layers are located at the end of the model, just before the output, and are similar to those in MLPs, seeking to emulate the desired outputs and generate the final representations.

2.1.3 Recurrent Neural Networks

Another example of neural networks are Recurrent neural networks (RNNs). In this type of architectures, the model not only looks at the current input but also at what as happen one step before in time. This means that after producing an output, it is copied and sent back to the recurrent network.

RNNs differ from feed forward neural networks as an internal state can be used to process sequential data. This can be useful in tasks where the inputs are not independent from each other, e.g. image recognition on images that have multiple slices. In brief, given a sequence $x = (x_1, x_2, \cdots, x_t)$, an RNN takes as input x_1 from the sequence and outputs the hidden state h_1, that together with x_2 are the input of the next cell. This goes on until the end of the sequence is reached. The next equations represent how hidden states and the cell output evolve over time, respectively.

$$\begin{aligned} h_t &= g(h_{t-1}, x_t, \theta) \\ y_t &= f(h_t, \theta) \end{aligned} \tag{2.5}$$

The first equation says that, given parameters θ (which are composed of weights and bias for the model), the hidden state at time *t* is dependent on the previous hidden state h_{t-1} and the input, given from the sequence of data, x_t. This part of the RNN is what demonstrates that this type of network has ”memory”, as the previous calculations of the hidden state are able to influence the current calculation of the hidden state. The second equation shows that, the output at time *t* is only dependent to the hidden state computed at the same time *t* with the same given parameters θ.

However, one issue in general with this type of network is that it suffers from the vanishing gradient problem, i.e. the repeated multiplications of the gradient while back-propagating to earlier layers, causes its value to become significantly smaller, saturating the performance of the network. Therefore, when dealing with long sequences of data, RNNs have difficulties modelling relationships between inputs separated by large periods of time.

Long short-term memory (LSTM), first presented by Hochreiter and Schmidhuber [13], is an RNN variant that is capable of overcoming the previously exposed problem. These units are able to be applied to long sequences of data while being quick and stable, thus making them more useful in practice.

LSTM, as seen in Figure 2.3, can be explained as a neural network that accomplish modeling sequential data by having a recurrent hidden state regulated by gates. The key to an LSTM is the cell state and the ability to update it by using gates. At time step *t*, that in our proposed aproach corresponds to a slice *t* for a given input sequence of a three dimensional image, a sigmoid layer called ”forget gate” $\mathbf{f}_t$

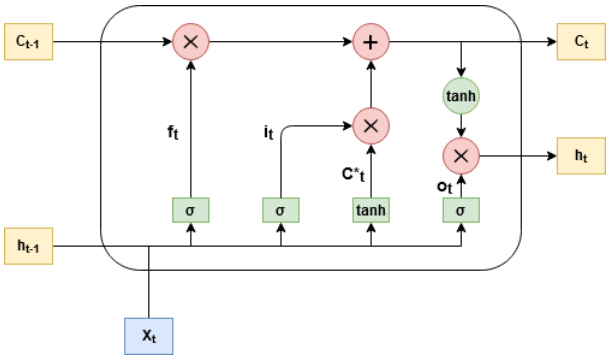


Figure 2.3: Illustration of a LSTM unit.

decides which part of the memory cell will be forgotten or kept, an "input gate" $\mathbf{i}_t$ controls which values are going to be updated and a *tahn* gate $\tilde{\mathbf{c}}_t$ creates a vector of new candidate values. This last two combine to create an update to the cell state. Lastly a gate $\mathbf{o}_t$ produces an output based on a filtered cell state. These gate values are calculated through linear combinations of the current input $\mathbf{x}_t$ and the previous state $\mathbf{h}_t$ with a sigmoid function (σ). An LSTM units can be formally defined as follows.

$$\begin{aligned}
\mathbf{f}_t &= \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + b_f) \\
\mathbf{i}_t &= \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + b_i) \\
\tilde{\mathbf{c}}_t &= \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + b_c) \\
\mathbf{o}_t &= \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + b_o) \\
\mathbf{c}_t &= \mathbf{c}_{t-1} \circ \mathbf{f}_t + \tilde{\mathbf{c}}_t \circ \mathbf{i}_t \\
\mathbf{h}_t &= \mathbf{o}_t \circ \tanh \mathbf{c}_t
\end{aligned} \tag{2.6}$$

In the Equation 2.6, the matrices $\mathbf{W}_q$ and $\mathbf{U}_q$ contain the weights of the input and recurrent connections, respectively, where *q* can either be the input gate *i*, output gate *o*, the forget gate *f* or the memory cell *c*, depending on the activation being calculated.

An alternative to LSTMs are Gated Recurrent Units (GRUs). These were introduced by Cho et al. [14] and are similar to LSTMs. However, GRUs have only two gates (a reset gate and a update gate) and there is no cell state, as the hidden state is now responsible for transferring information.

2.2 Deep Learning for Image Classification

One of the earliest successful adoptions of convolutional neural networks for image recognition was introduced by Lecun et al. in 1998 [15]. In the study, the authors proposed an architecture called LeNet outperformed all other existing models, at the time, for handwritten digit recognition tasks.

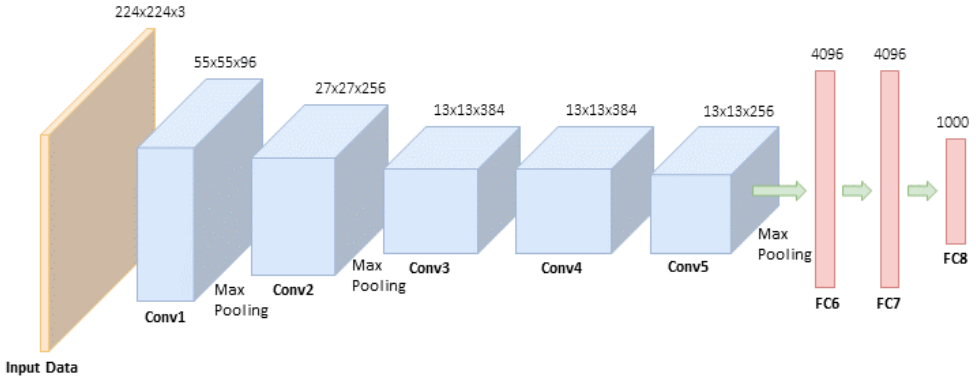


Figure 2.4: Illustration of AlexNet architecture.

The architecture of LeNet5 consists of an input layer, that receives a 36×36 size image, and six other trainable layers. From the fed data, the network follows with convolutional layers interspersed with sub-sampling layers. Then the resulting representation is passed to a fully connected layer that classifies the input image. In more detail, there are three convolutional layers with kernel sizes of 5×5, called C1, C3, and C5, that are intertwined with two layers of sub-sampling operations, S2 and S4 with pooling sizes of 2×2, responsible for reducing the size of data computed. The last layer corresponds to a fully connected layer, F6, which is fully connected to the previous layer C5, and outputs 84 graphs.

Since LeNet was proposed, several different versions of CNN have been proposed to improve model performance. In 2011, Krizhevsky et al. [16] presented a deep CNN architecture to perform image classification and recognition. The proposed approach in AlexNet (Figure 2.4), consists of five convolutional layers and three fully connected layers, where the output of the last fully connected layer is applied to a softmax activation, providing the predictions. The model was trained on 1.2 million images from the large data set ImageNet [9], with 60 million parameters. To tackle such large number of parameters, AlexNet was trained on a multi-graphic processing unit (GPU) (i.e. two GPUs) environment by systematically distributing the neurons on both the GPUs. To reduce over-fitting, the authors turned to data augmentation and dropouts methods. The data augmentation was performed in two ways: with image translations and horizontal reflections, were a random patch of 224×224 (and its flipped version) was extracted from a 256×256 image and then fed to the network; and with changes of the intensity of the RGB channels by performing Principal Component Analysis (PCA) on the pixels. The dropout technique also used to reduced over-fitting, consisted of setting to zero the output of each hidden neuron with probability 0.5, so that if a neuron was dropped it did not contribute to the forward or to the back-propagation of the error. The AlexNet architecture won the ILSVRC-2012 (ImageNet Large Scale Visual Recognition Competition 2012) [17] by large margin to other previous state-of-the-art models. The difference between the top-five test errors of AlexNet (15.3%) and the second prize winner (26.2%) was around 10%.

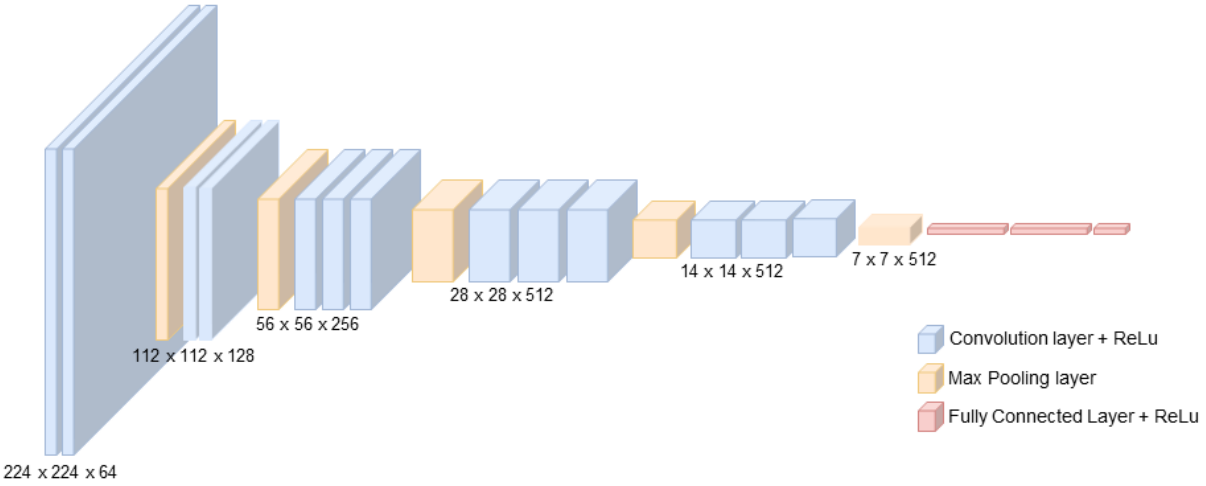


Figure 2.5: Illustration of VGG16 architecture.

In 2014, Simonyan and Zisserman [6] realise the importance of depth in convolutional neural networks and presented a more profound deep network architecture called Visual Geometry Group (VGG) network, illustrated in Figure 2.5. The approach taken by the authors confirmed that adding more convolutional layers would improve results accuracy in image recognition tasks. Proof of that was the ILSVRC in 2014 [17], were VGG was able to secured the first position for the localization task and the second position for the classification task. In more detail, the VGG network takes as input an image of size 224×224 and uses 3×3 filters for convolutional layers with 2×2 pooling layers followed by two consecutive fully connected layers of size 4096 and a softmax activation layer. However, this approach has its drawbacks, as adding consecutive layers to the network increases the number of parameters that cause networks to suffer from errors and over-fitting, and makes them generally more difficult to train. This was proven by the authors, by testing different depth configurations over the architecture. It was noted that by increasing the depth of the network from 11 to 19 layers the error declined, but once the network reached 19 layers the same error saturated. Simonyan and Zisserman (2014) found that the models with 16 and 19 layers were the better performers.

In supervising learning, a deeper network typically means a bigger number of parameters, which makes the network more prone to over-fitting, especially if the size of the used training set is small. Another limitation of increasing the neural network size is the huge increase of computational resources, as a linear increase in the number of filters is synonym of a quadratic increase of computation. This is exacerbated if, in deeper layers, weights come close to zero, resulting in a waste of computational resources.

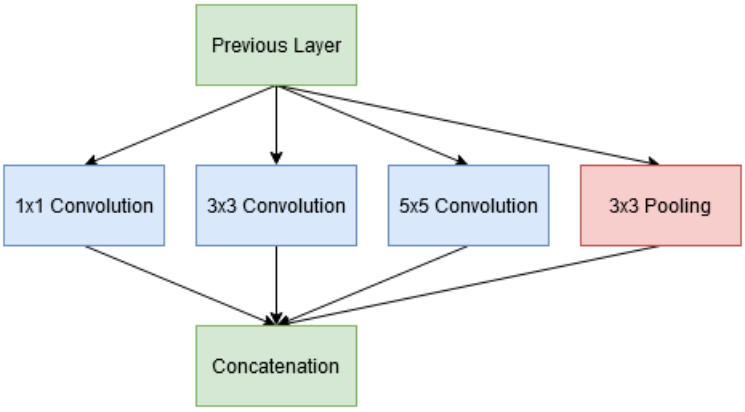


Figure 2.6: Illustration of the Inception module.

With the objective of maintaining a sustainable computational budget, Szedegy et al. introduced in 2014 the Inception network [18]. The main idea behind the inception module is that the optimal neural network topology can be built by clustering neurons to the correlation statistics in the input images, i.e. the analysis of the correlation statistics in the previous layer of activations and the clustering of the neurons with highly correlated outputs for the next layer. In images, the correlation tends to be local, and therefore, performing convolutions over the local patches can cluster the neurons. In the lower layers (i.e. the closer ones to the input), there is a high correlation between local pixels in a surrounding patch, thus these can be covered by a small 1×1 convolution. Nonetheless, it is expected that there will be a smaller number of spatially spread-out clusters, and to that end these can be quantified by 3×3 and 5×5 convolutions. In order to take effect of 1×1, 3×3 and 5×5 convolutions, the authors combined them along with a 3×3 pooling layer, to serve as input to the next layer. This inception module is illustrated in Figure 2.6. The introduced concept of the inception module in Inception alleviates the problem of vanishing gradients and allows us to move deeper into the network. However to decrease computational expenses, the authors suggest reducing network dimensions by introducing Inception-V2 and Inception-V3 [19]. In Inception-V2, inexpensive 1×1 convolution convolutions were inserted to reduce dimensionality, before the expensive 3×3 and 5×5 convolutions are performed. After the convolutions are performed, all three are concatenated together in conjunction with the max-pooling operation. In addition, in this module, the 1×1 convolution used ReLU activation. Inception network was entered in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14) [17] and was able secured the first position in the classification task as well as in the object recognition task.

As already stated, very deep architectural neural networks are often difficult to train due to the problem of vanishing and exploding gradients. Considering this problem, He et al. [7] presented ResNet to avoid the degradation issue. ResNet uses skip connections, which allows us to take the activation

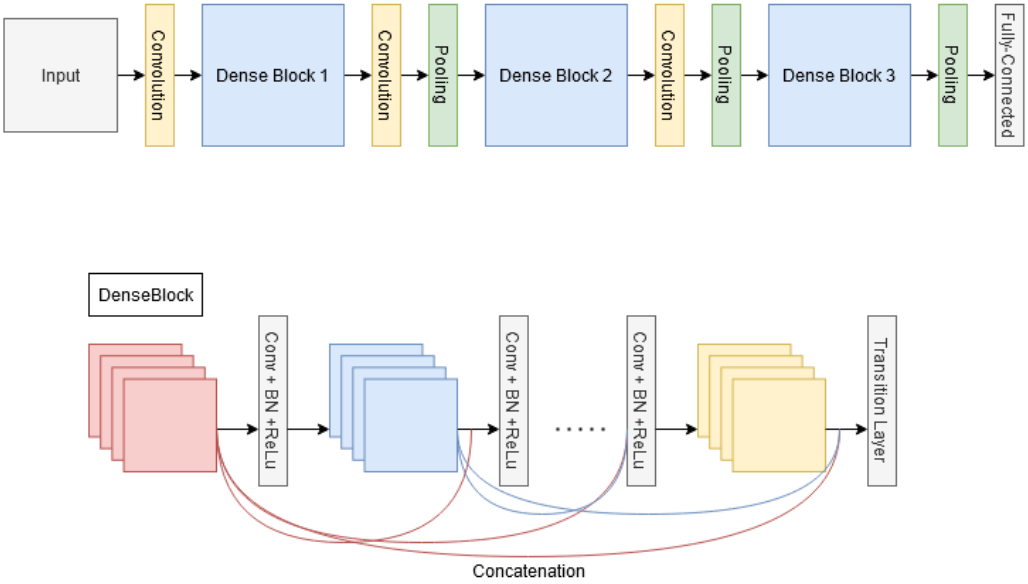


Figure 2.7: Illustration of the DenseNet architecture and a DenseBlock.

from one layer and suddenly feed it to another layer, skipping one or more convolutional layers. Original ResNet uses batch normalization after each convolutional layer and before the activation. Using these skip connections, we can train very deep network architectures, without having vanishing gradients. ResNet is one of the most popular deep learning architectures in the literature. When it was first present, ResNet secured first place in ILSVRC-2015 with a 3.57% test error on the ImageNet dataset. This model was used in our approach and is detailed in Chapter 3.

With the idea of short connections between layers close to the input and those close the output introduced by ResNet, Huang et al. in 2017 presented Dense Convolutional Networks (DenseNets) [8] for image classification. The authors argue that CNNs can be significantly deeper, more accurate, and efficient to train if they hold shorter links between initial and later layers. The authors propose to use additional connections to alleviate the vanishing-gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters. Each layer's input consists of all the output feature-maps of all the preceding layers. Therefore, the output feature-maps of a given layer will be used as part of the inputs of all the subsequent layers, as illustrated in Figure 2.7. This means that we can interpret the flow of feature-maps as the global state of the network, as each layer contributes with k feature-maps of its own to the global state. Thus, the value of k represents the growth rate of the network, adjusting the quantity of new information each layer provides to the global state. To improve the computational efficiency of the model proposed, the authors suggest two variants to the main network. One of those variants introduces a bottleneck layer that adds a 1×1 convolution before

each 3×3 convolution, reducing the number of input feature maps. Another technique proposed by the authors, is the addition of a compression layer that reduces the dimensions of the layer's output feature maps by generating (θm) output feature maps, where the compression factor θ can have the values between 0 and 1 (including 1). The proposed architecture was evaluated on four object recognition benchmark tasks. In general, although requiring less computation, the model was able to achieve improvements over the state-of-the-art on most data sets. The results also showed that without compression and bottleneck layers, DenseNets seem to perform better as the number of layers and growth factor increases. In turn, DenseNets with bottleneck and compression were shown to be more parameter-efficient, achieving robust results per number of parameters of the network. This characteristic can also be interpreted as being less prone to over-fitting.

2.3 Deep Learning Methods for 3D Medical Image Analyzis Tasks

In a very short time, deep learning techniques have become an alternative to many machine learning algorithms that were traditionally used in medical imaging. With the appearance of larger data sets of labeled medical images, deep learning methods to perform classification or segmentation tasks have achieved performances similar or even better of clinical experts [1, 20]. Recent examples of publicly available data sets, supporting this type of developments, include the MRNet data-set [1] for the classification of MR knee scans.

Previous work on the analysis of two dimensional medical images, such as X-rays, has proved to be successful. For instance Rajpurkar et al. [2], found that an algorithm based on Densely Connected Convolutional Networks (DenseNet) can detect and localize lesions at a comparable rate to radiologists. With decreasing computational costs and more availability of better graphic processing units, three dimensional medical images also became possible targets for for deep learning methods.

A state-of-the-art approach by Korolev et al. [21] constructed a full 3D CNN, trained to classify Alzheimer's Disease (AD) using MRI data from the Alzheimer's Disease Neuroimaging Initiative (ADNI). In their work, some well-known baseline 2D deep architectures, such as VGGNet and ResNet, were converted to their 3D counterparts. However models like these have millions of parameters and are harder to train on smaller data sets.

A different approach followed by Bien et al. [1] is another example for the use of deep learning on the task of classifying 3D medical images. The authors proposed the use of a well-know 2D deep architecture, AlexNet [16], to perform feature extraction on each slice. They then combined the vectors obtained to generate an encompassing representation, before finally using a logistic regression model to generate a single prediction for each exam. On the MRnet dataset, the authors managed to obtained a high classification accuracy (namely 85%, 86,7% and 72,5% on abnormal detection, anterior cruciate

ligament (ACL) tear detection, and meniscal tear detection respectively) which compared fairly with the clinical expert's accuracy.

Another application of deep learning for evaluating knee MR images was brought forward by Liu et al. [22]. The authors developed a fully automated deep learning–based cartilage lesion detection system by using a joint segmentation and classification convolutional neural network. The proposed model was trained on a small data set of MR knee images and the obtained results were compared to practicing clinicians. The results indicated a high overall diagnostic accuracy for detecting cartilage lesions.

A similar approach to the three dimensional classification solution proposed by in this work, was presented by Liu et al. [5] in 2018. The paper proposed a framework of a conventional CNN and a Gated Recurrent Unit (GRU) to learn and classify Fluorodeoxyglucose Positrons Emission Tomography (FDG-PET) images, sequenced into two dimensional slices. The architecture achieved a good performance on the classifications of alzheimer's disease (AD) and mild cognitive impairment results (MCI), on images of the ADNI dataset.

Nokivok et al. [20] reported on the use of a Convolutional Long Short-Term Memory (C-LSTM) network in 3D scans, to address the issues caused by implementing a 3D CNN approach. In brief, the proposed model processes 3D volumetric scans as a time-series of 2D slices, using time distributed convolutions. It then feeds the output of the convolutions onto a bidirectional C-LSTM block, in order to leverage spatio-temporal correlations of the order-preserving slices. The neural network showed competitive and sometimes superior performance on liver and vertebrae segmentation tasks, leaving the authors to plan about future use of this model on other imaging tasks such as classification.

2.4 Overview

Image classification is one of the fields in which the application of machine learning methods has most successfully contributed over the years. Among those methods, Section 2.2 reported six deep learning models. The first described method was a pioneering neural network, named LeNet, which is characterized by the low number of convolution layers and the inability of processing high-resolution images. As an attempt to address this issue, the AlexNet was introduced, a significantly deeper LeNet based model, with better results' accuracy. The next introduced network was the VGG network, which outperforms the majority of the previously introduced networks, while being a considerable deep network with a high number of trainable parameters. Nonetheless, having deep and parameter overloaded networks increase the computation budget. The Inception was announced as being a deep network with great accuracy results while maintaining the complexity constant. ResNet introduced skip connections, to tackle vanishing gradient problems in deep networks, being one of the most popular deep learning architecture.

Lastly, Section 2.3 introduced state-of-the-art applications of deep learning in medical images segmentation and classification tasks. Several ideas from the works reported in this chapter were considered in our approach, namely time-distributed wrappers, ResNet and the LSTM networks. All papers presented in this chapter are summarized in Table 2.1.

Table 2.1: Summary of the related work present in this dissertation.

Author	Task and Method	Results and Conclusions
Lecun et al. [15]	Introduced a simple CNN architecture (LeNet) to perform handwritten digit recognition.	Outperformed all other existing models, at the time, for handwritten digit recognition tasks.
Krizhevsky et al. [16]	Proposed a deep CNN architecture (AlexNet) to perform image classification and recognition.	AlexNet architecture won the ILSVRC-2012 by a large margin.
Simonyan and Zisserman [6]	Introduced VGGNet for image classification and recognition.	Secured the first position for the localization task in ILSVRC-2014.
Szedegy et al. [18]	Presented the Inception network for image classification and recognition.	Won the ILSVRC14 in image classification and recognition. Introduced the concept of inception module.
He et al. [7]	Proposed ResNet for image classification and recognition	Introduced Residual blocks with skip connections to avoid degradation issues. Won ILSVRC-2015 with a 3.57% test error.
Huang et al. [8]	Proposed DenseNet to perform image classification and recognition.	Introduced Dense Blocks to alleviate the vanishing-gradient problem and reduce the number of parameters.
Rajpurkar et al. [2]	Used a DenseNet architecture to classify x-ray chest images.	The proposed model was able to detect and localize diseases at a comparable rate as radiologists.
Korolev et al. [21]	Converted well known 2D architectures to 3D for AD classification on brain MRI.	Results were comparable to other modern techniques. Main advantage was ease of use.
Bien et al. [1]	Used AlexNet for feature extraction of each slice and combined vectors to classify knee MRI.	Achieve 85%, 86,7% and 72,5% on abnormal detection, ACL tear detection, and meniscal tear on the MRnet data set.
Nokivok et al. [20]	Proposed an approach that combined a time-distributed wrapped CNN with C-LSTM for CT image segmentation.	Results on liver and vertebrae segmentation showed that C-LSTM had better performance over a state-of-the-art model.
Liu et al. [22]	Developed cartilage lesion detection system by using VGGnet for segmentation and classification.	The results obtained indicated a high overall diagnostic accuracy for detecting cartilage lesions.
Liu et al. [5]	Proposed a framework that combines conventional CNN and a gated recurrent unit (GRU) perform image classification.	The architecture achieved a good performance ont he ADNI data set classification task.

3

Methodology

Contents

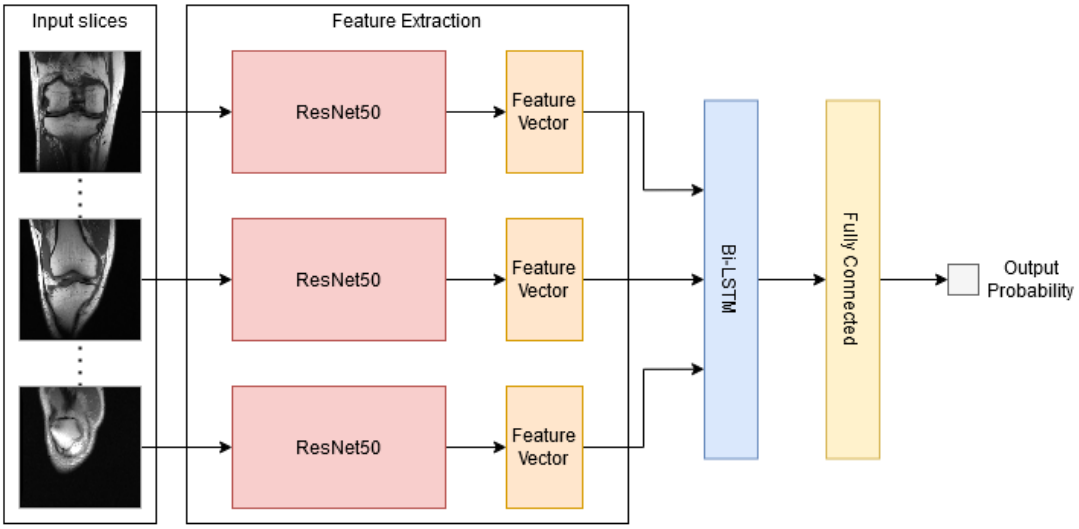


Figure 3.1: Overview of the proposed CNN-LSTM architecture.

This chapter describes the approach used to address the task of classifying 3D multi-planar medical images using deep learning techniques. Using current state-of-the-methods, the model combines the use of convolutional neural networks and recurrent neural networks. Figure 3.1 illustrates the overview of the approach implemented. Section 3.1 of this chapter, addresses the application of Residual Neural Networks (ResNet) for extracting meaningful features (e.g., edges and limits) of the images fed into the network. Then, Section 3.2 describes in detail the architecture of the model proposed and the LSTM component. In Section 3.3, an alternative model is presented. This model takes the already proposed approach and unifies the single labels into a multi-label neural network. Lastly, Section 3.4 summarizes the present chapter.

3.1 Feature Extraction with Residual Neural Networks

With the increase of computing power availability, deep learning architectures started to become more popular as they proved to be a breakthrough in image classification tasks. To accomplish image recognition and classification tasks, deep models often involve stacking multiple convolutional and pooling layers in a network for producing a feature vector, followed by fully-connected layers that produce a final classification. The evolution of deep learning with convolutional neural networks can be seen from the introduction of LeNet by LeCun et al. [15], with seven layers, to more recent approaches as the VGGNet by Simonyan and Zisserman [6], that had as much as 19 layers. This push in depth resulted in improvements for image processing accuracy. However as network depth increases, it was noted that accuracy gets saturated and then degrades rapidly.

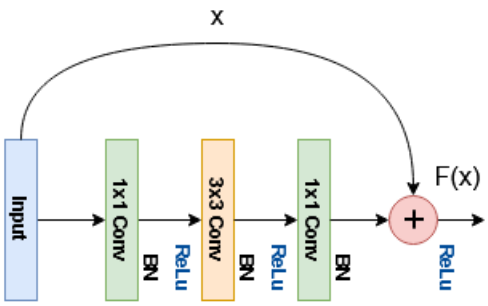


Figure 3.2: Illustration of a Residual Block.

As already introduced in Chapter 2, Residual neural network (ResNet), introduced by He et al. [7], proposes to address the issue of degradation by applying residual blocks. In more detail, ResNet models are based on deep residual learning. Instead of trying to stack layers to fit a desired mapping $H(x)$ directly from x, these layers are designed deliberately to fit a residual mapping $F(x)$. Formally, the stacked non linear layers are made to fit $F(x) = H(x) - x$, recasting the original mapping into $F(x) + x$.

This method of residual learning is adopted in blocks of few stacked layers, also called residual building blocks. In these building blocks, a stack of layers learns the residual mapping $F(x)$, and the operation $F + x$ is performed by a shortcut connection, also called identity mapping, together with an element-wise addition. The identity mapping does not introduce extra parameters nor computation complexity and, in cases were the dimensions of x and F are different, a linear projection can be performed by the shortcut connection to match the dimensions. An example of a residual block can be seen in Figure 3.2, were the stacked layers are three different convolutional layers, and a shortcut connection from the input is added to the output of the last layer in the block, represented as $F(x)$.

In the paper presented by He et al. [7], several models of residual networks were presented and compared to their plain network correspondent. It was clear that ResNet performed better than their plain counter parts, as the 18 and 34 layered residual networks, with short connections every pair of 3×3 filters, trained and evaluated on the ImageNet [9] data set had lower validation error. The ResNet34 (i.e. with 34 layers) fared better than all other architectures, achieving a top-1 error of 25.03%.

In the approach proposed in this dissertation, a 50-layer ResNet model (ResNet50) was leveraged to perform feature extraction on the MRNet data set [1]. This network was chosen to be part of our model, as it performed better than the 34 layered version with almost the same computational cost (3.6×10^9 FLOPs (ResNet34) versus 3.9×10^9 FLOPs (ResNet50)). The ResNet50 model is able to perform better at the same computational power, due to the the bottleneck architecture implemented as it allows to go deeper. Residual blocks present on this architecture are modified to have a bottleneck design, in detail, instead of utilizing a 2 layered residual block as seen in ResNet34, a stack of 3 layers consisting of 1×1, 3×3 and 1×1 convolutions. Bottleneck Residual Block is illustrated in Figure 3.2. ResNet50

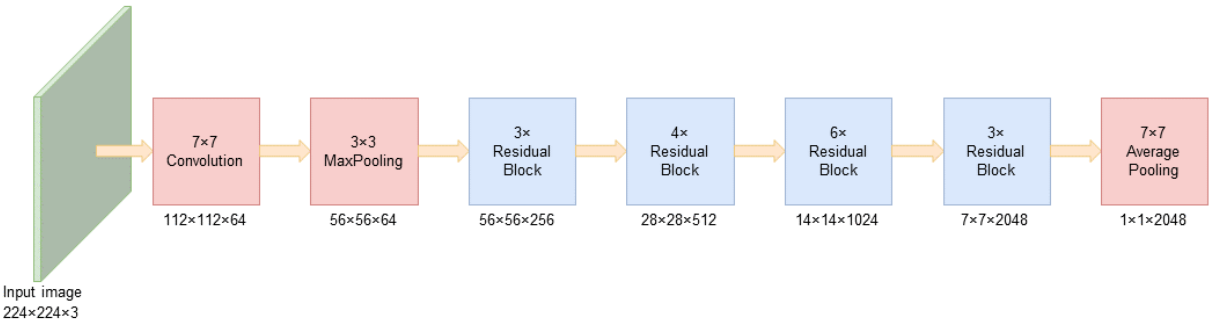


Figure 3.3: Illustration of the ResNet50 architecture.

architecture is shown in Figure 3.3 and consists of a BN-Relu-Conv block (Batch Normalization layer, ReLu, and Convolutional layer block), followed by a max-polling layer, and 4 stages of bottleneck residual blocks, before a final average pooling layer.

3.2 CNN-LSTM Architecture

When dealing with a classification task including 3D images, a simple and direct way of extracting spatial features would be to build a three dimensional CNN. However, when dealing with large sized 3D images (in our case $224 \times 224 \times 24$ voxels) a deeper CNN is required to be able to accurately classify those images. Going deeper means that a bigger number of training samples is needed to achieve good performance, although extensive data sets of medical images, specially 3D images, are not readily available. This creates a problem, as 3D CNNs are no longer a solution to the classification and recognition tasks of medical images, due to their large number of learnable parameters and low number of samples in existing data-sets, that lead to the low performance in these deep models [23].

In this dissertation we propose a new classification model, that is supported by a combination of a 2D CNN and an RNN, that learns the features of 3D knee MR images and classifies them in terms of abnormal exams, anterior cruciate ligament (ACL) and meniscus tears. As 3D images can be interpreted as times series of 2D slices, a method to capture features on 2D images can be used together with another method to extract the correlated features between slices, cooperating as one to learn and acquire the full 3D spatial features and improve image classification.

To better extract features from the 2D slices from the decomposed 3D image, the ResNet50 network was leveraged to produce feature vectors. Pre-trained weights on ImageNet [9] were used to save computational power and quickly identify basic features (e.g., edges), as training a model from scratch generally demands a larger data set than the one used in this work. Using a technique of transfer

learning with pre-trained weights on an existing large benchmark data set could bring a bias to our model, as these weights could have been a result of the training done on images present on the samples being used to learn. Nonetheless, this is not the case as images from the MRNet data set are not present on the ImageNet. As a result, our model needed to train on the data set being fed, so fine-tuning was done and the later layers of the residual model were unfroze, as first-layer features are general and last-layer features are more specific [24]. To obtain the desired feature vector from the input slices, the last fully-connected layer was removed from the original ResNet50 network, so that the last layer was an average pooling layer outputting a vector of $n \times 1 \times 1000$, were *n* is the number of input 2D slices for each 3D image. Time-distributed wrappers were applied to the residual network, as this allows decomposing the three dimensional image as intended into several 2D image slices and apply every layer of the model to those slices.

Following the feature extraction done to the individual slices, an inter-slice extraction of features must be done to fully capture the 3D nature of the original image. Then, in order to weight in spatio-temporal correlations within the order-preserving sequence of slices a recurrent neural network must be applied, as the connections between nodes on a RNN form a directed graph along a temporal sequence. These networks can be applied in our case to obtain inter-slice features, as the time-distributed wrapped residual network outputs sequential data from our 3D input.

To fulfill the main intuition of extracting inter-slices features and correlating features from correlating slices, an RNN type layer was added after the ResNet50 output. A long short term memory layer was chosen, as it is able to model relationships between inputs separated by large sequences of data, by having a recurrent hidden state regulated through gates.

In the model, a bidirectional extension for the LSTM unit (bi-LSTMs) was used to enable the network to learn spatio-temporal correlations of the slices in a forward direction ($\overrightarrow{h_{it}}$), and in a backward direction ($\overleftarrow{h_{it}}$). Both states from the independent LSTM cells are concatenated, $h_{it} = [\overrightarrow{h_{it}}, \overleftarrow{h_{it}}]$, providing a more wide-raging summary of the inter-slice features.

The resulting output from the bidirectional LSTM layer was then combined with a fully-connected layer, before a final layer containing a sigmoid activation function outputs a prediction in the range of $[0, 1]$. This end to end network, that can be seen in Figure 3.1, was entirely trained using the Adam optimizer [10] with a learning rate starting at 0.0001. This small learning rate was chosen due to the impact high learning rates have on pre-trained networks, as we risk losing previous knowledge by distorting the CNN weights too soon and too much. To calculate the error and propagate it via back-propagation, the binary cross-entropy loss was employed as the loss function.

Due to MRNet data set containing different image planes for each same training sample (i.e. different 3D images for the axial, coronal, and sagittal plane of a sample), nine networks were trained in total, one for every plane and classification task. In order to simplify the results obtained, and hopefully improve

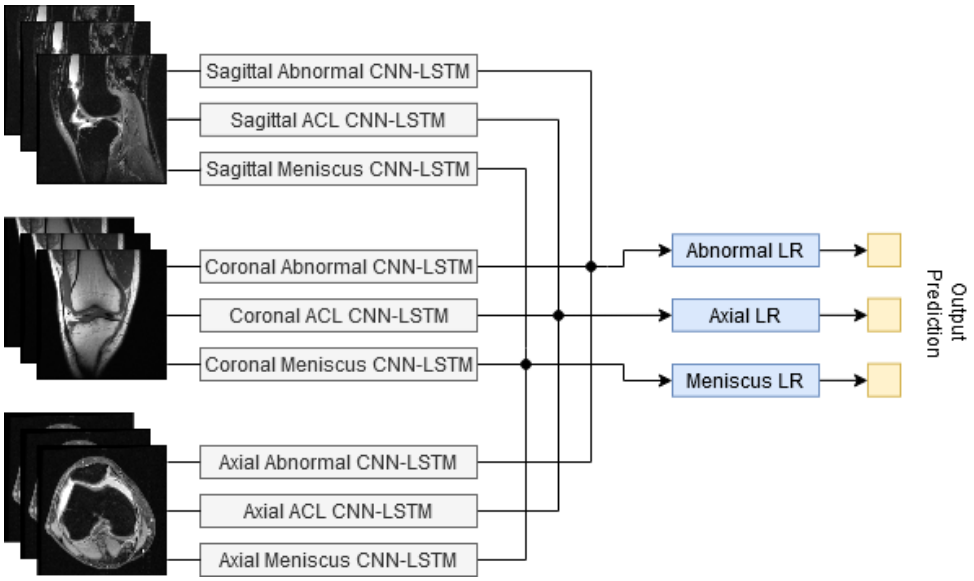


Figure 3.4: Illustration of the combined predictions using Logistic Regression.

the quality of the prediction, a logistic regression was trained to combine the output probabilities of each plane on the same classification task, and generate a single prediction. This method was selected as it allows us to give less weight to a less accurate network, and more weight to a better performing network in the final prediction. To this end, three logistic regression models were trained, one for each classification task (i.e. abnormal exams, ACL tear, and meniscus tear). Figure 3.4 illustrates the method previously explained.

3.3 A Multi-Label Approach

As an alternative to the CNN-LSTM approach presented in the previous section, a multi-label architecture was also proposed with the intention of simplifying and speeding up the process of training the neural networks proposed, as training 3 networks (i.e., one for each plane) is faster than training 9. An overview of this neural network can be seen in Figure 3.5. This model relies heavily on the CNN-LSTM architecture, as it features the same ResNet model pre-trained on ImageNet weights for intra-slice feature extraction, and a LSTM layer to correlate slices and capture inter-slice features.

In more detail, this network has a common branch to all outputs composed of non-trainable layers of the ResNet50, time-distributed as mentioned before, before branching out into the different outputs. Each branch contains the remaining unfrozen layers of ResNet, minus the fully-connected layer, as it was removed. The different branches are able to train over the intended targets. The resulting feature vector is fed onto the branch Bi-LSTM layer, before passing through a fully-connected layer with a sigmoid activation function, to finally output a prediction in the range of $[0, 1]$. It is important to notice that the

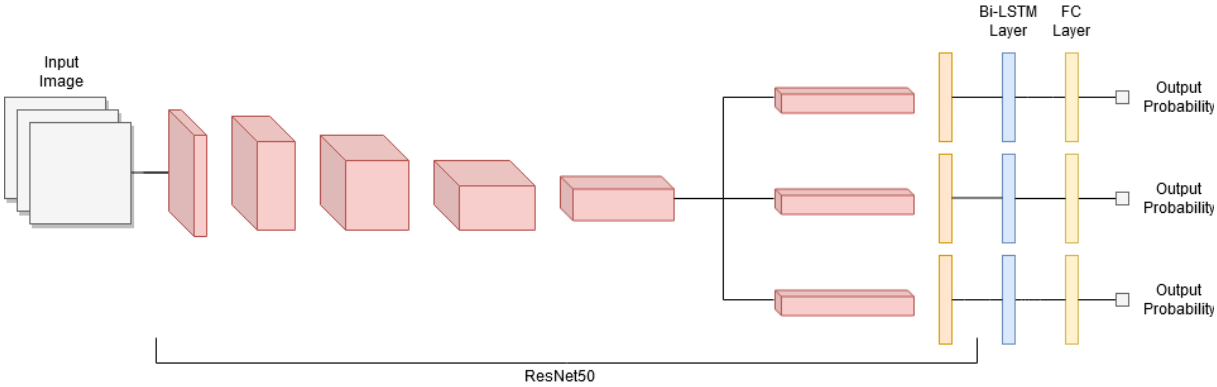


Figure 3.5: Overview of the multi-label approach architecture.

branches do not share parameters between themselves. This approach was also trained using the Adam optimizer, and the same learning rate and loss function as the previous CNN-LSTM model.

To combine the several output probabilities from each plane, three logistic regressions were used in the same manner as the architecture before, producing each a single final prediction.

3.4 Overview

This chapter detailed the architecture used to classify 3D medical images. The model consists in two components joined together in an end to end neural network. One of the components is based on CNNs to extract intra-slice features, and the other based on RNNs to correlate and extract intra-slice features. The convolution neural network used on the main architecture is presented in Section 3.1.

Section 3.2, describes the problems of using a 3D CNN to address this classification task, and details the proposed architecture. It reports the use of the previously presented ResNet50 as feature extractor for 2D images on the neural network, introducing the transfer learning technique with ImageNet weights. A RNN variant, named LSTM, was also used, as well as its bidirectional extension (bi-LSTM). This section further describes the way the model was trained, and the solution used to combine the different prediction probabilities from the different plane networks.

The last section, Section 3.5, reports on an alternative approach, characterized by being a multi-label network. The rest of the section proceeds to describe the model and the way it was trained.

4

Experiments and Discussion

Contents

Table 4.1: Statistical characterization of the data set used in the experiments.

Statistics	Training	Test
Exams with abnormality (%)	913 (80.80)	95 (79.17)
Exams with ACL tear (%)	208 (18.41)	54 (45.00)
Exams with meniscal tear (%)	397 (35.13)	52 (43.33)
Total number of exams	1,130	120

This Chapter presents the experimental evaluation of the proposed deep learning architecture. Section 4.1 describes the data-set and its characteristics, together with the pre-processing and augmentation methods utilized. Section 4.2 presents the obtained results obtained. The first experiment details the results over the MRNet data-set with base models, while the second experiment presents the obtained results with the proposed models on the knee MRI classification task. Lastly, Section 4.3 presents a summary of the chapter, providing an overview of the obtained results.

4.1 Data Set Analysis and Experimental Methodology

As previously mentioned, with increasingly more databases of medical images available, machine learning methods can now leverage this data, with the objective of improving workflows and playing a crucial role in assisting clinicians. An example can be the data set used in this work. MRNet [1] provides knee MRI exams performed at the Stanford University Medical Center between January 1, 2001, and December 31, 2012, that were manually reviewed in order to build a data set of 1,250 knee MRI examinations.

The data set contains 1,008 (80.64%) abnormal exams, with 262 (20.96%) anterior cruciate ligament (ACL) tears and 449 (35.92%) meniscal tears. ACL tears and meniscal tears occurred concurrently in 156 (12.48%) exams. Examinations were performed with a standard knee MRI coil and a routine non-contrast knee MRI protocol. From each exam, sagittal plane T2-weighted series, coronal plane T1-weighted series, and axial plane PD-weighted series were extracted to constitute the data set. The number of images in these series ranged from 17 to 61 (mean 31.48, SD 7.97). The exams are split into a training set (1,130 exams from 1,088 patients) and a validation set (120 exams from 113 patients). These figures can be seen in close detail on Table 4.1.

Taking into account the range of the number of slices in each image, pre-processing the data set was needed, as typical neural networks only allow a fixed size of data to be fed into. The original images from the data set have a $n \times 256 \times 256$ size voxel, were *n* is the number of slices present in each image. To fix the number of slices, interpolation was applied to every sample to resize the number of sequences to 24. This number of slices was chosen due to its closeness to the mean average, and reduced size while still maintaining relevant information. Image size was also altered to fit the ResNet with pre-trained

weights on ImageNet, as these weights were trained on 224×224 images, so a re-scaling was done from 256×256.

A data augmentation strategy was also utilized to help reduce over-fitting as the original data set is of a small size. The employed technique is based on the method presented by Hendricks et al. [25] called AugMix. This data augmentation scheme improves on previous techniques by mixing together the results of several augmentation chains in convex combinations, avoiding aggressive augmentation methods and augmentation primitives in chain that can lead to quick image degradation. This technique was applied to our data set in a limited fashion, as medical images should not consider some of the transformations used in the original paper, and each transformation has to be replicated to all slices in a sample. Small gamma variations, rotations and translations were performed in the data-set which allowed us to double the number of training samples.

Model training was done with batches of 32 instances, using the Adam optimizer [10] with a learning rate starting at 0.0001, that could be redefined if validation loss had stopped improving employing a technique available on the Keras library called *ReduceLROnPlateau*. The number of epochs was also defined through a criteria based on a validation loss, stopping when that metric had not improved in 8 epochs. To assess the quality of the predictions, the following metrics were used: accuracy, macro precision, macro recall, and area under the ROC curve (AUC).

4.2 Experimental Results

This Section presents the results for the experiments developed in this dissertation. The first experiment compares different state-of-the-art CNN architectures for the classification of images in the MRNet data set. Results for the proposed architectures of the knee MRI classification task are presented and compared with the original MRNetwork [1] in the second experiment.

4.2.1 Base Model Assessment

To choose which convolutional neural network would better fit our needs as a feature extractor for intra-slices, three different architectures were tested on the knee MRI classification task. These models include VGG16, ResNet50, and DenseNet201. All were trained in the MRNet data set as out-of-the-box, architectures pre-trained with ImageNet weights. To allow the use of these models with 3D images, time-distributed wrappers were used, and a global average pooling layer was added to the end, in order to to group the slices together. The predictions were obtained through a sigmoid activation layer.

Table 4.2 presents the results obtained across all models on the MRNet data set, were bold values represent the best result for each label. In the first set of experimental results, it is possible to verify that the ResNet50 architecture, with pre-trained weights on ImageNet, outperformed the other models

Table 4.2: Comparison between three deep learning algorithms for image classification.

Model	Axial				Coronal				Sagittal			
	Accuracy	Precision	Recall	AUC	Accuracy	Precision	Recall	AUC	Accuracy	Precision	Recall	AUC
VGG16												
Abnormal	0.825	0.870	0.916	0.832	0.850	0.897	0.916	0.739	0.775	0.895	0.811	0.829
ACL	0.775	0.765	0.722	0.825	0.625	0.585	0.574	0.697	0.700	0.514	0.556	0.787
Meniscus	0.650	0.566	0.827	**0.796**	0.675	0.603	0.731	0.751	**0.683**	**0.630**	0.654	0.731
ResNet50												
Abnormal	**0.867**	0.869	**0.979**	0.890	0.792	0.872	0.863	0.757	0.833	**0.912**	0.874	**0.895**
ACL	**0.808**	**0.830**	0.722	**0.849**	0.717	0.700	0.648	0.806	0.633	0.593	0.593	0.674
Meniscus	0.608	0.532	0.808	0.667	0.617	0.550	0.635	0.689	0.625	0.554	0.692	0.722
DenseNet201												
Abnormal	0.808	0.909	0.802	0.820	0.758	0.859	0.832	0.666	0.858	0.906	0.916	0.846
ACL	0.633	0.561	**0.852**	0.650	0.550	0.500	0.019	0.544	0.758	0.805	0.611	0.767
Meniscus	0.592	0.520	0.750	0.662	0.675	0.594	0.789	0.680	0.658	0.571	**0.846**	0.732

on abnormality and ACL tear classification, while DenseNet201 seemed to be the worse model except on the sagittal plane were it outperform the rest of the architectures. The VGG16 model managed to be perform better on the meniscal tear task on the sagittal plane. Globally, these models with small alterations were able to achieve good results. However, the most stable architecture in all labels and planes was ResNet50, as it performed reliably better than the other methods, accomplishing an AUC of 0.895 in the abnormal classification task with sagittal plane images. This result was expected, as VGG16 is a shallower network than ResNet50, and DenseNet201 is a very deep network and thus had learning limitations due to the small data set.

4.2.2 Knee MRI Classification Task

Table 4.3 presents results obtained for the knee MRI classification task in the MRNet data set, with the proposed approaches in this work, namely the CNN-LSTM network and the Multi-label CNN-LSTM model. This set of experimental results was obtained after the predicted probabilities of the models for the different planes were combined using logistic regression. The most beneficial series, determined from the coefficients of the fitted logistic regression, were the sagittal plane for abnormalities, the axial plane for ACL tears and the coronal plane for meniscal tears, for the CNN-LSTM architecture. For the alternative model the most beneficial series were the axial plane for abnormalities and ACL tears and the coronal plane for meniscus tears.

The CNN-LSTM network managed to achieve better results than those achieved by the Multi-label Model in all categories. The main architecture attained a top accuracy of 0.908 on the abnormal task, and a top AUC of 0.870. Results for the multi-label model managed to be acceptable as well, as it performed better than the better base model in Section 4.2.1. However, it was understandable that the simpler model would fare better than the more complex, as more parameters with a small data set can lead to degradation of results.

Table 4.3: Comparison between our CNN-LSTM model, the alternative multi-label approach, and the MRNet model.

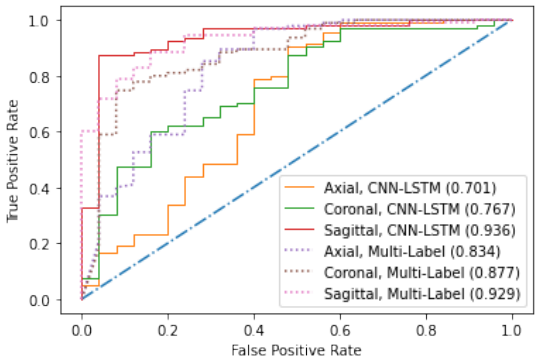
Model		Accuracy	Precision	Recall	AUC
CNN-LSTM + Logistic Regression	Abnormal	0.908	0.906	0.908	0.839
	ACL	0.875	0.877	0.875	0.870
	Meniscus	0.700	0.712	0.701	0.706
Multi-Label Model + Logistic Regression	Abnormal	0.858	0.880	0.858	0.660
	ACL	0.842	0.857	0.842	0.831
	Meniscus	0.700	0.706	0.701	0.701
MRNet [1]	Abnormal	0.850	-	-	0.937
	ACL	0.867	-	-	0.965
	Meniscus	0.725	-	-	0.847
Unassisted general radiologist [1]	Abnormal	0.894	-	-	-
	ACL	0.920	-	-	-
	Meniscus	0.849	-	-	-

When comparing our best results, obtained with the CNN-LSTM approach, to the original implementation of MRNet on the same data set by Bien et al. [1], we verify that for the AUC metric, the MRNet model performed better in abnormality tear detection, ACL tear detection, and meniscal tear detection, with AUC values of 0.937, 0.965 and 0.847, respectively. However, in the accuracy metric, our model performed better in abnormality detection and ACL tear detection when compared against MRNet (which obtained 0.850 and 0.867, respectively, in that metric). Additionally, when compared to results obtained by unassisted general radiologists in abnormality detection, provided in the MRNet paper, both of our models have no significant differences in the performance metrics.

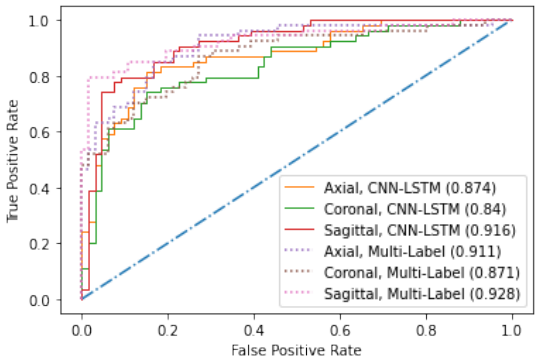
Figure 4.1 illustrates the AUC values for each class and plane for the two proposed models. In Figure 4.1(a), AUC results are presented for the abnormal classification tasks, were we can verify that the better performing network is CNN-LSTM in the sagittal plane, with an AUC of 0.936. This explains why the sagittal plane was the most beneficial series when classifying abnormalities, as it performed much better than the other planes. When looking at Figure 4.1(b), it is observable that the AUC values are very similar between the models. The same can be said with Figure 4.1(c), that illustrates the AUC values of the meniscal tear classification task.

4.3 Overview

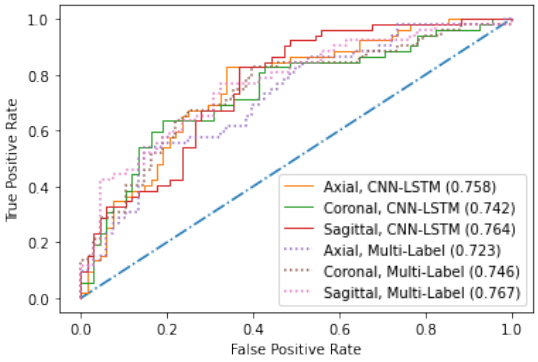
This chapter presented an evaluation of the performance and results obtained on the experimental tests made on the proposed deep learning architectures. Section 4.1 presented the data set used in this dissertation, and the pre-processing steps taken, as well as the experimental methodology. The following section 4.2, provided 2 experimental sets. The first one in subsection 4.2.1 compared three deep learning networks with minimal changes, trained on the MRNet, were the Residual Network had the best performance, with a top AUC of 0.895 on abnormality detection. Lastly, in subsection 4.2.2



(a) Illustration of the abnormal exam ROC curves.



(b) Illustration of the ACL tear ROC curves.



(c) Illustration of the meniscus tear ROC curves.

Figure 4.1: Comparison of ROC curves on the classification tasks of abnormal exams, ACL and meniscus tear.

our proposed approaches to the MRNet classification problem, were compared against each other, and the CNN-LSTM model proved to be better, achieving a value of 0.870 in the AUC metric on ACL tear detection, and 0.903 in abnormality detection accuracy. A comparison with the original MRNet model was also made and was observed that in accuracy metrics our model was superior, but failed to surpass the AUC metrics obtained by Bien et al. [1].

5

Conclusion

Contents

This chapter presents a summary of the main conclusions and contributions achieved on this dissertation and points to future work to be developed in order to better approach the problem of 3D medical image classification using deep learning.

5.1 Summary of Main Conclusions

This dissertation presented an approach based on CNNs and RNNs, to adress the task of classifying multi planar 3D medical images, specifically knee MRIs according to 3 classes. Considering the studies reported in the field of deep learning towards the task of classifying 3D medical images, a novel neural network was introduced. The architecture starts with a convolutional neural network trained to extract features from intra-slices fed from input 3D images, followed by a recurrent neural network to correlate and extract inter-slice features, outputting a prediction in a end-to-end network.

In this approach, to understand which architecture would better fit our model and extract features from 2D slices, three convolutional neural networks were experimented: Visual Geometry Group network (VGGNet) [6], Residual Neural Networks (ResNet) [7] and Densely Connected Convolutional Networks (DenseNet), all pre-trained in the ImageNet data set [9], were ResNet50 performed better than the other two models, achieving an AUROC of 0.895 in the abnormal classification task with sagittal plane image.

For the full proposed model, adding a Long Short Term Memory (LSTM) layer after the CNN feature extraction contributed to a better performing model, and added the capability of modeling relationships between slices. Logistic regression to combine multiple predictions also contributed for the good performance of the model, as less accurate prediction weighted less on the final output.

Although the multi-label architecture had a worst performance than the main approach due to the reasons already stated, this model is still validated as it was better than the ”vanilla” approach, and had no significant differences against unassisted radiologists in abnormality detection.

In brief, the results obtained from the proposed model reveal that a combination of CNNs and RNNs is a possibility for 3D medical image classification, and that it can be used to assist in clinicians diagnosis and improve workflow.

5.2 Future Work

Despite the results obtained, there is room for improvement. As future work, I believe that combining the several planes would lead to performance improvement and better computational cost. Onishi et al. [26] propose a multiplanar neural network with Deep Convolutional Neural Network (DCNN) and generative adversarial networks (GAN) to detect and classify pulmonary nodules. The study demonstrated that a multi planar classifier improved over a single cross section design.

Regarding the CNN architecture implemented, more recent deep learning networks can be used to improve upon our results. Inception-ResNet introduced by Szegedy et al. [19] is an example of a state-of-the-art CNN that can be utilized to accelerate training at relatively low computational costs. Future work can also consider implementing a more recent deep learning network for feature extraction such as EfficientNet [27]

Concerning the RNN proposed in the architecture, instead of reducing the feature mapping to a single feature vector to introduce to the LSTM layer, a possible alternative is to use ConvLSTM units. Shi et al. [28] presented those units arguing that it leads to a better capture of spatiotemporal correlations.

Bibliography

[1] N. Bien, P. Rajpurkar, R. L. Ball, J. Irvin, A. Park, E. Jones, M. Bereket, B. N. Patel, K. W. Yeom, K. Shpanskaya, S. Halabi, E. Zucker, G. Fanton, D. F. Amanatullah, C. F. Beaulieu, G. M. Riley, R. J. Stewart, F. G. Blankenberg, D. B. Larson, R. H. Jones, C. P. Langlotz, A. Y. Ng, and M. P. Lungren, "Deep-learning-assisted diagnosis for knee magnetic resonance imaging: Development and retrospective validation of MRNet," *PLoS Medicine*, vol. 15, no. 11, pp. 1–19, 2018.

[2] P. Rajpurkar, J. Irvin, R. L. Ball, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. P. Langlotz, B. N. Patel, K. W. Yeom, K. Shpanskaya, F. G. Blankenberg, J. Seekins, T. J. Amrhein, D. A. Mong, S. S. Halabi, E. J. Zucker, A. Y. Ng, and M. P. Lungren, "Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists," *PLoS Medicine*, vol. 15, no. 11, pp. 1–17, 2018.

[3] R. Golan, C. Jacob, and J. Denzinger, "Lung nodule detection in ct images using deep convolutional neural networks," in *Proceeding of the 2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 243–250.

[4] H. S. Parmar, B. Nutter, R. Long, S. Antani, and S. Mitra, "Deep learning of volumetric 3D CNN for fMRI in Alzheimer's disease classification," in *Medical Imaging 2020: Biomedical Applications in Molecular, Structural, and Functional Imaging*, vol. 11317, International Society for Optics and Photonics. SPIE, 2020, pp. 66 – 71.

[5] M. Liu, D. Cheng, and W. Yan, "Classification of Alzheimer's Disease by Combination of Convolutional and Recurrent Neural Networks Using FDG-PET Images," *Frontiers in Neuroinformatics*, vol. 12, 06 2018.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the 3rd International Conference on Learning Representations*, 2015, pp. 1–14.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.

[8] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, 2017, pp. 2261–2269.

[9] Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 248–255.

[10] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” in *Proceedings of the 3rd International Conference on Learning Representations*, 2015, pp. 1–15.

[11] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning Representations by Back-Propagating Errors*. Cambridge, MA, USA: MIT Press, 1988, p. 696–699.

[12] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines.” in *ICML*. Omnipress, 2010, pp. 807–814.

[13] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder–decoder for statistical machine translation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Oct. 2014, pp. 1724–1734.

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June-2015. IEEE, 2015, pp. 1–9.

[19] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, 2016, pp. 2818–2826.

[20] A. A. Novikov, D. Major, M. Wimmer, D. Lenis, and K. Buhler, “Deep sequential segmentation of organs in volumetric medical scans,” *IEEE Transactions on Medical Imaging*, vol. 38, no. 5, pp. 1207–1215, 2019.

[21] S. Korolev, A. Safiullin, M. Belyaev, and Y. Dodonova, “Residual and plain convolutional neural networks for 3D brain MRI classification,” in *Proceedings of the International Symposium on Biomedical Imaging*, 2017, pp. 835–838.

[22] F. Liu, Z. Zhou, A. Samsonov, D. Blankenbaker, W. Larison, A. Kanarek, K. Lian, S. Kambhampati, and R. Kijowski, “Deep learning approach for evaluating knee MR images: Achieving high diagnostic performance for cartilage lesion detection,” *Radiology*, vol. 289, no. 1, pp. 160–169, 2018.

[23] D. Shen, G. Wu, and H.-I. Suk, “Deep learning in medical image analysis,” *Annual Review of Biomedical Engineering*, vol. 19, no. 1, pp. 221–248, 2017.

[24] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *Advances in Neural Information Processing Systems*, vol. 27. Curran Associates, Inc., 2014, pp. 3320–3328.

[25] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, “AugMix: A simple data processing method to improve robustness and uncertainty,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[26] Y. Onishi, A. Teramoto, M. Tsujimoto, T. Tsukamoto, K. Saito, H. Toyama, K. Imaizumi, and H. Fujita, “Multiplanar analysis for pulmonary nodule classification in CT images using deep convolutional neural network and generative adversarial networks,” in *Proceedings of the International Journal of Computer Assisted Radiology and Surgery*, vol. 15, 11 2019.

[27] M. Tan and Q. V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” *36th International Conference on Machine Learning, ICML 2019*, vol. 2019-June, pp. 10 691–10 700, 2019.

[28] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo, “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting,” in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS’15. Cambridge, MA, USA: MIT Press, 2015, p. 802–810.