U.PORTO

FEP FACULDADE DE ECONOMIA UNIVERSIDADE DO PORTO

CLASSIFICATION OF PULMONARY NODULES IN 2-[^{18}F]FDG PET/CT IMAGES WITH CONVOLUTIONAL NEURAL NETWORKS

Victor Manuel Alves

Dissertation

Master in Modeling, Data Analysis and Decision Support Systems

Supervised by

Professor Doutor João Manuel Portela da Gama

Professor Doutor Jaime dos Santos Cardoso

2021

To my parents for being a permanent source of inspiration.

To my wife for all the unlived moments due this project.

Biographical Sketch

Victor Manuel Alves holds a master degree in Medicine (2010) from the NOVA Medical School of Lisbon. In 2016, he obtained the degree of Nuclear Medicine specialist. Since then, he is Nuclear Medicine physician at Centro Hospitalar Universitário de São João.

Acknowledgements

I would like to thank my advisor, Prof. Doutor João Gama, and my co-advisor, Prof. Doutor Jaime Cardoso, for their support, their suggestions and for reviewing my Dissertation.

I would like to thank my hospital, Centro Hospitalar Universitário de São João (CHUSJ), for having acknowledged the scientific merit of the project and having granted me the necessary authorizations and give me access the data. This includes, of course, a thanks to my Head of Department, Dr. Jorge Gonçalves Pereira.

I would like to express my sincere gratitude to Dr.ª Fernanda Gonçalves, Head of Archives at CHUSJ, and to her team, for the expeditious way in which they made me available the imaging exams from the physical archive.

I would also like to thank Dr. Luis Hugo Duarte, Head of Nuclear Medicine Department at IPO Porto, and Dr. José Manuel Oliveira, Coordinator of Nuclear Medicine at Lenitudes and at HPP-Medicina Molecular, for kindly providing some missing imaging exams that had been performed in their centres.

Abstract

Early diagnosis is the principal predictor of survival in lung cancer. Pulmonary nodules are a common incidental finding in radiological scans. Although mostly benign, a small proportion represents early-stage lung cancer, being a diagnostic challenge. 2-[^{18}F]FDG PET/CT is useful for further characterization of pulmonary nodules and prevent an invasive diagnostic procedure if the exam is negative. Deep learning algorithms have potential for improving the diagnosis of pulmonary nodules and, in this way, reducing the proportion of patients that requires an invasive procedure as well as reducing the false omission rate.

The main objectives of this research were to create an annotated database of pulmonary nodules that included 2-[^{18}F]FDG PET images, and develop a Convolutional Neural Network model for binary classification of indeterminate solid pulmonary nodules.

A total of 113 participants met the eligibility criteria. The image data pre-processing included coregistration, spatial resampling, manual detection of the nodules and cropping a cubic region of interest. A final model was selected from a set of candidate models previously trained, optimized and evaluated by 4-fold-cross-validation. That model was subsequently assessed in a test set. Models of three types of 3D Convolutional Neural Networks architectures were trained from random weight initialization (Stacked 3D CNN, VGG-like and Inception-v2-like models) both in original and augmented datasets. Transfer learning, from ImageNet with Resnet50, was also used.

The final model (Stacked 3D CNN model) obtained an area under the ROC curve of 0.8385 (95% CI: 0.6455 - 1.0000) in the test set. The model had a sensibility of 80.00%, a specificity of 69.23% and an accuracy of 73.91%, in the test set, for an optimized decision threshold derived from the cross-validation that assigns a higher cost to false negatives.

In conclusion, a 3D Convolutional Neural Network model was successfully developed. It was relatively effective at distinguishing benign from malignant pulmonary nodules in 2-[^{18}F]FDG PET images.

Keywords: Deep learning, Convolutional Neural Networks, Medical Imaging, Positron Emission Tomography, Pulmonary Nodules.

Resumo

O diagnóstico precoce é o principal preditor de sobrevivência no cancro de pulmão. Os nódulos pulmonares são um achado incidental comum em exames radiológicos. Embora, a maior parte seja benigna, uma pequena proporção representa cancro do pulmão em estadio inicial, tornando-se um desafio diagnóstico. A 2-[^{18}F]FDG PET/CT é útil para caraterização adicional de nódulos pulmonares e evita um procedimento diagnóstico invasivo, se o exame for negativo. Algoritmos de *deep learning* têm potencial para melhorar o diagnóstico de nódulos pulmonares e, dessa forma, reduzir a proporção de doentes que requer um procedimento diagnóstico invasivo, bem como reduzir a taxa de falsas omissões.

Os principais objetivos desta investigação foram criar uma base de dados anotada de nódulos pulmonares que incluísse imagens de 2-[^{18}F]FDG PET, e desenvolver um modelo de Rede Neuronal Convolucional para classificação binária de nódulos pulmonares sólidos indeterminados.

Um total de 113 doentes preencheram os critérios de elegibilidade. O pré-processamento dos dados de imagem incluiu co-registo, reamostragem espacial, deteção manual dos nódulos e recorte de uma região cúbica de interesse. Um modelo final foi selecionado a partir de um conjunto de modelos candidatos previamente treinados, otimizados e avaliados por validação cruzada com 4 *folds*. Esse modelo foi posteriormente avaliado num conjunto de teste. Modelos de três tipos de arquiteturas de Redes Neuronais Convolucionais 3D, foram treinados a partir de inicialização aleatória de pesos (modelos Stacked 3D CNN, VGG-like e Inception-v2-like), tanto no conjunto de dados original, como num conjunto aumentado. Transferência de aprendizagem, a partir do ImageNet com Resnet50, foi também usada.

O modelo final (modelo Stacked 3D CNN) obteve uma área sob a curva ROC de 0,8385 (IC 95%: 0,6455 - 1,0000) no conjunto de teste. O modelo teve sensibilidade de 80,00%, especificidade de 69,23% e exactidão de 73,91%, no conjunto de teste, para um limiar de decisão otimizado derivado da validação cruzada que atribui um custo mais alto aos falsos negativos.

Em conclusão, um modelo de Rede Neuronal Convolucional 3D foi desenvolvido com sucesso, tendo sido relativamente eficaz em distinguir nódulos pulmonares benignos de malignos em imagens de 2-[^{18}F]FDG PET.

Keywords: Deep learning, Redes Neuronais Convolucionais, Imagem Médica, Tomografia de Emissão de Positrões, Nódulos Pulmonares.

Contents

List of Figures

List of Tables

List of Abbreviations

2-[^{18}F]FDG 2-deoxy-2-[^{18}F]fluoro-D-glucose.

2D Two-dimensional.

3D Three-dimensional.

CNN Convolutional Neural Network.

CT X-ray Computed Tomography.

DICOM Digital Imaging and Communications in Medicine.

PET Positron Emission Tomography.

PET/CT Positron Emission Tomography/ Low dose Computed Tomography.

ROC Receiver Operating Characteristic.

SUL standardized uptake value normalized by the lean body mass.

SUV standardized uptake value normalized by the body mass.

SUVmax maximum standardized uptake value normalized by the body mass.

Chapter 1

Introduction

1.1 Motivation

Pulmonary nodules are a common incidental finding in imaging scans performed for various indications, including lung cancer screening (Elia et al., 2019). Although most of them are benign, a small proportion represents early-stage lung cancer (Elia et al., 2019).

The early diagnosis of lung cancer remains essential in reducing mortality of the leading cause of cancer death worldwide, despite therapeutic progress in recent years (Woodard et al., 2016; Sung et al., 2021). The broad overlapping of the attenuation and morphological features between early-stage lung cancer and some benign lesions in radiological imaging scans, such as the X-ray Computed Tomography (CT), represents a challenge of image interpretation for physicians (Elia et al., 2019).

2-deoxy-2-[^{18}F]fluoro-D-glucose (2-[^{18}F]FDG) Positron Emission Tomography/ Low dose Computed Tomography (PET/CT) has become an indispensable tool in the evaluation of pulmonary nodules, outperforming other imaging modalities (Callister et al., 2015). Nevertheless, it has well-known causes of false positives, which may lead to unnecessary interventions, and false negatives that preclude its application in certain lesions (Callister et al., 2015).

2-[^{18}F]FDG PET/CT interpretation by the physicians in the evaluation of pulmonary nodules fundamentally lies in the intensity of 2-[^{18}F]FDG uptake by the nodules, either qualitatively interpreting the images or extracting a quantitative feature, the maximum standardized uptake value normalized by the body mass (SUVmax) of the lesion (Callister et al., 2015).

There is potential for application of advanced image analysis techniques capable of extracting and analyzing more information from the PET/CT images and possibly integrating it with other relevant clinical data in order to reduce the uncertainty associated with the decision making process (Castiglioni et al., 2019).

Artificial Intelligence has gained popularity in recent years in many fields, including in Med-

ical Imaging, due to the huge availability of data, improved computing power and advances in algorithm development (Esteva et al., 2019). Deep learning algorithms have reached comparable performance to physicians or even outperformed them in specific tasks, in areas as diverse as dermatology (Esteva et al., 2017), ophthalmology (Kermany et al., 2018), pathological anatomy (Ehteshami Bejnordi et al., 2017) or radiology (Rajpurkar et al., 2018).

The application of deep learning to Nuclear Medicine imaging is in its early days. No study about classification of indeterminate pulmonary nodules was published. The development of machine learning-based decision support systems, namely with deep learning, has potential, among other things, to allow fast, reproducible and effective decision making, especially in complex cases, in an age where Nuclear Medicine physicians have to analyse large information volumes in order to make decisions (Y. Yang et al., 2018).

1.2 Definition of the Problem

A pulmonary nodule is defined as a rounded opacity with a maximum diameter of 3 cm, mostly surrounded by aerated lung, including contact with pleura (Callister et al., 2015). When it is not associated with atelectasis, lymphadenopathy or postobstructive pneumonia, it is called a solitary pulmonary nodule (Patel et al., 2013). It can be classified as solid or subsolid nodule, according to their morphology (Callister et al., 2015). The subsolid nodules are subdivided in ground-glass and partially solid nodules (Callister et al., 2015).

CT is the first step for evaluation of pulmonary nodules, despite its relatively low specificity due to the partial overlap of morphological characteristics between malignant neoplasms and lesions of another nature, which in a population with low cancer prevalence causes a low positive predictive value (Callister et al., 2015; MacMahon et al., 2017).

2-[^{18}F]FDG PET/CT has a higher specificity than CT, but has lower spatial resolution and is well-suited for the differential diagnosis of solid or partially solid pulmonary nodules, with a mean diameter greater than 8-10 mm in the previous CT (Callister et al., 2015; MacMahon et al., 2017). In case of the partially solid nodules, this dimension is for the solid component (Callister et al., 2015; MacMahon et al., 2017). An example of a 2-[^{18}F]FDG PET/CT exam is in the figure 1.1.

A higher specificity of the 2-[^{18}F]FDG PET/CT than the CT and the sequential diagnostic approach results in a higher positive predictive value, which reduces the proportion of the unnecessary invasive diagnostic procedures (Callister et al., 2015; MacMahon et al., 2017). This is relevant because biopsies can may be a source of complications such as pneumothorax and hemorrhage (Elia et al., 2019). Still, 2-[^{18}F]FDG PET/CT is not free of false positives, being inflammatory/infectious pathology a well-known cause of those (Ruilong et al., 2017). Nod-

ules of small size ($\leq$ 8 mm) as well as ground-glass nodules or partially solid nodules with small solid component which represent *in situ* or minimally invasive adenocarcinoma, respectively, are known causes of false negatives and preclude the use of 2-[^{18}F]FDG PET/CT in those situations (Callister et al., 2015). Others potential causes of false negatives are well-differentiated invasive adenocarcinomas and carcinoid tumors (Reginelli et al., 2019).

A meta-analysis performed by Ruilong et al. (2017) showed a sensitivity of 82% (95% CI: 76%-87%), specificity of 81% (95% CI: 66%-90%) and area under the Receiver Operating Characteristic (ROC) curve of 0.87 (95% CI: 0.84-0.90) for the 2-[^{18}F]FDG PET/CT in this task.

PET/CT is a hybrid tomographic imaging technique. Positron Emission Tomography (PET) measures the distribution of a positron (β^+)-emitting tracer in the body. In this kind of radioactive decay, two 511 keV gamma rays are emitted almost exactly 180° apart from each annihilation event between a positron and an electron of the matter, in this case, the patient body (Bailay et al., 2015). Whenever two 511 keV photons reach the circular ring of detector of the scanner in a time window of nanoseconds, it is assumed that they come from the same annihilation event (true coincidence) (Bailay et al., 2015). The exact spatial point where the a decay occurred is unknown and only can be assumed that it occurred in a line between the two points of the detector system (line of response) (Bailay et al., 2015). The set of lines of response and their angular projections obtained for every decay events during of a scanning period are stored in multidimensional arrays known as sinograms (Bailay et al., 2015). These PET data suffer several corrections before reconstruction in order to eliminate image artifacts and quantitative errors: normalization (understood here as compensation of variations in the sensitivity of different lines of response), random coincidence correction, attenuation correction, scatter correction, dead time correction (Bailay et al., 2015). The data acquired are reconstructed in Three-dimensional (3D) images most commonly using an iterative algorithm, the Ordered Subset Expectation Maximization (Bailay et al., 2015). The reconstructed images store counts of decay per voxel. In order to be compared to other related data they are converted in units of absolute activity concentration such as becquerels per millilitre (Bailay et al., 2015). The 3D images are saved as a sequence of adjacent 2D image transaxial slices through the patient body in Digital Imaging and Communications in Medicine (DICOM) files (Bailay et al., 2015). Orthogonal or obliques slices can be calculated by rearranging the pixel matrix (Bailay et al., 2015). Finally, the pixel intensity can be transformed in a color space by a display system, in order to produce an interpretable image for physicians (Bailay et al., 2015).

2-[^{18}F]FDG PET/CT measures the body distribution of 2-[^{18}F]FDG, an analog of glucose (Bailay et al., 2015). So, an increased tracer uptake is observed in normal or abnormal tissues with a high activity of the glycolytic metabolic pathways (E. Lin & Alavi, 2009). Tumors typically grow faster than normal tissues and use inefficiently the glucose, so mostly of them are

2-[^{18}F]FDG-avid (E. Lin & Alavi, 2009). The tracer concentration (C_i) depends not only of the tissue, but also of the injected activity (A) and the tracer distribution volume (Bailay et al., 2015). The patient weight (W) is a surrogate measure of the distribution volume (Bailay et al., 2015). So it was created a measure that considers these two factors, the standardized uptake value normalized by the body mass (SUV) which is the widely used measure in PET imaging (Bailay et al., 2015):

$$SUV(g/mL) = \frac{C_i(kBq/mL)}{A(kBq)/W(g)} \tag{1.1}$$

It is assumed that the density of tissue is equivalent to 1.0 g/mL, such that the units effectively cancel and the SUV becomes a dimensionless measure (Bailay et al., 2015). CT measures the tissue attenuation and provides attenuation and morphological information of the tissues also as multiple image slices of the body (Bailay et al., 2015). A low dose protocol is applied in this context. CT is spatially registered with PET and used for attenuation correction and anatomical location of PET images (Bailay et al., 2015).

Despite the advances in PET/CT instrumentation have resulted on improvement in image quality, is possible that the information may be being explored in a limited extent in most of publications and especially in clinical practice. Tracer uptake changes are usually visually detected and qualitatively characterized, and the intensity subsequently quantitatively measured. The same occurs in pulmonary nodules (Ruilong et al., 2017). Despite the success of this approach in numerous publications, in both diagnosis and prognosis, there are situations in which its discriminant power remains limited (Hatt et al., 2019). In pulmonary nodules, the integration of PET/CT information with the remaining clinical risk factors is usually performed through the clinical judgment (MacMahon et al., 2017). Scarce attempts to develop machine learning-based predictive models using more information extracted from images and, possibly, other relevant clinical features have been reported, but these are restricted to classical methods (Herder et al., 2005; Y. Yang et al., 2018; S. Chen et al., 2019; Teramoto et al., 2019; H.-Y. Guo et al., 2020).

Limitations of current methods in the diagnosis of lung cancer may be an opportunity for deep learning, namely for reducing the rate of false positives without reducing sensitivity (Y. Yang et al., 2018).

1.3 Objectives

1.3.1 Primary objectives

The primary objectives of the current dissertation are the following:

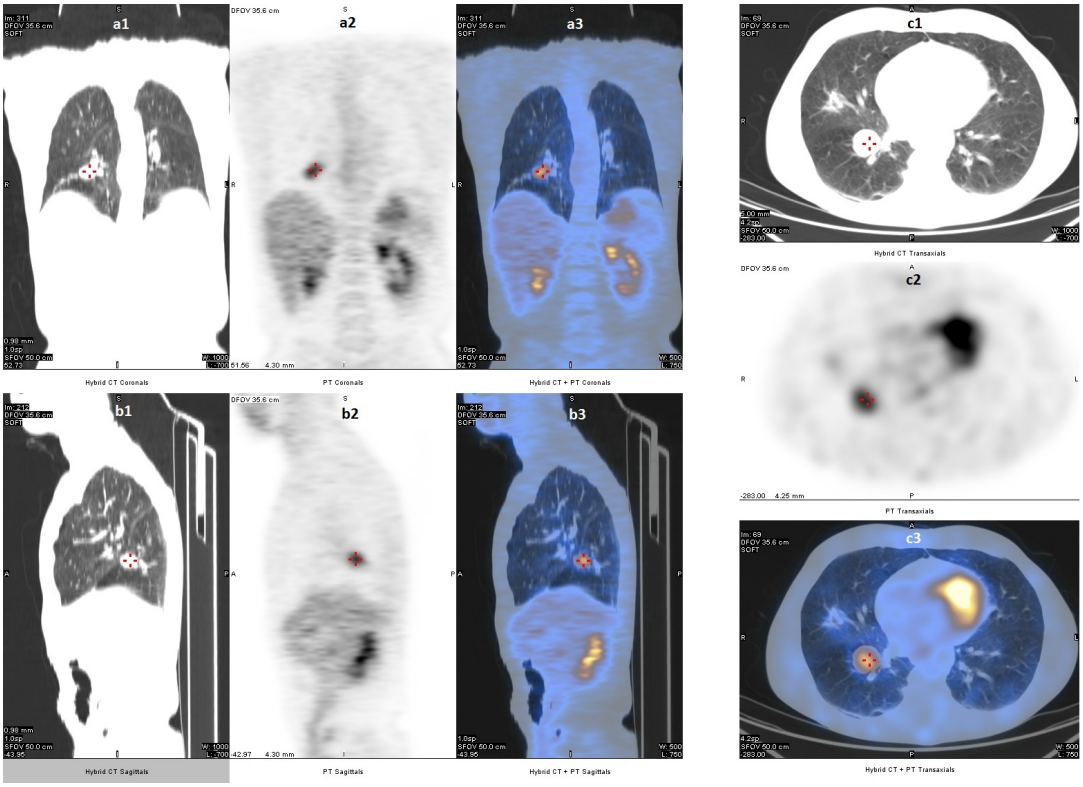


Figure 1.1: Example a 2-[^{18}F]FDG PET/CT exam of a patient with a nodule in the right lung. The nodule is represented in three orthogonal slices (a - coronal; b - sagittal; c - axial) on low dose CT, PET and image fusion (1, 2 and 3, respectively). A red mark was placed to identify the nodule.

- Create an annotated database of pulmonary nodules. This database will have two main types of datasets:
  - a 2-[^{18}F]FDG PET image dataset with cubic regions of interest of the nodules;
  - a tabular dataset with clinical and image features;
- Develop a Convolutional Neural Network (CNN) model for classification of pulmonary nodules. This step encompasses training, evaluation, tuning of several models by cross-validation. The best model will be selected to be evaluated in a disjoint set of unseen images (test set).

1.3.2 Secondary objectives

The secondary objectives are the following:

- Determine the optimum decision threshold for converting the predictions of the model into classes by threshold moving.

- Test the hypothesis that the CNN model outperforms the SUVmax for classification of pulmonary nodules. The SUVmax, together with the visual interpretation, is the current state-of-art for evaluation of pulmonary nodules in 2-[^{18}F]FDG PET/CT images.

1.4 Structure of the Dissertation

This dissertation has six chapters: *Introduction*, *Literature Review*, *Building a database of pulmonary nodules*, *Classification of pulmonary nodules*, *Results* and *Discussion*.

In the chapter *Introduction*, the relevance of the topic is explained, the problem is defined and the objectives of the dissertation are stated.

In the chapter *Literature Review*, an overview of the fundamentals of deep learning algorithms is provided, with emphasis on CNN, as well as their applications in PET/CT images for classification tasks.

In the chapter *Building a database of pulmonary nodules* are described the methods for building an annotated database of pulmonary nodules that includes PET images and tabular features. In this chapter, the target population and ground truth are defined, and the sampling, data collection and image data pre-processing methods are described. Descriptive statistics summarizing the ground truth and tabular features is also provided.

In the chapter *Classification of pulmonary nodules*, the machine learning task is formulated and the experimental setup for developing a deep learning model is described, which included cross-validation and testing on unseen examples.

The chapter *Results* describes the results of the several candidate models in the cross-validation and the results of the final model in the test set. A comparison is performed between the final model and the SUVmax measure.

In the chapter *Discussion*, the results of the classification models are interpreted and contextualized with the existing literature. The impact of the data quality on the predictive performance of the models is also addressed. The limitations of the research are detailed. In the end, several topics for future work are suggested.

Chapter 2

Literature Review

2.1 Introduction

Deep learning is a subfield of machine learning and concerns a group of artificial neural networks with several hidden layers (LeCun et al., 2015). Deep learning networks learn directly representations from raw data with no necessity of handcraft features being designed and extracted, overtaking an important limitation of classical machine learning algorithms (LeCun et al., 2015). The multiple layers allow to learn representations with a progressively higher level of abstraction (LeCun et al., 2015).

Such as the classical machine learning methods, deep learning can be divided on supervised and unsupervised depending on the existence or not of a target attribute (LeCun et al., 2015). Supervised tasks require labeled examples and the aim is to minimize a loss function that measures the difference between the predicted and the true label (LeCun et al., 2015).

CNNs are one of the most used network architectures. The first works on CNNs were done in the seventies (Litjens et al., 2017). The first successful network in a real-word application was the LeNet (LeCun et al.,1998, as cited in Litjens et al., 2017), capable of hand-written recognition. However, only in 2012, a network called AlexNet drew great attention from the Artificial Intelligence community owing to have won the ImageNet challenge by a large margin (Krizhevsky et al., 2012, as cited in Litjens et al., 2017). This only was possible due to the conjugation of new techniques that made possible the efficient training of the deep networks and the advances in core computing systems (Litjens et al., 2017).

The characteristics of deep learning algorithms make them the best methods for predictive tasks in many fields, nowadays, like computer vision and natural language processing (LeCun et al., 2015).

CNNs have become the most successfully algorithms for computer vision and numerous variants of this type of networks have arisen in the last years, making them more efficient (Khan

et al., 2020).

2.2 Deep learning in Computer Vision

2.2.1 Image classification

In computer vision, several tasks can be done by machine learning algorithms, such as classification, detection or segmentation (Rawat & Wang, 2017). Image classification consists of categorizing images into one of several predefined classes (Rawat & Wang, 2017). Instead of classify a whole image, one can also classify an object within a region of interest (Litjens et al., 2017), previously defined manually or by an automatic method.

Classification is a challenging task for machines because, among others, the viewpoint-dependent and in-class object variability (Rawat & Wang, 2017). The classical approach needs handcrafted features, which are used as input for a classifier (Rawat & Wang, 2017). This approach was highly dependent on the features extracted and requires domain expertise (Rawat & Wang, 2017). Differently, CNNs receive raw images, requiring few pre-processing (Litjens et al., 2017).

2.3 Fundamentals of Deep Learning Algorithms

2.3.1 Artificial Neural Networks

Artificial neural networks are the precursor of deep learning methods (Gu et al., 2018). Neural networks are nonlinear models that can approximate any function without necessity of assumptions about data distribution, being flexible for real world complex problems (Zhang, 2000). The basic computation unit of an artificial neural network is the neuron, in analogy with the human brain (Rawat & Wang, 2017). A basic artificial neural network has an input layer and an output layer. A multilayer feed-forward neural network has one or more hidden layers (J. Han et al., 2011). The input layer is fed with the feature vector and passes them to the following layer with no computation (J. Han et al., 2011). The hidden layers make successive non-linear transformations of the feature vector (J. Han et al., 2011). The output layer emits the predictions for each class, given the features vector (J. Han et al., 2011). Each neuron is a nonlinear function that takes, as net input, the weighted sum of the outputs from the units in the previous layer (J. Han et al., 2011).

2.3.2 Convolutional Neural Networks

Basic Principles

The CNN are a special type of multilayer feedforward networks. Its basic structure has several convolutional and pooling layers grouped into modules, and one or more fully connected layers (Rawat & Wang, 2017). In a conventional architecture, the pooling layers are placed between the convolutional layers and at the top of the network lies the full connected layers (Rawat & Wang, 2017; Gu et al., 2018).

The convolutional layer aims to learn feature representations from the inputs by using several convolution kernels (Gu et al., 2018). The neurons in the convolutional layer are arranged into feature maps, each neuron in a feature map receives the weighted sum of inputs from a neighbourhood of neurons in the previous layer and the set of these operations over whole input is a convolution (Gu et al., 2018). The size of neighbourhood is predefined and equal for all neurons in a feature map, and corresponds to the size of the convolution kernel (Gu et al., 2018). The set of trainable parameters is composed by the weights of a convolution kernel plus a bias (Gu et al., 2018). They are shared by the all spatial locations of the input, which greatly reduces the network complexity (Gu et al., 2018). Over the input convolved is applied a nonlinear function, resulting in a new non-linear feature map (Gu et al., 2018).

The pooling layers compute the maximum or the average value within a receptive field (Rawat & Wang, 2017). They reduce the resolution of the feature map and create invariance to small shifts and distortions, and merge semantically similar features (LeCun et al., 2015).

The output of the convolutional base may be flattened in a single vector of values and is taken by the fully connected layer to generate global semantic information from the data (Gu et al., 2018). The last layer is the output layer and contains an operator that depends on the predictive task, which gives the probability of a label given an example (Goodfellow et al., 2016). In classification is common to use softmax function (Gu et al., 2018).

The most common activation functions are the sigmoid, hyperbolic tangent and the rectified linear unit, but the last one is the most popular because allows a faster learning in networks with many layers and do not suffer from the vanishing gradient problem (LeCun et al., 2015; Rawat & Wang, 2017).

Training of the network is a global optimization problem and is performed with the backpropagation algorithm is the same way of the classical neural networks, through the stochastic descendent gradient or other similar (Rawat & Wang, 2017). The backpropagation algorithm calculates the gradient vector of a loss function with respect the weights and biases in order to know the magnitude and direction of adjusting the weights in the network (Rawat & Wang, 2017).

Main Recent Advancements

Numerous recent improvements have been proposed in order to reduce the computational cost and increase the performance in different tasks.

Convolutional layer. At level of the convolutional layer, the tiled convolution, the transposed convolution, the dilated convolution, the network in network and the inception module are examples of improvement proposals of the standard convolution. In tiled convolution, separate kernels are learnt within the same layer, providing rotational and scale invariant features, beyond the translational invariance (Le et al., 2010). The transposed convolution convert a single input to multiple output activations; the dimension of the resulting feature map depends of the stride (Gu et al., 2018). The dilated convolution is another variation that increase the receptive field by inserting a gap of zeros between the values of the kernel in an order which depends on a hyperparameter (dilation rate); it is used in tasks that need a large receptive field (Yu & Koltun, 2016, as cited in Gu et al., 2018). The network in network replaces the linear filter and the respective activation function of the convolutional layer by a micro-network able to represent more abstract features (M. Lin et al., 2014). Inception modules drastically reduce the number of parameters in the network by approximating an optimal local sparse structure (Szegedy et al., 2014)

Pooling layer. At level of pooling layer, others kind of pooling methods were proposed. The L_p pooling takes the weighted average of the activations (a_i) within a pooling region (R_j), being a trade-off between the average (p =1) and the max pooling($p = \infty$) (Rawat & Wang, 2017):

$$s_j = \left(\sum_{i \in R_j} a_i^p \right)^{1/p} \tag{2.1}$$

Mixed pooling combines max and average pooling, by randomly choosing either one or other at each location of a feature map (Rawat & Wang, 2017). Stochastic pooling randomly takes the activations within a pooling region according to a multinomial distribution, ensuring that non-maximal activations are also picked (Gu et al., 2018). In fractional max pooling, the stochastic nature lies in the selection of pooling regions, rather then the pooling operations (Rawat & Wang, 2017). Spectral pooling converts the input feature map into the frequency domain, then crops the frequency representation by maintaining only the central submatrix of the frequencies with the dimensions of the desired output, which is then converted into spatial domain (Gu et al., 2018). Spatial pyramidal pooling can be placed on top of the last convolutional layer to generate a fixed-length representation irrespective of the input size or scale, satisfying the size constrain of the input vector for the full connected neural network; the pooling is done in multiple local spatial bins proportional to the image size (He et al., 2014). The multi-scale orderless

pooling extract activation features from both the whole image and the local patches of several scales aggregated via vectors of locally aggregated descriptors (VLAD), then they are concatenated to form a new representation; this method captures fine-grained details and improves the invariance to large-scale global deformations (Rawat & Wang, 2017). In transformation-invariant pooling, new features are formulated from a predefined set of possible transformations, such that they are independent of any known nuisance variations of the input; over this set obtained is applied the max operator (Rawat & Wang, 2017).

Activation function. The rectified linear unit (ReLU), $f(x) = max(x, 0)$ is currently the most popular activation function; it leads to faster convergence and does not suffer of vanishing gradient problem of conventional non-linear functions (Rawat & Wang, 2017). However, the gradient descent is unable to fine-tune the weights of not previously activated units because their zero gradient (Lu et al., 2020). To compensate it, several variations of this activation function were designed which have still more faster convergence and better performance, as shown in the table 2.1 (Rawat & Wang, 2017). These variants of the ReLU compress the negative part instead of cancel it, allowing a small, not null gradient when the unit is not activated (Rawat & Wang, 2017).

Activation function	Expression
Leaky ReLU	$a = max(z, 0) + \lambda min(z, 0)$, $0 \leq \lambda \leq 1$, λ is predefined
Parametric ReLU	$a = max(z, 0) + \lambda_{\mathrm{k}} min(z_{\mathrm{k}}, 0)$, λ is learnt
Randomized ReLU	$a = max(z, 0) + \lambda min(z, 0), \lambda \sim U$
Exponential Linear Unit	$a = max(z, 0) + min(\lambda(\mathrm{e}^z - 1), 0)$, λ is predefined

Table 2.1: Activation functions - Leaky ReLU, Parametric ReLU, Randomized ReLU and Exponential Linear Unit.

Two alternative nonlinear functions were also proposed. The Maxout is an universal approximator which outputs the maximum value across the feature maps of a layer at a given position (Goodfellow et al., 2013). Probout output not the maximum, but one of the units among the feature maps according to a multinomial distribution (Gu et al., 2018).

Loss function. The training of the network is performed by minimizing a loss function (Q. Wang et al., 2020). The most suitable loss function depends on the machine learning problem (Q. Wang et al., 2020). The softmax cross-entropy loss is the most commonly loss function in multi-classification problems; it combines the softmax and the cross-entropy loss (Q. Wang

et al., 2020). The softmax is placed in the last layer of the fully connected network and gives a probabilistic output (Q. Wang et al., 2020). Examples of other loss functions for classification problems are sigmoid cross entropy loss, logarithmic loss, exponential loss, hinge loss, ramp loss and pinball loss (Q. Wang et al., 2020).

Regularization. Regularization includes any process that improves the model generalization against unseen examples. A regularizing effect can be obtained from a quite variate number of processes (Kukačka et al., 2017).

The model capacity can be adjusted to the complexity of the problem in order to prevent underfitting or overfitting (Kukačka et al., 2017). Different network architectures and different components (such as the type of convolutional or pooling layer or the activation function) can have a greater or lesser regularizing effect (Kukačka et al., 2017). Defining an experimental setup that includes a model selection step is a way of obtaining a model the best generalizes among several trained models with different hyperparameter setting (Kukačka et al., 2017). Stopping the training, as soon as the performance in validation set starts to get worse is another way to reduce the overfitting (Prechelt, 2012). Complexity of the model can be penalized by adding a regularization term to the empiric risk, the L_p norm regularization, where p typically assumes a value of 1 or 2 (Kukačka et al., 2017).

The dropout is a method used to reduce the overfitting by averaging many variations of the same fully connected network in an efficient way. A number of units is randomly omitted according to a Bernoulli distribution at each time a training batch is presented to the network, resulting a "thinned" network. The probability of dropping an unit (p) can be set at 0.5 or chosen from the validation set. On the other hand, in the test set, the unit previously dropped is now present, but its value is multiplied by p. For a network with n units, a total of 2^n thinned networks with shared weights are possible (Srivastava et al., 2014). DropConnect is similar to Dropout, but now randomly drops a subset of weights, according to a Bernoulli distribution (Gu et al., 2018).

Regularization via data can be done in three different ways. The most intuitive approach is gathering more data, but this is not always feasible. So the alternatives are to perform a preprocessing that transforms the data to some representation, which simplifies the learning task (Kukačka et al., 2017) or to perform data augmentation (Shorten & Khoshgoftaar, 2019).

Data augmentation entails a set of methods to artificially inflate the size of the training set when this one is of limited size as, for example, in medical imaging (Shorten & Khoshgoftaar, 2019). It assumes that more information can be extracted by this way from the original dataset and this improves the model generalization (Shorten & Khoshgoftaar, 2019). It can also be used for correction of imbalanced datasets (Shorten & Khoshgoftaar, 2019). Augmentation methods can be divided in oversampling and warping methods (Shorten & Khoshgoftaar, 2019).

Oversampling consists in duplicating instances of the minority class at random or interpolating new images for the existing k-Nearest Neighbors instances (SMOTE) (Shorten & Khoshgoftaar, 2019). Oversampling is suitable for correcting data imbalances, but not warrants a reduction of the overfitting and may even worsen it (Shorten & Khoshgoftaar, 2019). Data warping transforms the existing images while preserving the labels, and add them to the original dataset (Shorten & Khoshgoftaar, 2019). It can be further divided in basic image manipulations, such as geometric and photometric transformations, noise injection, kernel filters, mixed images, random erasing; deep learning approaches such as adversarial training, neural style transfer, generative adversarial networks-based augmentation; and meta-learning-based augmentation (Shorten & Khoshgoftaar, 2019).

Some of these methods are analyzed in more detail next. Geometric transformations include flipping, translation, rotation and cropping (Shorten & Khoshgoftaar, 2019). Geometric transformations are good to avoid positional biases (Shorten & Khoshgoftaar, 2019). However, the label is not necessarily preserved and the domain where is being applied should be taken into account when the transformation is designed and implemented (Shorten & Khoshgoftaar, 2019). Noise injection is done using a noise matrix; it is a good procedure to help the algorithm to learn more robust features (Shorten & Khoshgoftaar, 2019). Random erasing removes certain patches from the input image, preventing the model from overfitting to certain visual features in the image and enforcing it to pay attention to the entire image and to learn other features also presents. (Shorten & Khoshgoftaar, 2019)

The augmentation can be performed on learned representations instead of in the input images (Shorten & Khoshgoftaar, 2019). Unsupervised representation learning models, such as variational autoencoders or generative adversarial networks, offer a convenient way of learning useful representations for applying such transformations (Shorten & Khoshgoftaar, 2019). Augmentation using adversarial attacks, although may not represent probable examples in test set, they can improve weak spots in the learned decision boundary (Shorten & Khoshgoftaar, 2019). Generative adversarial networks-based augmentation creates artificial images which retain similar features to the original dataset (Shorten & Khoshgoftaar, 2019). Variational autoencoder is another generative algorithm which learns a low-dimensional representation from the data source and uses it for data augmentation (Shorten & Khoshgoftaar, 2019).

Regularization also can be achieved via optimization procedures. These techniques are described next.

Optimization. CNNs produce models that are difficult to train because they usually have millions of parameters and the loss function is non-convex (Rawat & Wang, 2017). The way as the weights are initialized contributes to prevent vanishing or exploding gradient problem and to promote fast convergence (Rawat & Wang, 2017). The main ways to initialize the weights

include randomly, with orthogonal matrices, unsupervised pretraining and transfer learning. Popular methods of random initialization are Xavier initialization and He initialization (Glorot & Bengio, 2010; He et al., 2015). The latter was specifically validated on rectifiers. The weight initialization follows an uniform or Gaussian distribution in an neuron-specific interval that maintains approximately constant the activation variance and back-propagated gradients variance along the different layers of the network (Glorot & Bengio, 2010; He et al., 2015).

Unsupervised learning algorithms (e. g. Restricted Boltzmann Machines, Auto-Encoders or Convolutional Auto-encoders) can be used to initialize the weights of all layers or, in some cases, only the first layer; the second phase corresponds to a supervised fine-tuning of the entire network (ur Rehman et al., 2019). This method reduce the dependency of the data and optimizes computation and convergence (ur Rehman et al., 2019).

Transfer learning is other option for when it is not possible to obtain a large labeled dataset about the domain of the problem or we do not have the necessary computational resources (Tajbakhsh et al., 2016). It consists in using a CNN that was previously trained in a large labeled dataset from another domain, preferentially related, or from the same domain, but in another task (Weiss et al., 2016). The convolutional network obtained from the source domain can be used as a feature extractor or as base network for model on the target domain (Tajbakhsh et al., 2016). When the transfer learning is used to create a feature extractor, the convolutional base of the source network are preserved and their weights are retained, and the fully connected network is replaced (Tajbakhsh et al., 2016). Thus, the extracted features from the target domain by the convolutional base are used to train the new fully connected network or other classifier (Tajbakhsh et al., 2016). When the transfer learning occurs for the whole network, the target convolutional network are initialized with the weights of the pre-trained source network with the same architecture (Tajbakhsh et al., 2016). Just the last fully connected layer is, if necessary, modified to have the same activation units as the number of classes of the target domain (Tajbakhsh et al., 2016). The weights of the last fully connected layer are initialized by a random method and the network is then fine-tuned in a layer-wise manner, starting by the last one; it is not always necessary fine-tuning all the layers; the number of layers fine-tuned relies one the dissimilarity between the source and the target domains (Tajbakhsh et al., 2016). The more layers are fine-tuned, the larger the dataset should be.

The backpropagation algorithm and an optimizer are used together to train a CNN. The gradient descent algorithm is a non-adaptive optimizer of widespread use (Ruder, 2017). It has three variants, the batch gradient descent updates the parameters θ of the objective function as $\theta_{t+1} = \theta_t - \eta \nabla_\theta E[\mathcal{L}(\theta_t)]$, where η is the learning rate and the $E[\mathcal{L}(\theta_t)]$ is the empiric risk calculated over the full training data (Gu et al., 2018). The stochastic gradient descent is a simplification, it does not calculate the exact value of the empiric risk and gives at each iteration

an estimation of the gradient based on a single example randomly picked from the training set (Gu et al., 2018). In practice, is used a mini-batch of examples instead of a single one in order to reduce the variance of parameter update and get a more stable convergence (Gu et al., 2018). Gradient descent does not guarantee convergence and the training may be terminated when reached a predefined number of epochs or a value of the training loss, or a performance measure stops improving in the validation set (Gu et al., 2018).

Adaptive optimization algorithms has recently arisen, such as RMSprop, Adam, Adagrad, Ada-delta, AdaMax and Nadam (Ruder, 2017). They shorten the training time by avoiding expensive tuning efforts of the optimizer (Curtis & Scheinberg, 2020).

The input distribution in a layer is affected by changes of parameters of the previous layers, so as data crosses subsequent layers, changes of the distribution are amplified and the learning ability is compromised, a phenomenon called internal covariate shift (Ioffe & Szegedy, 2015). Batch normalization introduces a step of normalization that fixes the mean and the variance of the input layers (Ioffe & Szegedy, 2015). The input normalized is then scaled and shifted for enhanced representation, using learnable parameters (Ioffe & Szegedy, 2015).

Even though the improvements of the normalized initialization and the batch normalization, they introduce a degradation problem on deep CNNs which can be corrected by shortcut connections (Gu et al., 2018). Highway networks use a learnable gating mechanism inspired by long short-term memory recurrent neural networks for regulating information flow across several layers without degradation (Gu et al., 2018). It uses gate functions to determine how much of an activation is to be transformed or just pass through (Gu et al., 2018; Rawat & Wang, 2017). On the other hand, residual networks uses shortcut connections that perform identity mapping, and their outputs are added to the outputs of the stacked layers (He et al., 2016). Thus, they fit a residual mapping with reference to the layer inputs, instead of learning unreferenced functions, being easier to optimize (He et al., 2016). Densely Connected Convolutional Networks connects each layer to every other layer in a feed-forward fashion (G. Huang et al., 2016). This architecture allows to reduce to vanishing-gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters (G. Huang et al., 2016).

2.4 Convolutional Neural Networks for Classification in PET and Multimodal Imaging

2.4.1 Types of input and 2D/3D CNN

Most of research on CNNs has been performed on natural images stored in a 2D (gray scale images) or a 3D (color images) tensor wherein the first two axes are the spatial dimensions and the third axis stores the channels of the color system, being three for RGB. A filter of a 2D convolution will have the same depth as the input image and just moves along the two first axes. The result is a feature map with two spacial dimensions, thus networks with these type of convolutions are usually called are 2D CNN. A convolutional layer with several filters outputs a feature map with a depth equal to the number of the filters in that layer (LeCun et al., 2015; Dumoulin & Visin, 2016).

On PET or on CT images, the data enter in the CNN as raw data or data transformed in a color space. The raw data are a 3D tensor with one channel. In case of PET, each voxel stores a measure of radioactivity concentration adjusted to dose injected and body weight, the SUV (Bailay et al., 2015). On CT images, the voxels store Hounsfield units (Bailay et al., 2015). Different approaches are described on the literature how to create CNN models with this type of inputs. If the network receives a slice or a set of slices (volume), but 2D convolutions are applied, this is a 2D CNN. When a volume enters a CNN that performs convolutions in 3 axes, this is a 3D convolution, and the network is called 3D CNN. One example of this type of networks is V-Net (Milletari et al., 2016).

2.4.2 Systematic review

A systematic literature review was performed on Pubmed[1] to find all papers about classification tasks on PET or multimodal imaging involving PET (PET/CT and PET/MRI) using CNN-based models. The search criteria were the following: *"convolutional neural network" AND (positron emission tomography OR PET OR PET/CT OR PET/MRI)*. From the 54 papers found, original papers respecting classification tasks (diagnostic or prognostic) were selected and analyzed (n=16).

Five studies were performed on Alzheimer disease, 1 on Parkinson disease, 1 on cardiac sarcoidosis, 2 on head and neck cancer, 3 on lung cancer, 2 on esophageal cancer, 1 on cervical cancer and 1 on unspecified brain pathology. There was no study about classification of indeterminate pulmonary nodules.

[1] https://pubmed.ncbi.nlm.nih.gov/

The description of the different approaches is divided according to the network architecture (2D or 3D).

2D CNN

Ypsilantis et al. (2015) sought to predict the response to neoadjuvant chemotherapy in patients with esophageal cancer from pre-treatment 2-[^{18}F]FDG PET images by building traditional models based on radiomics and CNNs. The dataset had 107 patients. From a given volume containing the tumor, all possible triplets of adjacent transaxial slices were extracted. Each triplet was treated as a three-channel input for the CNN and was labeled. The CNN architecture was composed of 4 convolutional and 4 max-pooling layers and a fully connected network. This model had an accuracy of 73.4%.

A classification model was developed by H. Wang et al. (2017) for differential diagnosis of 2-[^{18}F]FDG uptake in mediastinal lymph nodes in patients with lung cancer. From 168 patients were analysed 1397 lymph nodes. Six transaxial patches, 3 for PET and 3 for CT, were cropped around the center of each lymph node and resampled. The patches were translated and rotated for data augmentation, as such the sample of patches were extended by a factor of 729. The architecture of the network is based on AlexNet, but constrained to 5 layers. ReLU activation function, dropout regularization, categorical cross entropy loss function and Adadelta learning were used. The output is a patch-based probability of malignancy. The model obtain an area under the ROC curve of 0.9086 in the test set.

A CNN model for local staging of lung cancer on 2-[^{18}F]FDG PET/CT images was developed by Kirienko et al. (2018) from a dataset with 472 patients. The model aimed to distinguish between T1-T2 and T3-T4 tumors. The PET images were resampled in the CT space. Both images were rescaled to interval 0-1. 3D bounding boxes were cropped around the lesion centre in both images ($128 \times 128 \times N_{slices}$). Data augmentation with rotational transformations was performed. The algorithm was composed by two networks: a feature extractor and a classifier. The 3D region of interest was decomposed in N_{slices}, which corresponds to 2D images. The total of N_{slices} of all patients is the dataset that feeds the network. Therefore, each patch has 128×128, and the PET and CT data were stored in separate channels. The feature extractor is a 2D CNN. The classifier receives the input mean of the second to last layer of features extracted from all slices of a single patient in order to give a per patient prediction. The final model had an area under the ROC curve of 0.68 in the test set.

Y. Han et al. (2021) built a classification model for differentiating histologic subtypes of non-small cell lung cancer from fusion images of 2-[^{18}F]FDG PET/CT. The dataset had images from 1419 patients. Data augmentation was applied. Ten classical machine learning mod-

els and a VGG-16 model pre-trained on ImageNet were evaluated. The deep learning model outperformed all classical machine learning models, having an area under the ROC curve of 0.903.

Prediction of local relapse and distant metastases was performed by Shen et al. (2019) in cervical cancer patients. A total of 142 patients were included. A 3D region of interest centered on SUVmax was extracted and decomposed in slices for the 3 orthogonal axes. Rotational transformations were performed for data augmentation. A total of 1562 set of slices for the 142 tumors was generated and labeled. The model consisted of a 2D CNN with 3 multilayer perceptron convolution layers and 1 global average pooling. Batch normalization was performed. The output is a slice-based probability, so this probability was aggregated to obtain tumor-based probabilities. The model had an accuracy of 89% for local relapse and 87% for distant metastases.

Liu, Cheng, Yan, and the Alzheimer's Disease Neuroimaging Initiative (2018) proposed a model for diagnosis of Alzheimer disease from 2-[^{18}F]FDG PET images based on combination of 2D CNN and recurrent neural networks (RNNs). A total of 339 individuals have participated in the study. The volume of interest of the brain was decomposed in slices (2D images) in the 3 orthogonal axes: 193 coronal slices, 153 sagittal slices and 163 transaxial slices. Additionally, groups of 15 slices with an overlap of 6 slices were created to train a 2D CNN. A total of 20 deep CNNs in coronal, 17 CNNs in sagittal and 18 CNNs in axials were obtained to generate intra-slice feature vectors for 3D 2-[^{18}F]FDG PET images. The CNNs with low accuracy were discarded. The features generated from 4 slices was fed into the inputs of Bidirectional Gated Recurrent Unit (BGRU) in each axis. The output of each BCRU was concatenated to feed a fully connected network. The final classification was performed by weighted averaging of the predictions from the 3 axes. This method reached an area under the ROC curve of 0.953 to distinguish Alzheimer disease from normal controls, and 0.839 to distinguish mild cognitive impairment from normal controls.

Ding et al. (2019) proposed a CNN model based on 2-[^{18}F]FDG PET for diagnosis of Alzheimer disease. A total of 1002 participants and 2109 exams were included in the study. They converted the volume of interest in a grid of uniformly spaced 4×4 transaxial slices. This new 2D image is the input for the CNN. Data augmentation was performed with random width and height shift and zooming. The architecture Inception-v-3 was used. The weights were initialized with a network pre-trained on ImageNet. The model had an area under the ROC curve of 0.98, outperforming human readers.

Nobashi et al. (2019) built models to distinguish between normal and pathological brain 2-[^{18}F]FDG PET from a dataset with 289 participants. They converted data to .png format to transform the raw data into a color system. The window was previously defined by the opera-

tor. The images were analyzed in 3 different windows. For each window were obtained coronal, sagittal and transaxial slices. Each slice was individually annotated e corresponded to an example. A total of 9 models were built for the combination of windows and slice axis. The output of the models was a probability of disease by slice. To obtain the probability for a patient, the probability of the slices of the same axis and window was averaged. The architecture of the network was based on ResNet-50. Ensemble models were created for windows and/or slice orientations. The best model was the ensemble of the slices on 3-axes for the narrow window (area under the ROC curve of 0.8405).

Togo et al. (2019) developed a model for diagnosis of cardiac sarcoidosis from 2D cardiac polar maps generated from 2-[^{18}F]FDG PET images. The dataset had 85 patients. They used a pre-trained Inception-v3 network model in object recognition tasks as a high-level feature extractor. A 2048-dimensional feature vector was extracted from the pool3 layer. Because the feature extracted can be effective on the task of the source model, but not on classification tasks, a ReliefF feature selection algorithm was used to select more suitable features for classification. Finally, a linear Support Vector Machines was applied. The F1 score of the method was 0.854 in the test set.

3D CNN

Yee, Popuri, Beg, and the Alzheimer's Disease Neuroimaging Initiative (2020) proposed a 3D CNN with residual connections for diagnosis of Alzheimer disease with 2-[^{18}F]FDG PET imaging. A dataset with images from 1211 participants was used. This network takes as input a volume of the full brain appropriately pre-processed. The network has a total of 8 convolutional layers, 3 max pooling layers and two residual learning blocks to learn hierarchical features. The top layers consist of a global average pooling layer, a 1×1 convolutional layer and the softmax activation rather than a fully connected layers in order to reduce the number of parameters. Batch normalization and dropout were applied. This method had an area under the ROC curve of 0.976 to distinguish Alzheimer patients from control individuals, in an independent set.

Choi and Jin (2018) sought to predict the development of Alzheimer disease in patients with mild cognitive impairment through brain 2-[^{18}F]FDG PET and ^{18}F-florbetapir PET. The dataset had images from 492 patients. A 3D CNN was trained on dataset of Alzheimer disease patients and normal controls and transferred to the target dataset without fine-tuning. Two volumes of 2-[^{18}F]FDG PET and ^{18}F-florbetapir PET were spatially co-registered and stored in 2 different channels. The model yielded an area under the ROC curve of 0.89, in the test set.

Y. Huang et al. (2019) propose CNN models based on single modality (2-[^{18}F]FDG PET or MRI) and multimodality (2-[^{18}F]FDG PET/MRI) for diagnosis of Alzheimer disease. The

dataset had images from 1378 participants. The CNN was like VGG, being adapted to 3D images. On MRI images, models derived from 3 regions of interest were compared (hippocampus, segmented hippocampus and hippocampal mask). On PET images, models trained from two regions of interest were compared, a hippocampal and a bigger one including also the cortex. For models of multimodal imaging, the PET image was rigid-registered with the respective MRI image and the hippocampal interest region was cropped. Two types of approaches were evaluated, in one of them, the PET/MRI was input to the network as a 4D tensor. The first 3 are spatial dimensions and the fourth is the channel where the data of PET and MRI were stored for each spatial point. A CNN was adapted to receive this type of input. In the second approach, two separate convolutional/pooling bases from a VGG-11s were used to extract features from the two image modalities and the features were then concatenated and received by a fully connected network. This approach was further subdivided in two, being the weights either totally independent or shared by the two convolutional/pooling bases. The PET 3D CNN with hippocampal region had the best area under the ROC curve (92.69%), whereas the PET/MRI with 3D CNN and shared weights had the best accuracy (90.10%).

Manzanera et al. (2019) built a 3D CNN from 2-[^{18}F]FDG PET images for diagnosis of Parkinson disease. The dataset had images from 310 participants. A pre-processing with Scaled Subprofile Modeling using Principal Component Analysis was performed in order make a mask, selecting the voxels that include the patterns of brain metabolism related to the disease. Over this mask, 3 cubic regions of interest were cropped from different regions of the brain that provide input to 3 different 3D CNNs, respectively, with identical configuration. The output of the 3 CNNs are concatenated and passed thought an output layer. Due the small size of the dataset (n=310), the three different models created the CNN was shallow, having the 1 to 3 convolution layers. Batch normalization and dropout were applied. The best model had an area under the ROC curve of 0.94 in the test set, for the approach with one convolutional layer.

C.-K. Yang et al. (2019) developed a 3D CNN-based model to predict survival at 1 year in esophageal cancer from a 2-[^{18}F]FDG PET imaging dataset of 1107 patients. They cropped a volume that covered the body area from the hypopharynx to the stomach and included all of the esophagus and the peri-esophageal regions. Data augmentation was performed with geometric transformations and Gaussian noise injection. Data balancing with under- and over-sampling was done. The 3D CNN was based on ResNet and had 34 layers, anisotropic max pooling, Adam optimizer, batch normalization and a rectified linear unit. A pre-trained model was derived from a mix of this dataset with another one of lung cancer patients in a task of classification esophageal/lung cancer. The final model had an area under the ROC curve of 0.738, in the test set, to predict survival at 1 year.

First Zhou Zhou et al. (2018) as a preliminary result and then L. Chen et al. (2019) in an ex-

tended sample, both from the same team, they predict the malignancy of regional lymph nodes in patients with head and neck lung cancer. The dataset used by L. Chen et al. (2019) had 31 patients and 121 lymph nodes. A hybrid model was developed, combining a multi-objective Support Vector Machines model based on radiomic features and a 3D CNN from multimodality imaging (2-[^{18}F]FDG PET and CT). The output of the two models were combined by evidential reasoning. The input to the CNN model corresponded to patches of PET and CT that included the lymph nodes and their surrounding voxels. These patches are co-registered and form a 4D tensor, or 2 3D images, one in each channel. The minority class was oversampled with Synthetic Minority Oversampling Technique (SMOTE). Data augmentation was performed with rotational transformations. The architecture of the 3D CNN consisted of 12 convolutional layers, 2 max-pooling layers and 2 fully connected layers. Each convolutional layer has rectified linear units and batch normalization. Weights were initialized with Xavier initialization. The area under the ROC curve was 0.95 on this task, in the test set.

This page is intentionally left blank

Chapter 3

Building a database of pulmonary nodules

A database of pulmonary nodules was constructed with data of patients referenced to the São João University Hospital Centre, having a pulmonary nodule investigated by 2-[^{18}F]FDG PET/CT.

Once there are no PET facilities in that hospital, this imaging exam is currently supplied by external providers.

In order to ensure the quality of the data for modeling, the first steps were to define a study population, the ground truth and the sampling method. Individuals of the study population share a set of eligibility criteria. These criteria allow to identify the individuals of the interest and draw a sample as representative as possible using a sampling method, which is used to build the dataset. The models trained and evaluated in this dataset only will be valid for classifying pulmonary nodules of new individuals drawn from the same population, unless the models are subsequently validated in additional datasets built from other populations.

3.1 Target Population

Each participant cumulatively meets the following inclusion criteria:

- One or more indeterminate solid pulmonary nodules with more than 8 mm in average diameter. The average diameter should not exceed the 30 mm, according to nodule definition;
- The nodule detection was incidental or through screening;
- 2-[^{18}F]FDG PET/CT was performed for clarification of the nodule(s) and the reconstructed images are available in digital format;

- The nodule was biopsied or excised and obtained a histopathological or cytopathological examination, otherwise completed an imaging follow-up period.

Who meets at least one of the following criteria is excluded:

- History of lung cancer;
- History of other cancers, except:
  - Non-melanoma skin cancer, localized prostate cancer, *in situ* cervical cancer, *in situ* breast cancer, or superficial bladder cancer, which has been treated at least 6 months ago;

3.2 Definition of the ground truth

The ground truth corresponds to the definitive diagnosis of the pulmonary nodule, classifying the nodule according to the best existing evidence, which is in this case the histopathological or cytopathological examination, and/or the nodule behavior during the follow-up period with CT.

The ground truth is thus defined as follows:

1. A nodule is classified as malignant if biopsied or excised during the initial diagnostic workup or during the follow-up period, and the histopathological or cytopathological examination shows a malignant neoplasm.

   Malignant nodules can be further detailed, being grouped in seven categories: adenocarcinoma, squamous cell carcinoma, small cell lung cancer, large cell carcinoma, carcinoid tumour, metastasis, other/uncertain tumor.

2. A nodule is classified as benign if

   (a) Excised and the histopathological examination showed benign pathology;

   (b) Biopsied, the biopsy was diagnostic and the histopathological or cytopathological examination showed benign pathology;

   (c) Neither excised nor biopsied, or biopsied but non-diagnostic and during follow-up:

      i. The nodule disappeared;

      ii. The nodule decreased or kept the same size for, at least, two-year of follow-up;

iii. the nodule increased in size and thereafter was biopsied or excised and the histology was benign;

iv. Volume doubling time >600 days and <25% change in volume for, at least, one year of follow-up.

A minimum of two-year imaging follow-up is established for solid nodules when the mean diameter of the nodule on two perpendicular axes, obtained on axial slices, is used for follow-up. When the follow-up period was between one- and two-year, the nodular volume was estimated from the diameter on three orthogonal axes. These follow-up criteria are based on the doubling time of malignant solid nodules and is recommended in the two main guidelines (Callister et al., 2015; MacMahon et al., 2017) of pulmonary nodule management.

3.3 Sampling method

Every patient referred to São João hospital and who underwent 2-[^{18}F]FDG PET/CT between 2010 and 2019 was consecutively selected if he/she belongs to the defined population. If a patient underwent more than a PET/CT exam, only the first one was considered. If a patient has more than one nodule that fills the eligibility criteria, only the more suspicious was included.

3.4 Ethical aspects and Data Protection

This research was approved by the São João University Hospital Centre, EPE (Project no. 371/19), including the retrospective collection and processing of clinical data, and modeling. The project was approved by the Head of Nuclear Medicine Department, the Ethical Committee, the Responsible for Data Reuse, the Research Unit Coordinator and the Administration Board. The informed consent of the participants was waived due to retrospective nature of the research. The data were submitted to pseudonymization during the collection phase. After completed this phase, the data were anonymized.

3.5 Evaluation of eligibility and selection of participants

Among 7130 PET/CT exam requests within the established time interval were selected 292 2-[^{18}F]FDG PET/CT exams that aimed at clarifying the diagnosis of pulmonary nodules. Then, the eligibility criteria were checked for those 292 by consulting the medical records and the information of the histopathological/cytopathological examination, the standard-dose CT

and the 2-[^{18}F]FDG PET/CT. In the end, 113 participants met the eligibility criteria to be included in the sample used to create a database. One nodule per participant was included.

A table ("amostragem") in the database was created to record the verification of the eligibility criteria (A.1).

3.6 Data Collection and Preparation

The database is conceptually constituted by image and tabular data.

3.6.1 Image Data Collection

Building an image dataset required locating the digital image files of the study participants in different sources and assemble them in a single directory of an external hard drive. The figure 3.1 makes an overview about the steps of collection e preparation of image data.

Before 2015, only the printed reports were being delivered to the hospital by the external providers, as such the digital image data of 49 exams were directly requested to providers which kindly provided them. The exams performed between 2015 and 2017, were stored in Compact Disc (CD) in the clinical archive of the hospital, so a search in this archive was manually conducted to find 60 exams. The remaining exams were located digital imaging database of the institution - Picture Archiving and Communication System (PACS), since after 2017, this kind of exams were integrated there.

3.6.2 Initial Understanding of the Image Data

A 2-[^{18}F]FDG PET/CT exam is composed by three types of images: PET images reconstructed with attenuation and scatter correction, PET images reconstructed without attenuation and scatter correction and CT images. These images have a field of view between the base of the skull and the middle of the thighs and are acquired around 60 min after the tracer injection. Additional images may be acquired, for instance, at 120 min, which in this context are usually from the thorax.

For the purpose of the current project were only used the PET images reconstructed with attenuation and scatter correction acquired at 60 min, hereinafter referred as reconstructed PET images.

PET images are tomographic images, therefore represent volumes. These volumes are usually stored as series of DICOM files of axial slices, each one identified with its spatial coordinates. The DICOM format is a standardized format file in medical imaging (*Digital Imaging and Communication in Medicine*, n.d.). It has two parts, a header and a dataset encapsulated. The

header is formally called of File Meta Information, and is composed by a 128 byte File Preamble, followed by a 4 byte DICOM prefix, followed by the File Meta Elements. It stores numerous information, such as study date, department name, patient data, procedure and equipment details. The dataset may contain one or more 2D image frames. In case of PET images, one file usually contains just one 2D image frame, therefore can be read as a data matrix that stores information about the biodistribution of the 2-[^{18}F]FDG in MBq/mL.

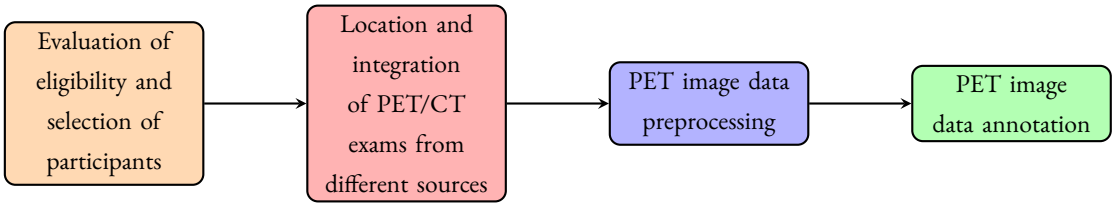


Figure 3.1: Flowchart for building the PET image dataset.

3.6.3 Image Data Preprocessing

In this section will be described the method of preprocessing of reconstructed PET images to obtain an image dataset of pulmonary nodules.

Software

The image preprocessing was performed in the 3D Slicer 4.10.2 r28257 (*3D Slicer*, 2020 (accessed January 26, 2020)). 3D Slicer is a free, open-source software for analysis and visualization of medical images (Fedorov et al., 2012).

Importing and reading data

Both reconstructed PET data files and the CT files were imported for 3D Slicer. The images were loaded with two plugins:

- DICOMScalarVolumePlugin - loads the data as scalar volumes;
- DICOMPETSUVPlugin - recalculates the voxel data to represent units of standardized uptake value normalized by the body mass (SUV) rather than MBq/mL.

Coregistration

The PET and CT data were corregistered with rigid registration using the Landmark Registration extension. That ensures the alignment of the different anatomical structures and the proper location of the pulmonary nodule of interest in PET images through CT information

when the nodule does not uptake 2-[^{18}F]FDG, for subsequent cropping of the corresponding volume of interest.

Parameters: Fixed volume: CT, Moved volume: PET, Local refinement method: local simpleITK, Local SimpleITK mode: Simple, Registration: affine registration; Linear registration - mode registration: rigid.

Image spatial resampling

The PET/CT exams were acquired in three different scanners, having each scanner different voxel size and anisotropic spacing of the PET volumes. As such, the volumes were resampled to have the save voxel size and isotropic voxels. The voxel side was defined as 1.5 mm which is smaller than the smaller voxel side of the 3 scanners. Linear interpolation was used for spatial resampling.

Nodule identification and region of interest cropping

The nodule was visually identified in the coregistered PET/CT images and a cubic region of interest was drawn and cropped to include the entire nodule. The center of this subvolume coincides with the center of the nodule. The subvolume has a side length of two times of the maximum possible diameter of the nodule (30 mm), that is 60 mm x 60 mm x 60 mm, corresponding to a 216000 mm^3. The cubic region of interest was cropped and saved in .nrrd format. This format stores a header and a tensor of data with rank of 3 and shape of 40 x 40 x 40. The first axis is the left-right, the second one is the posterior-anterior, and the third one is the superior-inferior. A directory was created to save the cropped images.

3.6.4 Tabular data collection

Several features related with lung cancer risk factors and handcrafted image features from PET and low-dose CT were defined. The information about the age, sex, smoking habits, occupational carcinogens, pulmonary diseases, family history of lung cancer and nodules location was collected from the clinical records. The information about the nodule morphology, nodule multiplicity, lymphadenopathy was collected by visual analysis of the low-dose CT images. The information about the maximum tracer uptake in the nodule and in possible adenopathies (SUVmax) was collected from the reconstructed PET before cropping. The PET/CT scanner model was automatically collected from the header of the DICOM files. A column with the relative paths for the cropped images was also added. Furthermore, both binary and multiclass target features were created from ground truth criteria by collecting information about the

histopathological/cytopathological examination and/or the nodule behavior during the follow-up period with standard-dose CT. The binary target was later used to annotate the preprocessed PET image dataset.

The table A.1 of the appendix A shows a description of the features defined, namely the type, scale, encoding, and unit of measurement. The tabular data were organized in a MySQL database (figure A.1). The features defined in the table A.1 were saved in the table "datanodules_1". The criteria for ground truth definition and its resulting target variables are in the table "amostragem".

A thorough definition of the some features is described following in order to make the table A.1 easier to interpret. The patient age is the age at the time of the PET/CT scan. Pack-year is a measure of person's exposure to tobacco, taking into account how long a person has smoked, and how much he/she has smoked. Pack years is the number of cigarettes smoked per day / 20 $\times$ number of years smoked. Occupational exposition to carcinogens is defined as exposition to carcinogens causally associated with lung cancer (arsenic, chromium, nickel, asbestos, tar, and soot) (Alberg et al., 2013). The nodule multiplicity intends to records the total number of pulmonary nodules detected in a patient, including nodules that not filled the selection criteria to the study, by visually analysis the low-dose CT scan. Family history of lung cancer is present when the disease affects a first degree relative (de Groot et al., 2018). The nodule diameter corresponds to the mean of the long-axis and the perpendicular axis, both measured in a axial slice of the CT and rounded to the nearest unit.

Spiculation is a kind of appearance of the nodule contour in CT scan. Pleural indentation consists of a linear opacity that extends from a peripheral nodule to the visceral pleura in CT images. Mediastinal or hilar lymphadenopathy is defined as at least one enlarged non-calcified lymph node in mediastinum or pulmonary hila. A lymph node is defined as enlarged when its transaxial short-axis measures at least 10 mm in low dose-CT. The CT data are collected from the low dose-CT scan performed as part of the protocol of the PET/CT scan.

3.7 Descriptive Statistics of the Dataset

3.7.1 Characterization and quality of the ground truth

The dataset has 113 participants. The number of participants selected per year is presented in the chart A.2 of the appendix A. One nodule was included by participant. Fifty-one (45.1%) malignant pulmonary nodules were found. The remaining were benign. The table 3.1 shows the distribution of the nodules according to the type, detailing the histological type of the malignant nodules.

The ways of obtaining the ground truth, that is the definitive diagnosis of the nodule, were recorded in the table 3.2 as a quality measure. When the definitive diagnosis was obtained by follow-up CT imaging, the median follow-up was 2.6 years (minimum: 1.3 years; maximum: 8.3 years); 85.4% of the participants had a follow-up time $\geq$ 2 years.

Class	Absolute frequency	Relative frequency (%)
Adenocarcinoma	31	27.4
Squamous cell carcinoma	4	3.5
Small cell lung cancer	2	1.8
Large cell carcinoma	2	1.8
Carcinoid tumor	7	6.2
Metastasis	0	0.0
Other/uncertain cancer	5	4.4
Benign nodule	62	54.9

Table 3.1: Definitive diagnosis of the nodules.

Criterion	Absolute frequency	Relative frequency (%)
Histological diagnosis	71	62.8
Citological diagnosis	1	0.9
Follow-up imaging	41	36.3

Table 3.2: Ways of obtaining the ground truth.

3.7.2 Characterization of tabular data

In order to have an understanding of the profile of the participants whose nodes compose the dataset, an univariate analysis of the different registered attributes was carried out. Qualitative attributes were characterized by absolute and relative frequencies. The qualitative features are in the tables 3.3 and 3.4. Quantitative attributes were characterized with central tendency, dispersion and shape measures and are shown in the table 3.5.

Feature	Absolute frequency	Relative frequency (%)
Sex		
Male	76	67.3
Female	37	32.7
Smoking habits		
No smoking history	34	30.1
Current or previous smoking history	79	69.9
Occupational exposition to carcinogens		
No exposition	110	97.3
With exposition	3	2.7
Emphysema or pulmonary fibrosis		
No disease	71	62.8
Disease is present	42	37.2
Family history of lung cancer		
No familiar history	110	97.3
With familiar history	3	2.7
Nodule Multiplicity		
Solitary nodule	71	62.8
Multiple nodules	42	37.2
Nodule location		
Right upper lobe	35	31.0
Right middle lobe	11	9.7
Right lower lobe	23	20.4
Left upper lobe	19	16.8
Left lower lobe	25	22.1
Nodule spiculation		
No spiculation	79	69.9
Spiculation is present	34	30.1
Pleural indentation		
Absent	86	76.1
Present	27	23.9
Mediastinal or hilar lymphadenopathy		
No lymphadenopathy	105	92.9
With lymphadenopathy	8	7.1

Table 3.3: Distribution of the patients according to sex, risk factors and imaging features. In case of several nodules, the imaging features refer to the dominant nodule.

Feature	Absolute frequency	Relative frequency (%)
Scanner model		
Siemens Biograph 6	57	50.4
GE Discovery LS	42	37.2
GE Discovery IQ	14	12.4

Table 3.4: Distribution of the patients according to the PET/CT scanner model.

	Mean	Median	SD	Maximum	Minimum	IQR	Skewness	Kurtosis
Age	64.47	65.00	11.21	90.00	28.00	14.00	-0.45	0.36
Pack-year	45.33	40.00	43.46	258.00	0.00	66.00	1.36	3.78
Ex-smoker time	4.53	0.00	9.62	45.00	0.00	3.50	2.47	5.78
Nodule diameter	14.07	13.00	4.59	30.00	9.00	5.00	1.32	1.49
Nodule SUVmax	2.99	1.64	3.12	18.13	0.53	2.51	2.31	6.21
Adenopathy diameter	13.44	13.00	2.51	19.00	11.00	2.00	0.99	-0.07
Adenopathy SUVmax	4.70	4.30	2.87	13.55	0.89	2.52	1.49	2.67

Table 3.5: Description of tabular quantitative features. SD - standard deviation. IQR - interquartile range.

Chapter 4

Classification of pulmonary nodules

4.1 Formulation of the machine learning task

Two schemes of annotation of the pulmonary nodules were performed, resulting in a binary and in a multiclass target. Although the latter details better the main pathological categories of pulmonary nodules, the former was preferred because it simplifies the machine learning classification task.

The collected data are quite diverse in terms of data structure and type of information stored, but for modelling purposes in the current dissertation, only the PET data will be used. The remaining preprocessed data serve to contextualize the dataset and support future work.

Therefore, the supervised machine learning problem is a single classification task, single label, binary classification problem that uses cubic regions of interest from PET in the format of 3-axis tensors, as input data for a 3D CNN.

A mathematical formulation of the problem is stated as follows. Let X be a input tensor, where $\mathsf{X} \subseteq \mathbb{R}^{d_1 \times d_2 \times d_3 \times d_4 \times d_5}$ and d_1 is an unknown value of size of the population. $\mathsf{X_i} \subseteq \mathbb{R}^{d_2 \times d_3 \times d_4 \times d_5}$ is the i instance of X in the population. d_2, d_3 and d_4 are the shape of the volume of interest, and d_5 is the number of channels. $\boldsymbol{Y}$ is a vector of targets where each $\boldsymbol{Y_i} \in \{benign\ nodule : 0, malignant\ nodule : 1\}$ has a correspondence to each $\mathsf{X_i}$. Assuming that n independent and identically distributed instances $\mathsf{X_i}$ were randomly drawn from the population to create a training set $\mathbb{S}$, having fixed and unknown distributions of $P(\mathsf{X})$ and $P(\boldsymbol{Y}|\mathsf{X})$. Let be $f_{\mathcal{CNN}(S)} : \mathsf{X} \rightarrow \boldsymbol{Y}$, a function learned by a 3D CNN from $\mathbb{S}$. $f_{\mathcal{CNN}(\mathbb{S})}$ is called classifier or hypothesis and belongs to a set of functions of the hypotheses space $\mathcal{F}$. The learning problem consists of choosing from the hypotheses space, the function $f^*_{\mathcal{CNN}(\mathbb{S})}$ that best maps each input $\mathsf{X_i}$ to each class of the target attribute, $\boldsymbol{Y} = f(\mathsf{X}; \theta)$, by discovering the weights (θ) configuration that minimizes misclassification risk.

The true or functional risk $R(f_{CNN(\mathbb{S})})$ cannot be directly computed because the true

data distribution is unknown $p_{data} := P(\mathsf{X}, Y) = P(\mathsf{X})P(Y|\mathsf{X})$. So, one can estimate the p_{data} from the empirical distribution in the training set, $\widehat{p}_{data}$, and determine the empirical risk $R_E(f_{CNN(\mathbb{S})})$ by averaging the result of a loss function between each predicted target $f(\mathsf{X_i}; \theta)$ and the correspondent true target class $\boldsymbol{Y_i}$ of the training set (Vapnik, 1999), as follows:

$$R_E = \mathbb{E}_{(\mathsf{X},Y)\sim\widehat{p}_{data}}[\mathcal{L}(f(\mathsf{X}; \theta), Y)] = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(f(\mathsf{X}_i; \theta), Y_i) \tag{4.1}$$

The result is known as structural risk, when a regularization term ($R(...)$) is added to the equation (Q. Wang et al., 2020).

Training a CNN is an optimization problem that aims to approximate the $f^*_{\mathcal{CNN}(\mathbb{S})}$ by minimizing the structural/empirical risk, a surrogate of the functional risk. Risk minimization on the training set is prone to overfitting, so a dissociation between the true and the estimated risks is expected to occur at some point during the training. An estimate of the functional risk can only be obtained from a validation set and cannot be used for updating the model parameters, but only to decide when the training should be halted (Goodfellow et al., 2016).

The empirical and structural risk are commonly referred to as training loss in the literature and, for the sake of simplification, the latter denomination will be henceforth adopted, throughout this dissertation, unless expressly stated otherwise.

4.2 Computational resources

The experiments of training and evaluation were performed in R $4.0.0$-$4.0.3$ (R Core Team, 2019). Tensorflow for R $2.2.0$ (Allaire & Tang, 2019) and Keras R $2.3.0.0$ (Allaire & Chollet, 2019) packages provide an R interface to Tensorflow and Keras Python packages. Anaconda Navigator $1.9.12$ (*Anaconda Software Distribution*, 2020) (conda $4.9.2$) was installed. An environment "r-reticulate" with Python $3.7.8$ was created to install Tensorflow. All experiments were performed with Tensorflow $2.1.0$, GPU version (Abadi et al., 2015). Keras comes packaged with TensorFlow.

Tensorflow (Abadi et al., 2015; Allaire, n.d.) was originally developed by researchers and engineers working on the Google Brain team. It is an open-source software library for numerical computation using data flow graphs. It has a flexible architecture and allows deploying computation to one or more CPUs and GPUs with a single API. It is one of the most used ecosystems for building and training deep learning models.

Keras (F. e. a. Chollet, 2015) is a high-level neural networks API capable of running on top of TensorFlow, CNTK, or Theano. It is used for easy and fast prototyping, and for training deep learning modes. Runs on CPU and GPU.

A laptop with Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz processor, 8.0 GB of RAM memory and graphic cards NVIDIA GeForce MX150 and Intel(R) UHD Graphics 620 was used for the experiments.

4.3 Input data organization for training and evaluation

Two approaches were used in the preparation of the image data before being presented to the network. In the first case, image data files were combined in a single file which contains all image data as a 4-axis tensor. In the second one, the image data files were organized on directories.

The first way allows a faster presentation of the input data to the network and is the ideal approach to train models on the original dataset, but is no longer feasible on augmented data because the size of the file would exceed the RAM limit of the computer used. The latter approach was preferred with data augmentation. It requires a batch generator which description is detailed in the section 4.5.

Organizing data files on directories is following described. When the data was primary stored as single data file, the data splitting was identical.

The data files were randomly split into 5 stratified partitions of similar size. The stratification was performed by the target class in order to preserve the same class distribution of the original data. Four partitions were used for 4-fold cross-validation and the fifth one was for testing. The organization of the sub-directories can be consulted in figure 4.1. Therefore, stratified 4-fold cross-validation was used during the experimentation phase for training, evaluating, comparing different model configurations and, in the end, for choosing the best model. In each fold, three partitions were used for training and the fourth was for model evaluation, so the validation partition was different at each fold, but two training partitions were shared between the folds. Therefore, one can obtain evaluation metrics for each fold and the mean value of the four models with the same hyperparameter configuration. This method was preferred because it guarantees lower variance than the hold-out method. The number of folds was set to 4 instead a higher value to obtain a lower computational cost and less biased evaluation metrics at the expense of the higher variance (Kohavi, 1995).

Since tuning a model is a repetitive process, there is some leakage of information from the validation partition into the model, even it is not directly trained on it, resulting in overfitting of the model to the validation set and optimistic performance metrics (F. Chollet & Allaire, 2018). To get a model with unbiased performance estimates, a test set partition was used only once to evaluate the best model selected among all ones trained and assessed during the cross-validation phase.

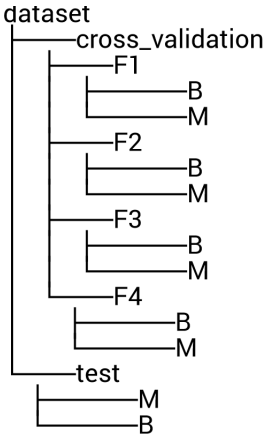


Figure 4.1: Image sub-directories for cross-validation and testing. There are 2 main sub-directories: cross-validation and test. The cross-validation sub-directory is additionally divided into four sub-directories, which are the partitions (F1 to F4). Each partition has the image files grouped by sub-directories of the target class: B - benign nodules and M - malignant nodules.

4.4 Exploratory analysis of the cross-validation dataset

An axial slice crossing approximately the centre of each PET volume (slice number 20) and, therefore, the centre of each nodule of the cross-validation dataset was selected and visualized as an image in order to obtain a characterization of the nodules. The test set was excluded to avoid information leakage during the construction of the models. Figures 4.2 and 4.3 represent the PET axial images of the cross-validation dataset grouped by the target class. The images are shown in a grey scale inverted, having a SUV window with a range between 0 and 5. Therefore, the image regions with a SUV equal or greater to 5 appear black.

The mean and standard deviation of the images (chosen axial slice) for the cross-validation dataset, grouped by target class is represented in figure 4.4.

4.5 Batch generator, shuffling and data normalization

When the dataset is organized as multiple image data files in a directory (and subdirectories), batches of images are extracted from the directory and presented to the model for training and / or evaluation. This is usually done in the R Keras package by image data batch generators, one for training and another for evaluation. However, the image data batch generator available in Keras is not valid both for volumetric images and for the file format where images are stored, requiring that a custom image batch generator for 3D CNN was created.

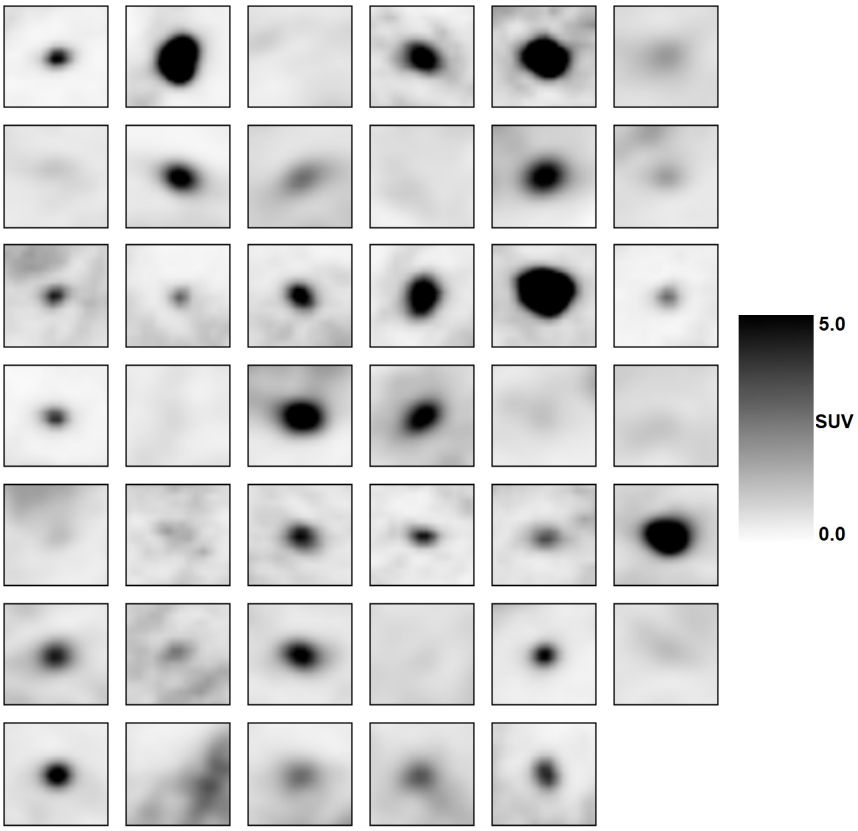


Figure 4.2: Malignant pulmonary nodules in PET images

An image data batch generator is a kind of iterator that extracts batches of images indefinitely from the selected directory. A Python iterator can be created from an R function with the *reticulate* package (Ushey et al., 2020).

The customized generator returns objects as lists. Each list has two elements, the former is a 5-axis tensor of input data and the latter is a vector encoding the target class for the batch of instances. The input tensor has the following shape [batch size, left-right, posterior-anterior, superior-inferior, channels] or [batch size, 40, 40, 40, 1]. It should be noted that each returned batch is a tensor with one more axis (corresponding to the channels) than the original tensor in order to be compatible with the network input.

Data were shuffled during the training which means that the order of the instances is randomly permuted at each epoch.

The input data underwent min-max normalization to the range [0, 1]. When a batch generator is used this occurs during during the batch generation. The normalization consists of

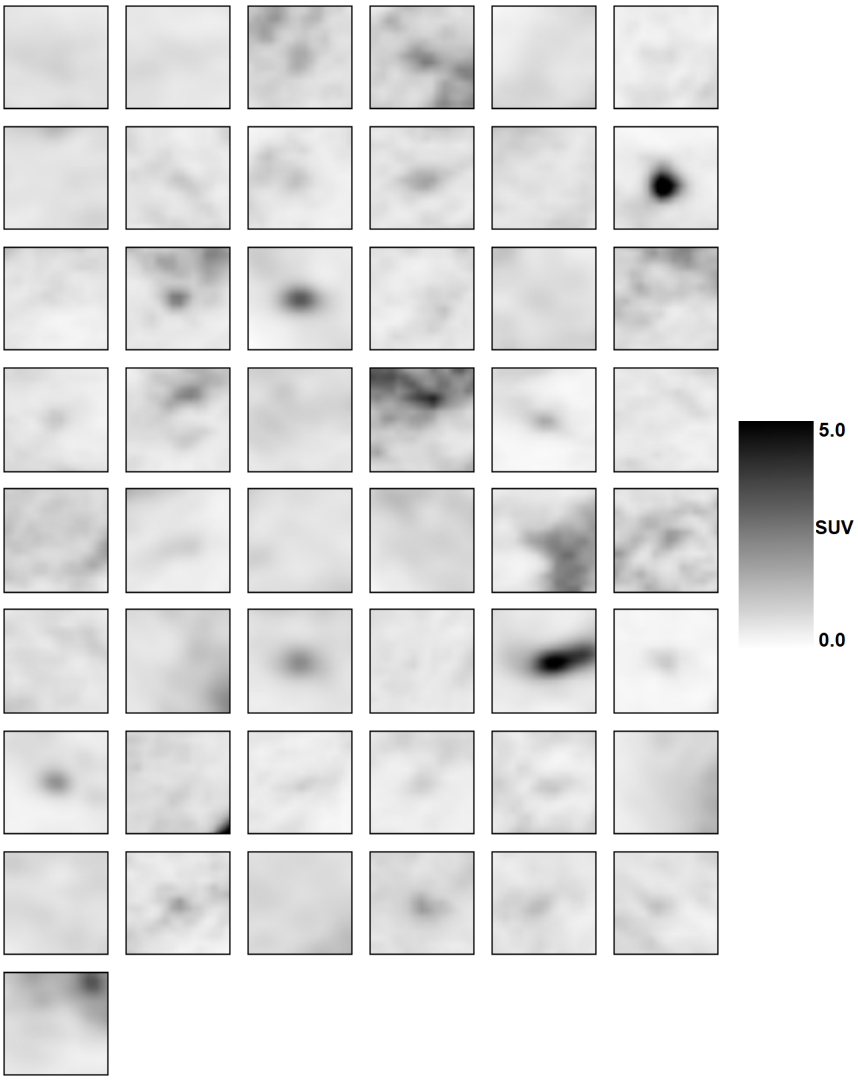


Figure 4.3: Benign pulmonary nodules in PET images

applying to each element a of the input tensor, the operation described in the equation 4.2:

$$a_{normalized} = \frac{a - min(\mathsf{X}_{\mathbb{S}(fold\ i)})}{max(\mathsf{X}_{\mathbb{S}(fold\ i)}) - min(\mathsf{X}_{\mathbb{S}(fold\ i)})} \tag{4.2}$$

$max(\mathsf{X}_{\mathbb{S}(fold\ i)})$ and $min(\mathsf{X}_{\mathbb{S}(fold\ i)})$ are the maximum and the minimum SUVs, respectively, of the training set for a given fold. The normalization procedure occurs separately in each fold. Training and validation sets of each fold were both normalized with values of the training set of the respective fold. The test set was normalized with the maximum and the minimum values of the training set of each fold as described in the section 4.9.

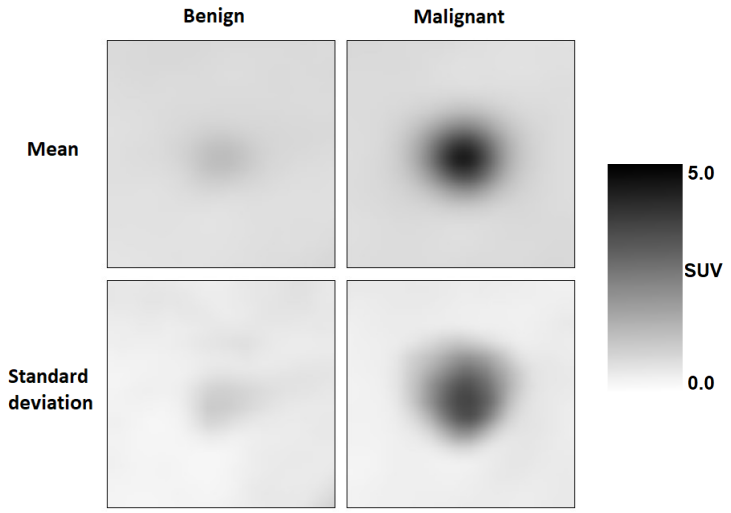


Figure 4.4: Mean and standard deviation of the cross-validation images (axial slice number 20) grouped by the target class.

4.6 Loss function

The binary cross-entropy loss was the loss function used for training. It measures the distance between the true target class and the prediction of the class. The smaller the value of cross-entropy, the closer the two probability distributions (Q. Wang et al., 2020). The binary cross-entropy loss is a differentiable non-convex function, being the main term of the empiric (and structural) risk. This is the amount to be minimized by an optimizer, enabling model parameters to be updated through back-propagation. It can be considered a combination of the cross-entropy loss and the sigmoid function. Sigmoid function can be seen as a particular case of the softmax function and was preferred because this is a binary classification problem (Q. Wang et al., 2020).

The output layer of the CNN has a single unit that outputs a probabilistic prediction of the reference class ($y = 1$) given the i^{th} input image, $P(y = 1|\mathsf{X_i}; \theta)$. Since the prediction of the class $y = 0$ is the complement of $P(y = 1|\mathsf{X_i}; \theta)$, a second neuron in this layer is not necessary. A probabilistic output is achieved with a transformation of the net input ($z^{(i)}_{y=1} = \mathbf{w}^T_{y=1} \cdot \boldsymbol{\gamma}^{(i)} + b_{y=1}$) of that neuron for the i^{th} image using the sigmoid function as activation function (Q. Wang et al., 2020):

$$p^{(i)}_{y=1} = \frac{1}{1 + e^{-z^{(i)}_{y=1}}} \tag{4.3}$$

The binary cross-entropy loss for the i$^{\text{th}}$ instance is

$$\mathcal{L}(\theta; (p^{(i)}, y^{(i)}) = -(y^{(i)}\, log_2(p^{(i)}_{y=1}) + (1 - y^{(i)})\, log_2(1 - p^{(i)}_{y=1})) \tag{4.4}$$

The amount to be minimized during the training is the following

$$R_{structural} = -\frac{1}{n}\sum_{i=1}^{n}\left[y^{(i)}\, log_2(p_{y=1}^{(i)}) + (1-y^{(i)})\, log_2(1-p_{y=1}^{(i)}) + R(...)\right] \tag{4.5}$$

The set of parameters θ that minimizes the structural risk is given by the equation 4.6 (Q. Wang et al., 2020). In practice, the algorithm receives batches of images, an estimate of the structural risk and respective gradient is computed at the batch level, and a correspondent update of parameters θ occurs. The structural risk at each epoch is the mean value over the batches.

$$\hat{\theta} = \arg\min_{\theta\in\mathbb{R}} R_{structural} \tag{4.6}$$

Binary cross-entropy is applied by defining the argument *loss* of the function *compile* of the Keras package with *binary_crossentropy*.

4.7 Optimizer

The Adam optimizer was chosen for training the CNNs. Described by Kingma and Ba (2017), it is an efficient algorithm for first-order gradient-based optimization that scales to large-scale and high-dimensional machine learning problems. It calculates individual adaptive learning rates or, in other words, effective step sizes for different parameters of the network from estimates of first (mean) and second (uncentered variance) moments of the gradients (g) which are determined from exponential moving averages from g and g^2, respectively. Equation 4.7 describes the update process of the parameters in the network:

$$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{4.7}$$

where $\hat{m}_t$ is the bias-corrected first-moment estimate:

$$\hat{m}_t = \frac{1-\beta_1}{1-\beta_1^t} \cdot \sum_{i=1}^{t} \beta_1^{t-i} \cdot g_i \tag{4.8}$$

and $\hat{v}_t$ is the bias-corrected second-moment estimate:

$$\hat{v}_t = \frac{1-\beta_2}{1-\beta_2^t} \cdot \sum_{i=1}^{t} \beta_2^{t-i} \cdot g_i^2 \tag{4.9}$$

t is the timestep, α is the step size, also called learning rate, β_1 and β_2 are exponential decay rates for the 1^{st} and 2^{sd} moment estimates, respectively, ϵ is a small quantity to prevent divisions by zero.

Adam optimizer has the following properties (Kingma & Ba, 2017):

- the magnitude of parameter updates are invariant to rescaling the gradient, which means if the gradient g is rescaled by a factor c, the term of parameters updating will be $-\alpha \cdot \frac{c \cdot \hat{m}_t}{\sqrt{c^2 \cdot \hat{v}_t} + \epsilon}$;
- The effective step size (Δ_t) is approximately bounded by the step size hyperparameter (α) because in most of the cases $\left| \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \right| \leq 1$, but more commonly ≈ 1;
- A stationary objective is not necessary;
- Sparse gradients do not prevent its use;
- It performs a form of step size annealing. The authors make an analogy of the $\frac{\hat{m}_t}{\sqrt{\hat{v}_t}}$ with the signal-to-noise ratio, where the smaller the ratio, the greater the uncertainty about the direction of the true gradient, and the smaller the Δ_t.

It has a low memory footprint, proper performance with sparse gradients and in on-line learning, and faster convergence than other optimizers (Kingma & Ba, 2017).

Adam optimizer was set in the algorithm by using the Keras function *optimizer_adam* as an argument for *compile* function. The default value was kept for all hyperparameters, excepting the step size argument, *lr*, being β_1 and β_2 the same as in original paper:

optimizer_adam(lr = {value}, beta_1 = 0.9, beta_2 = 0.999, epsilon = 1e-07, decay = 0, amsgrad = FALSE, clipnorm = NULL, clipvalue = NULL).

The step size hyperparameter was tuned by training the baseline model with values $\in \{0.1, 0.01, 0.001, 0.0001, 0.00001, 0.000001\}$, then followed by a more detailed search in the interval where the performance metric had a better result.

4.8 Criteria for selecting the best epoch during training

While typically the optimization of a CNN does not halt in a local minimum and the structural risk decreases throughout the optimization procedure, the validation loss saturates at some point and starts to increase again. The validation loss is an estimate of the functional risk, which cannot be directly used for model optimization, but can be used as a stopping criterion, ensuring that the minimization of the structural risk does not occur beyond the point of best generalization, obtaining a regularizing effect (Mahsereci et al., 2017). An early stopping criterion warrants the training is stopped before convergence to avoid overfitting (Prechelt, 2012).

In the experiments performed, the maximum number of training epochs was defined to be 100, after preliminary training runs. The stopping criteria were the following: the validation loss was the measure to be minimized; the patience was set to be 10, that is the number of training epochs with no additional decreasing of the validation loss after which the training will be

stopped; the training can also stop before a tolerance of 10 epochs if the epoch 100 is reached. The trained model in the epoch with the lowest validation loss was saved. This procedure was repeated for each fold of the 4-fold cross-validation.

The implementation of the early stopping was different depending on whether the original or the augmented data were used. The implementation was trivial with the original data. A list of two callbacks arguments was added to the function *fit()*: one for early stopping and the other for saving the best model.

In the augmented data, the training duration was set by a similar strategy to early stopping aiming a regularizing effect, but with a workaround imposed by a known issue in the *fit_generator* function[1] of the Keras package, at the moment the experiments were being performed, which precluded model evaluation at the end of each epoch of training when a customized image generator was used. Therefore, a fixed number of 100 epochs was set before training, then a version of the model (and its parameters) was saved in format .hdf5 for every epoch, passing the *callback_model_checkpoint* function was an argument of the *fit_generator*. At the end of the training run, each version of the model is evaluated in the validation set and obtained the validation loss. The best model version of a training run is defined according the criteria already described above. The best model was separately saved and the procedure was repeated for every fold.

At the end of training, four model versions were obtained during the 4-fold cross-validation, which have different values of parameters θ, but identical hyperparameter configuration.

4.9 Performance Metrics and Model Selection

The area under the ROC curve was the selected performance metric to evaluate models with different hyperparameter configuration, in the validation partition of each fold of the 4-fold-cross-validation, regarding to their ability to discriminate between benign and malignant pulmonary nodules. So, the value of the area under the ROC curve for a given model on each fold was recorded. Mean and standard deviation were determined. Models with different hyperparameter configuration were compared by their mean area under the ROC curve.

Models with different network architectures were trained and the hyperparameters were optimized. The best model of each network architecture was retrained and evaluated again by 10 iterations on the 4-fold-cross-validation, to evaluate the reproducibility of the training and, consequently, the model performance. The average performance metrics of the different models was compared and the best model was selected.

[1] https://github.com/rstudio/keras/issues/1090

The best model obtained in the cross-validation was evaluated in a separate test set to determine their generalization performance over unseen examples. The test set is a disjoint set regarding the training and the validation set.

The best model has indeed four versions, each one with the same hyperparameters configuration, but different value of parameters learned from the folds of the cross-validation. Therefore, an ensemble classifier was built from the four versions of the best model derived in the 4-fold-cross-validation, by averaging their output probabilities, weighted by the size of each training partition. This ensemble classifier was evaluated in the test set and the area under the ROC curve was determined.

The normalization of the test set was specific of the version of the best model and used information of the training set of the respective fold where each model version was trained.

Models which were trained on augmented data were also evaluated in original validation and test sets.

The area under the ROC curve was chosen as the main metric to evaluate the models because it is invariant to priory class probabilities (Bradley, 1997) and this dataset has a slight predominance of the positive class, it is decision threshold independent (Bradley, 1997), and exhibits more discriminancy and consistency than the accuracy (Jin Huang & Ling, 2005).

The area under the ROC curve is equivalent to the probability that a randomly picked negative example will have a smaller probability of belonging to the positive class than a randomly picked positive example (Jin Huang & Ling, 2005). The value of this metric was computed with the trapezoidal rule from non-parametric ROC curves using the function *auc* available in the package pROC (Robin et al., 2011).

The 95% confidence interval of the area under the ROC curve was determined for the test set according to the method described by DeLong (DeLong et al., 1988).

Additional performance metrics were computed for the ensemble CNN classifier in the test set: accuracy, sensitivity and specificity (Sokolova et al., 2006). The calculation of these metrics depends on a decision threshold. Instead of using the standard decision threshold of 0.5, an optimal decision threshold was determined. A decision threshold was determined for each version of the best model in the respective validation partition. The four decision thresholds were averaged (weighted mean by the size of each training partition) and the resulting threshold was applied to convert the output probabilities of the ensemble model into classes, in the test set. If the predicted probability was equal to or higher than the threshold, the nodule was classified as malignant; otherwise, it was classified as benign.

The optimal threshold was determined according to two different approaches. In the first one, the value of the optimal threshold was the posterior probability which maximizes the Youden index, which combines of sensibility and specificity, $Sensibility + Specificity - 1$

(López-Ratón et al., 2014).

In another scenario, the cost of a false negative was higher than the cost of a false positive. Therefore, a minimum sensitivity was set to 95% and the cut-off point which maximizes the specificity was searched. Consequences are different for false negatives and false positives, which could justify the use of this method for determining the threshold. Whereas a missed malignant lesion implies a delay in diagnosis and treatment, a false positive results in psychological distress for the patient and a non-necessary biopsy associated with discomfort and potential adverse effects.

The OptimalCutpoints package (López-Ratón et al., 2014) was used for determine the thresholds.

4.10 3D CNN models

The network architecture of a 3D CNN can be viewed as a generalization of a 2D CNN. This kind of architecture was proposed because the input data are volumes. 3D CNN differs from 2D CNN by performing 3D convolutions and 3D pooling operations. The mathematical formulation of a 3D convolution to obtain an output value γ at position (x, y, z) on the jth feature map in the ith 3D convolutional layer is as follow (Rao & Liu, 2020):

$$\gamma^i_{j,xyz} = \sigma\left(b^i_j + \sum_{m=1}^{M^{i-1}}\sum_{p=0}^{P^{i-1}}\sum_{q=0}^{Q^{i-1}}\sum_{r=0}^{R^{i-1}} w^i_{jm,pqr}\gamma^{i-1}_{m,(x+p)(y+q)(z+r)}\right) \tag{4.10}$$

where σ is the activation function, b^i_j is the common bias for the jth feature map, $w^i_{jm,pqr}$ is the (p, q, r)th value of the 3D kernel for the jth feature map at ith layer associated to the mth feature map in the (1 - i)th layer. P^i, Q^i and R^i are the dimensions of the kernel in the ith layer.

This section explains the procedures conducted to build CNNs with different architectures. Three principal groups of architectures were defined. When some detail of implementation is shared among groups of CNN architectures, a more detailed explanation of its use is provided the first time it appears in the text.

The resulting models were trained and evaluated primarily in the original dataset by cross-validation. The best algorithm of each architecture was also trained by 4-fold-cross-validation with augmented training sets which construction is described in section 4.11. Lastly, a pre-trained model in a large dataset was fine-tuned and evaluated in the original dataset. This is a 2D CNN and is described in a separate section (section 4.12).

A list of the experiments performed is provided in the section 4.14.

4.10.1 Stacked 3D CNN models

Network architecture

Different models were trained using several variations of a given network architecture which consists of convolutional and pooling layers alternately stacked and connected to a fully connected network.

Firstly, a model with high capacity was obtained, then hyperparameter tuning and regularization were performed in order to the match capacity of the model with the complexity of the task. The number of layers, the kernel size and the number of filters were treated as hyperparameters; as such, a extensive experimentation with different combinations of these hyperparameters was made. The necessary number of layers to obtain a model with a high capacity was determined by starting from the simplest model and adding successive layers until the model overfits, being the number of layers finally tuned. Therefore, most of the training was focused on models with three or four pairs of convolutional and pooling layers.

Regarding the kernel size, models were trained either using a fixed or a variable kernel size in the convolutional layers. The following kernel sizes were evaluated: [3, 3, 3], [4, 4, 4] and [5, 5, 5]. The kernel stride was 1 in the convolutional layers. Padding was not used unless two convolutional layers were placed in sequence.

The number of filters was the same or increasing across the convolutional base, by duplicating or increasing 50% at each convolutional layer, having the first layer 4, 8 or 16 filters.

The max pooling was applied to pooling layers. The kernel size was set to [2,2,2], having a stride of 2 and no padding.

A fully connected network with two or three layers was connected to the top of the convolutional base. This network receives a one-dimensional tensor obtained by one of three ways: either performing a proper combination of successive convolutional and pooling layers, flattening or applying global average pooling on the output of the last layer of the convolutional base.

Activation function

The same activation function was chosen for the units of the convolutional base and the fully connected network (excluding the output layer).

Experiments were firstly conducted with Rectified Linear Unit (ReLU). ReLU was later replaced by Leaky ReLU and additional experiments were performed. Leaky ReLU has the advantage of allowing a small, non-zero gradient when an unit is not active and thus prevents ”dying ReLU” (Gu et al., 2018; Lu et al., 2020).

Weight initialization

Part of the experiments was performed with the weight initialization scheme proposed by Glorot and Bengio (2010), commonly known as Xavier initialization. Then, the initialization scheme was changed to He initialization (He et al., 2015) and additional experiments were conducted.

The Xavier initialization (Glorot & Bengio, 2010) consists of initializing the biases with zero and the weights with a uniform distribution defined in a neuron-specific interval as follows:

$$W \sim U\left(-\sqrt{\frac{6}{n_j + n_{j+1}}}, \sqrt{\frac{6}{n_j + n_{j+1}}}\right) \tag{4.11}$$

where n_j is the number of incoming connections and n_{j+1} is the number of outgoing connections. This scheme of initialization introduces a normalization factor that maintains approximately constant the activation variance and back-propagated gradients variance along the different layers of the network, preventing problems of vanishing or exploding gradients, which can slow down or even completely hinder the convergence process (Glorot & Bengio, 2010).

Although, Xavier initialization has been popularized, it was originally derived for sigmoid, hyperbolic tangent and softsign activation functions, having assumptions that are not valid for rectifiers (Glorot & Bengio, 2010).

A new scheme of initialization was meanwhile developed to specifically address the rectifiers, becoming known as He initialization (He et al., 2015). The biases are initialized with zero and the weights according to equation 4.12.

$$W = N \sim \left(mean = 0, sd = \sqrt{\frac{2}{n_j}}\right) \tag{4.12}$$

Xavier and He initialization are implemented by default in Keras package (Allaire & Chollet, 2019). The models were recompiled every new fold of cross-validation so that the training is started with random parameters according to the scheme selected.

Batch size

Full batch learning was preferred when the original dataset when used. Mini-batch learning with batch sizes of 8 or 16 was used with augmented data.

Regularization procedures

Besides tuning procedures to find an optimally sized network architecture and the early stopping, L2 regularization and dropout were applied on some models. L2 regularization consists of adding a regularizer term to the empiric risk that penalizes the model complexity. That

term corresponds to the squared L2 norm of the weights, $\|\mathbf{w}\|_2^2$. The relative contribution of the penalty is given by the hyperparameter λ. As consequence, before each weights update, the size of the weights is shrunk and hypothesis space is constrained (Goodfellow et al., 2016).

In some models, dropout was applied to the fully connect network of the CNN. Dropout prevents the network, at least in this part of the CNN, from becoming too dependent on any one of neurons. Units are randomly removed with a probability of p, along with their incoming and outgoing connections, on each training step (Srivastava et al., 2014).

Another type of regularization is data augmentation. Details of creation an augmented dataset are in the section 4.11. More details about the stacked 3D CNN models trained can be found in section 4.14.

4.10.2 VGG-like models

Models with symmetrical convolutions

VGG networks (Simonyan & Zisserman, 2015) have a simple and homogeneous architecture based on stacked convolutional layers, some of them followed by a max-pooling layer, and ending with a fully connected network on the top. The network width gradually increases whereas the resolution of the feature maps decreases. The depth of the network and the efficient use of 3×3 convolutions are prominent features that contribute to their high classification performance in several domains.

The original 2D architecture of the VGGNet (Simonyan & Zisserman, 2015) is generalized in the current research for a 3D network and its depth and width is adapted to the type of problem and size of the training set.

Thus, networks with 7 to 9 layers of parameters were trained, instead of the 11 to 19 layers of parameters of the original publication (Simonyan & Zisserman, 2015). Similarly, every convolutional layer has a very small kernel size ($3 \times 3 \times 3$). Blocks of two or three stacked $3 \times 3 \times 3$ convolutional layers without max-pooling between them are used rather than a single layer with a higher kernel size ($5 \times 5 \times 5$ or $7 \times 7 \times 7$, respectively), being a way to achieve the same effective receptive field with a lower number of parameters. This operation is the same as that performed in a 2D CNN, but now the receptive field has one more axis. The decomposition of a convolutional layer with a larger kernel size in several ones with a smaller kernel size imposes a greater reduction in the number of parameters in a 3D than in a 2D network, and therefore a greater regularizing effect.

Assuming an equal number of input and output characteristic maps, represented by m and ignoring the number of biases, for a given block of n stacked $k \times k \times k$ convolutional layers, the number of weights can be determined by $n(k^3m^2)$ (Simonyan & Zisserman, 2015).

So, a stack of two $3 \times 3 \times 3$ convolutional layers has $2(3^3m^2) = 54m^2$ weights, whereas the equivalent single $5 \times 5 \times 5$ convolutional layer would have $5^3m^2 = 125m^2$, which represents a reduction of 57% of the number of weights. A similar implementation in a 2D CNN only imposes a reduction of 28% of the number of weights. Replacing a $7 \times 7 \times 7$ convolutional layer by the equivalent stack of three $3 \times 3 \times 3$ convolutional layers leads to a reduction of the number of weights in 76%, whereas in a 2D CNN, that would lead to a reduction of 45%. Furthermore, the addition of convolutional layers implies the incorporation of more non-linear activation functions which could make the decision function more discriminative (Simonyan & Zisserman, 2015).

Similarly to the VGGNet (Simonyan & Zisserman, 2015), padding is applied to the convolutional layers for preserving the spatial resolution of the feature maps. Spatial resolution only decreases at each max-pooling layer. Unlike in the original publication (Simonyan & Zisserman, 2015), overlapping max-pooling is applied since preliminary experiments have showed a slight improvement with this type of pooling compared to non-overlapping max pooling. The established hyperparameters for the operation were pool size of $3 \times 3 \times 3$, strides of 2 and padding. Overlapping pooling is a feature of the Alexnet model (Krizhevsky et al., 2012)

The output of the convolutional base is either flattened or suffers global average pooling and serves as input for a fully connected network with 2- or 3- layers. In some models there is only the output layer.

The activation function and the weight initialization scheme were those chosen for the previously described Stacked 3D CNNs (Leaky ReLU and He initialization, respectively) because of their demonstrated advantages. Besides early stopping, in some models regularization was performed with L2 weight regularization, dropout and/or data augmentation.

Full batch training was preferred on the original dataset due to its small size. Training with a batch size of 8 was performed with augmented data.

Models with asymmetrical convolutions

A variation of the models described behind has received a further inspiration from the Inception-v3 (Szegedy et al., 2015). Excepting the first convolutional layer, each one of the remaining $3 \times 3 \times 3$ convolutional layers was replaced by two stacked asymmetrical convolutional layers having a kernel size of $3 \times 3 \times 1$ and $1 \times 1 \times 3$, respectively. Two stacked asymmetrical convolutional layers have the same effective receptive field as a symmetrical $3 \times 3 \times 3$ convolutional layer, but are computationally much more efficient because the number of weights decreases by around 44% or 56%, as assuming the number of feature maps increases by a factor of 2 or is kept constant, respectively, and ignoring the biases. The relative reduction of the number of

weights ($Relative\ change(w_{i1+2}, w_i)$) by factorization of a symmetrical convolutional layer i into two asymmetrical layers is determined for the case where the number of output feature maps is double of the input feature maps in the first of a pair of asymmetrical layers. Being the number of weights of a symmetrical convolutional layer, $w_i = k^3 \cdot m_{in} \cdot 2m_{in}$, and w_{i1} and w_{i2} the number of weights of the asymmetrical convolutional layers, then

$$w_{i1} = k \cdot k \cdot 1 \cdot m_{in} \cdot 2m_{in} \tag{4.13}$$

$$w_{i2} = 1 \cdot 1 \cdot k \cdot (2m_{in})^2 \tag{4.14}$$

$$w_{i1+2} = 2k \cdot m_{in}^2 \, (k + 2) \tag{4.15}$$

$$Relative\ change(w_{i1+2}, w_i) = 2k^{-2} + k^{-1} - 1 \tag{4.16}$$

The expansion of the feature maps occurs in the first convolutional layer after a pooling layer. In the following convolutional layers of the same convolution block, the number of the filters is the same. So, the relative reduction of the number of weights achieved by implementing asymmetrical convolutional layers can be determined as following in those layers:

$$w_{i1} = k \cdot k \cdot 1 \cdot m^2 \tag{4.17}$$

$$w_{i2} = 1 \cdot 1 \cdot k \cdot m^2 \tag{4.18}$$

$$w_{i1+2} = m^2 \left(k^2 + k\right) \tag{4.19}$$

$$Relative\ change(w_{i1+2}, w_i) = k^{-2} + k^{-1} - 1 \tag{4.20}$$

The output of the convolutional base was flattened before the output layer. There is no fully connected network. The remaining aspects of the training are similar to those described for models with symmetrical convolutions.

More details about the VGG-like models can be found in the section 4.14.

4.10.3 Inception-v2-like models

The introduction of inception modules in a CNN allows approximating an optimal local sparse structure from dense components within the convolutional base, and by this way, building a computationally more efficient network (Szegedy et al., 2014). Greater computational efficiency leaves resources available for escalation of the network width and depth according to the needs of each problem (Szegedy et al., 2014).

Inception modules consist of blocks of several convolutional layers with different filter size and a pooling layer that receive the same input, propagate the information in parallel and concatenate the output before passing it to the next layer. Much of the computational efficiency is achieved by using 1×1 convolutions to compute reductions of the number of feature maps before expensive 3×3 and 5×5 convolutions as the Inception modules used in GoogLeNet (Szegedy et al., 2014). Even more efficient versions of Inception modules were proposed for the Inception-v2 and Inception-v3 (Szegedy et al., 2015) which replace convolutions with larger filter size by factorizing them in stacked 3×3 convolutions or in stacked asymmetrical convolutions, ensuring the same receptive field. Inception networks consist of several stacked Inception modules. The reduction of the feature map size is performed by an Inception module having layers with stride of 2, rather than a max-pooling layer. In this reduction module, the feature map size is reduced while their number is expanded, being an efficient way of feature map size reduction and of avoiding a bottleneck representational.

Inception-v2-like models use a number of Inception building blocks adapted to the type of problem and size of the training set. The Inception architecture is generalized in the current problem for a 3D network since the inputs are volumes.

The Inception model proposed for this problem has four standard Inception modules as represented in figures 4.5 and 4.6, wherein after the second and the fourth there is a modified Inception module that reduces the feature map size (figures 4.7 and 4.8). Inception modules are placed after a convolutional and a max-pooling layer. The architecture of the Inception modules is similar to that described in the (Szegedy et al., 2015). In the two first modules $5 \times 5 \times 5$ convolutions are factorized in two stacked $3 \times 3 \times 3$ convolutions. In the two last modules, the $3 \times 3 \times 3$ convolutions are additionally factorized in stacks of $3 \times 3 \times 1$ and $1 \times 1 \times 3$ convolutions. The output of the last reduction module is converted in a vector by global average pooling and then is the input for the output layer of the network.

Weights were initialized according to the method described by (He et al., 2015). Leaky ReLU was the activation function. The batch size was 16 in the original dataset and 8 in the augmented dataset. L2 weight regularization was applied in all layers with parameters.

More details about the Inception models trained can be found in the section 4.14.

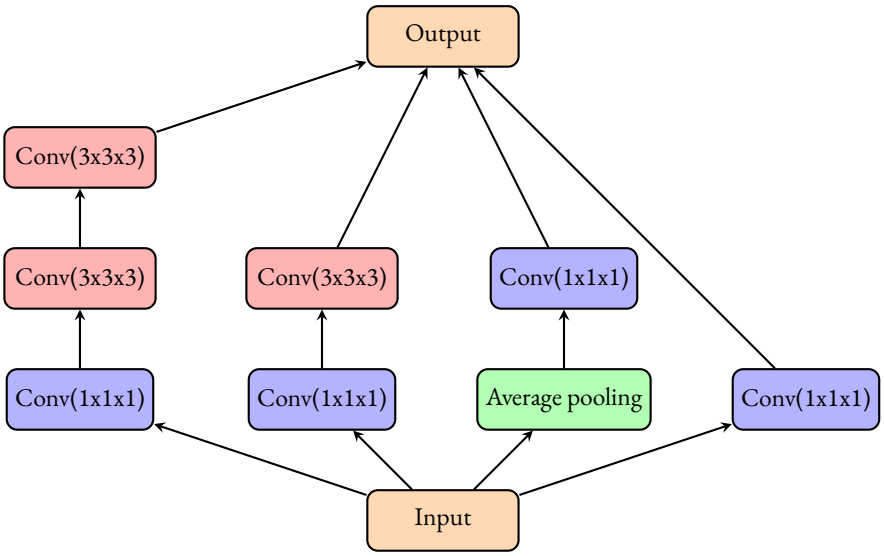


Figure 4.5: 3D Inception module 1

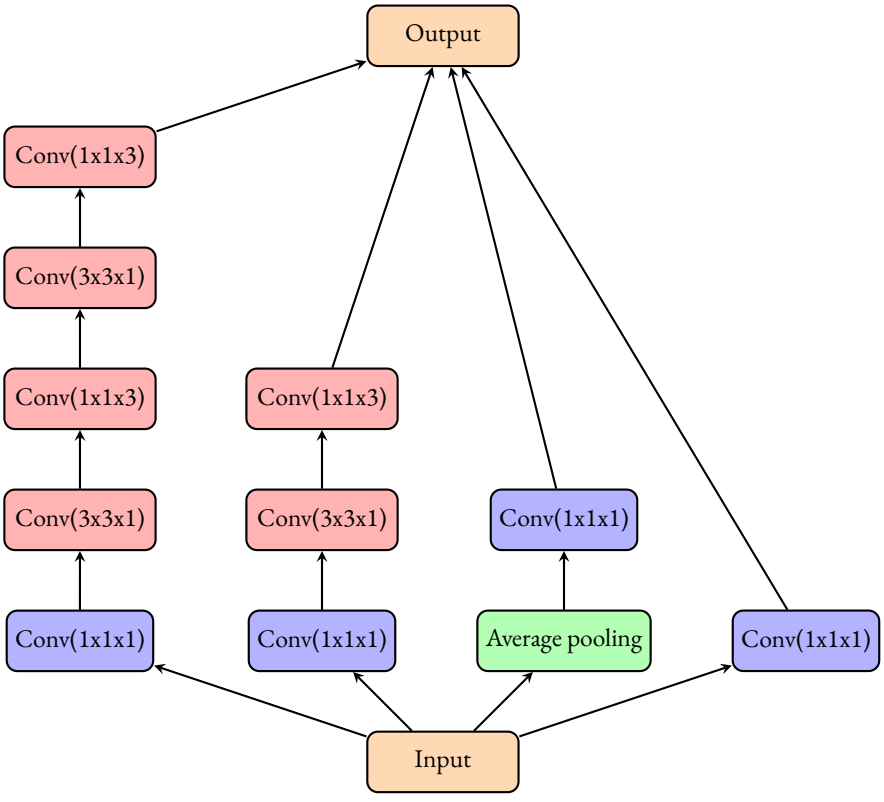


Figure 4.6: 3D Inception module 2

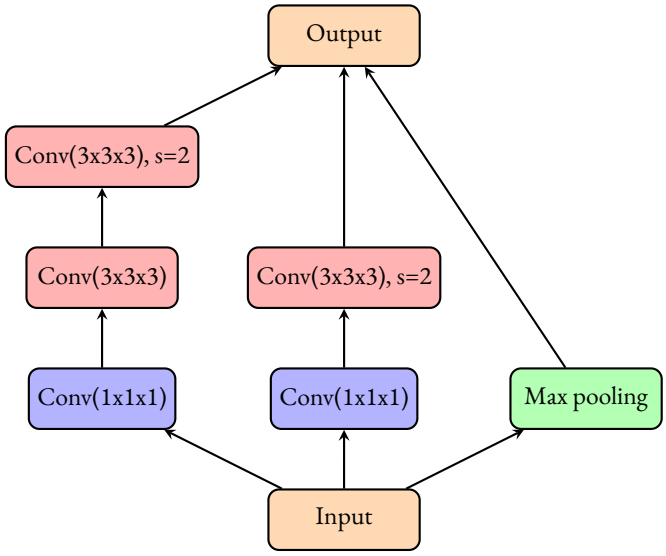


Figure 4.7: 3D Inception reduction module 1; s=2 means strides of (2, 2, 2)

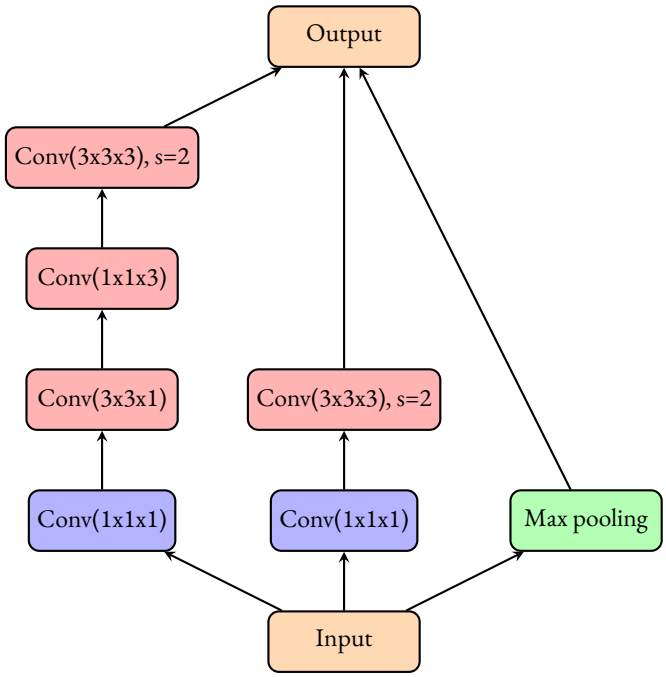


Figure 4.8: 3D Inception reduction module 2; s=2 means strides of (2, 2, 2)

4.11 Data augmentation and class balancing

Offline data augmentation was performed with geometric transformations using as a starting point the 4 partitions data of the original dataset, previously created for cross-validation. Data augmentation was done independently in each partition. Translations, rotations and noise injection were applied to the original images. Test set was not augmented. The augmented dataset comprises original and augmented images, having around 4900 images. O size of the augmented dataset was determined by the computational resources available for training models in a larger dataset.

During the cross-validation, there was an augmented training set on each fold where the models were trained. The evaluation of the models occurred in the validation set of the original dataset.

The augmentation factor was class-specific in order to perform class balancing, being the minority class augmented to a greater extent than the majority class.

4.11.1 Translations

The translated images were created by augmenting the images of the minority class (malignant) by a factor of 20, and those of the majority class (benign) by a factor of 16.

The translations were random shifts between -10 and 10 pixels on any of the 3 axes. A maximum amplitude of 10 pixels (15 mm) was chosen to ensure that the nodules are not moved out of the tensor and the label is preserved, once the maximum diameter of the nodules is 30 mm and the cubic image tensors have 60 mm side. Background voxels were filled with 0.

Let a voxel be in the position (x, y, z) of an image tensor represented as a column vector, and a matrix for three-dimensional transformations. Both represented in a homogeneous coordinate system. The translation operation can be theoretically represented as following (Comninos, 2006):

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{4.21}$$

t_x, t_y and t_z are the values of the translations in the x-, y- and z-axis, respectively.

Instead of implementing the operations of three-dimensional translations in R from scratch, the function *translate* from package EBImage (Pau et al., 2010) was used which offers the possibility to carry out two-dimensional transformations. So, to achieve three-dimensional transformations with two-dimensional transformation matrices a set of adaptations was made by

building a function which, firstly, iterates slice by slice the x- and y-axes translations along the z-axis and then iterates z-axis translations slice by slice along the x-axis over the previous tensor.

4.11.2 Rotations

Rotated images were created by augmenting the images of the minority class (malignant) by a factor of 21, being a factor of 7 per axis, and the those of the majority class (benign) by a factor of 15, being 5 per axis.

Each example was augmented by applying separately random rotations around the x-, y- or z-axis so that each original example yields augmented examples with different rotation axes, but each new augmented example has a rotation applied only around a given axis. The rotations were performed randomly between -45° and 45°.

After the spatial transformation of the coordinates of the voxels, an intensity interpolation with a bilinear interpolator was applied. Background voxels were filled with 0.

Following are represented the rotation matrices in a right-handed homogeneous coordinate system (Comninos, 2006):

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(v) & -\sin(v) & 0 \\ 0 & \sin(v) & \cos(v) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad Around\,the\,x-axis \tag{4.22}$$

$$\begin{bmatrix} \cos(v) & 0 & \sin(v) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(v) & 0 & \cos(v) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad Around\,the\,y-axis \tag{4.23}$$

$$\begin{bmatrix} \cos(v) & -\sin(v) & 0 & 0 \\ \sin(v) & \cos(v) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad Around\,the\,z-axis \tag{4.24}$$

As the rotations should be around an axis which runs through the centre of the tensor not the origin, and the nodule is in the centre of the tensor. Thus, the rotations are not around the axes of the coordinate system but indeed around parallel axes of them. So a composed operation should be applied which translates the nodule so that rotation axis of the coordinate system coincides with the parallel coordinate axis, then rotates the tensor around that axis and finally translates again the nodule so that the rotation axis is moved back to its original position

(Comninos, 2006). That operation can be describes thus for the voxel P of an image tensor (Comninos, 2006):

$$P_{rotated} = T^{-1} * R * T * P \tag{4.25}$$

Three-dimensional rotations of the image tensors were performed by iterating two-dimensional rotations along the perpendicular slices to each rotation axis. A new function was created to do this, which integrates the *rotate* function from the EBImage package (Pau et al., 2010).

4.11.3 Gaussian noise injection

Gaussian noise injection was applied to create augmented images. The minority class was augmented with this method by a factor of 20, while the majority class was augmented by a factor of 16. The Gaussian noise injected had mean 0 and three different values of standard-deviation (0.1, 0.3 and 0.5). The distribution of the standard deviation in the minority class has the following 0.1 (7) and 0.3 (7) and 0.5 (6), while for the majority class was 0.1 (6) and 0.3 (5) and 0.5 (5).

The noise injection process is explained next. A tensor of Gaussian noise with the same shape of the original images was created. This tensor was filled with random values with a normal distribution with mean 0 and the standard deviation specified using the function *rnorm*. Lastly, this tensor was added to the original image tensor and an input image tensor with Gaussian noise injection is created. The decision of adding Gaussian noise was made because the noise in PET images can be modeled with a Gaussian distribution (Teymurazyan et al., 2013), so the different augmented images simulate PET images with different levels of noise.

4.12 Transfer learning

Part of the convolutional base of the Resnet50 (He et al., 2016), pre-trained in the Imagenet dataset, was used as a feature extractor. A fully connected network with two layers was added to its top and initialized with the He initialization. The ReLU activation was added to the former layer of the fully connected network and sigmoid activation was added to the latter. Because the dataset of the current problem is quite different from Imagenet dataset, only the earliers layers of Resnet50 were used (until *conv3_block1_out*). Additionally, a few of the top layers (from *conv3_block1_1_conv*) were fine-tuned with a very low learning rate. More details about the training of this network can be found in table 4.1.

It is noteworthy that the Resnet50 is not a 3D CNN, but a 2D CNN. The convolutional base only admits exactly 3 inputs channels, and the two first axes should be no smaller than 32. Once the input of PET images is a tensor with [40, 40, 40], only the 3 central slices (19, 20 and

21) of the third axis were used, being each one stored in a different channel. As the nodules are at the centre of each volume, they are necessarily intersected by the selected slices.

Resnet50 pre-trained in Imagenet is available in Keras through the function *application_resnet50* (Allaire & Chollet, 2019).

4.13 Reproducibility

In order to perform reproducible experiments, seeds were set for all sources of randomness throughout the pipeline. This is important to warrant that the difference among different models are not random in nature in the dataset. Additionally, it allows reproduction of results when a model is retrained under identical conditions. In spite of the efforts made, a source of non-determinism persisted which was attributed to asynchronous floating point operations on Tensorflow GPU [2]. In order to deal with the non-deterministic behavior of Tensorflow, the best models were retrained 10 times under identical conditions as described in the section 4.9.

4.14 List of experiments on cross-validation

Table B.1 (appendix B) presents a list of models trained with different architectures in either the original or the augmented training set. It also includes a pre-trained model. A description of several hyperparameters is provided, such as batch size, type of layers, number of filters, kernel size, activation function, weight initialization, learning rate and regularizers. Other characteristics of the machine learning task remain unchanged and were already described in the previous sections.

Although the list presented is intended to be representative of the experiments performed, this is not an exhaustive list, so it does not include all the performed experiments.

Table 4.1 shows a selected list of models from table B.1, where is presented the best model for each architecture and type of dataset (original or augmented).

4.15 Paired comparison between the CNN model and the SUVmax

The ROC curve of the SUVmax in the nodule was derived from the test set. This measure is widely used in clinical practice to assist in the interpretation of PET/CT images. As the SUVmax is well validated and its use consolidated, a hypothesis test was performed to infer about a

[2] https://developer.nvidia.com/gtc/2019/video/s9911

Table 4.1: Main models trained by cross-validation

Type	**Batch size**	**Architecture**	σ	**Initialization**	**LR**	**Regularizer**
Stacked 3D CNN	68	conv(8,3,3,3) + mpool + conv(16,3,3,3) + mpool + conv(32,3,3,3) + mpool + conv(64,3,3,3) + flatten + fcn(32,16,1)	LeakyReLU (α=0.3)	He	0.001	L2(0.00098)
Stacked 3D CNN	8	conv(8,3,3,3) + mpool + conv(16,3,3,3) + mpool + conv(32,3,3,3) + mpool + conv(64,3,3,3) + flatten + fcn(32,16,1)	LeakyReLU (α=0.3)	He	0.0001	L2(0.03) and data augmentation
VGG-like	68	conv(8,3,3,3) + overlap mpool + conv(16,3,3,3) + conv(16,3,3,3) + overlap mpool + conv(32,3,3,3) + conv(32,3,3,3) + conv(32,3,3,3) + overlap mpool + flatten + fcn(1)	LeakyReLU (α=0.3)	He	0.0005	L2(0.002)
VGG-like	8	conv(8,3,3,3) + overlap mpool + conv(16,3,3,3) + conv(16,3,3,3) + overlap mpool + conv(32,3,3,3) + conv(32,3,3,3) + conv(32,3,3,3) + overlap mpool + flatten + fcn(1)	LeakyReLU (α=0.3)	He	0.0001	L2(0.06) and data augmentation
Inception-v2-like	68	conv(8,3,3,3) + mpool + Inception + Inception + Reduction + Inception + Inception + Reduction + gap + fcn(1)	LeakyReLU (α=0.3)	He	0.0005	L2(0.0006)
Inception-v2-like	8	conv(8,3,3,3) + mpool + Inception + Inception + Reduction + Inception + Inception + Reduction + gap + fcn(1)	LeakyReLU (α=0.3)	He	0.0001	L2(0.04) and data augmentation
ResNet50 pre-trained	68	ResNet50 (base) + gap + fcn (8,1)	ReLU	He	5×10^{-7}	Transfer learning and dropout(0.5)

possible difference in the area under the ROC curve between the CNN ensemble model and the SUVmax in the population. The null hypothesis is that the areas under the curve of the CNN ensemble model and the SUVmax are equals in the population. The alternative hypothesis is that their areas under the curves are different. If the null hypothesis is rejected and the AUC ROC curve is higher in the test set for the CNN ensemble model than for the SUVmax, this argues for the superiority of the CNN ensemble model. The type I error (α) was predefined as 0.05.

$$H_0: \text{AUC ROC}_{\text{CNN}} = \text{AUC ROC}_{\text{SUVmax}}$$
$$H_1: \text{AUC ROC}_{\text{CNN}} \neq \text{AUC ROC}_{\text{SUVmax}}$$

The non-parametric test developed by DeLong et al. (1988) which makes a paired comparison of the area under the ROC curves, is applied if the area of one ROC curve is uniformly higher that the other across all operating points, that is, the curves do not cross each other; otherwise a hypothesis test based on the ROC shape proposed by Venkatraman and Begg (1996) is applied.

The ROC curves and the hypothesis testing were performed using the pROC package (Robin et al., 2011).

It should be emphasized that as this is a secondary objective of the dissertation, the test set was not sized to ensure the desirable statistical power for the conducted statistical test; as such the results should be interpreted with caution (Button et al., 2013) and considered as preliminary, deserving further confirmation by well-powered studies.

Chapter 5

Results

This chapter shows the results of the trained models in the cross-validation and the results of the best model evaluated on the test set.

5.1 4-fold cross-validation

This section presents the results for the best model obtained from each architecture and type of dataset, using cross-validation (table 4.1). A more complete list of results about the trained models can be found in tables B.2 (validation loss) and B.3 (area under the ROC curve) of appendix B.

5.1.1 Early stopping

The epoch with the minimum validation loss obtained by early training stopping is shown in the table 5.1 for each fold of the main trained models. The model obtained in that epoch was saved.

5.1.2 Area under the ROC curve

Table 5.2 shows the area under the ROC curve for the main models, using cross-validation. This measure is presented for each fold as well as the mean and standard deviation of all folds.

The mean area under the ROC curve has ranged from 0.8035 (model 28) to 0.8864 (model 12). The best performance was reached by a 3D stacked CNN model in the original dataset.

Regardless of the type of model, it was consistently found that the fold one yielded a performance lower than the remaining folds, which corresponds to models trained in partitions two, three and four, and evaluated in partition one of the cross-validation data. Additionally,

Table 5.1: Minimum validation loss and respective epoch by fold for the main trained models obtained by early stopping. V. Loss - Validation loss; Augm. - Data Augmentation; No. - Number of the model according to the order on the table B.2.

	F1		F2		F3		F4	
Model (No.)	**Epoch**	**V. Loss**	**Epoch**	**V. Loss**	**Epoch**	**V. Loss**	**Epoch**	**V. Loss**
Stacked 3D CNN (12)	1	1.0159	13	0.7510	10	0.7286	32	0.6241
Stacked 3D CNN + Augm. (22)	78	0.8269	48	0.7109	23	0.6632	100	0.6013
VGG-like (19)	5	1.2068	19	0.9084	12	0.8852	33	0.8948
VGG-like + Augm. (25)	34	0.8372	19	0.6767	15	0.5610	64	0.4629
Inception-v2-like (21)	10	1.1227	28	0.8556	18	0.8627	41	0.8205
Inception-v2-like + Augm. (27)	34	0.8392	29	0.7389	71	0.6039	48	0.5697
ResNet-50 pre-trained (28)	80	0.6910	49	0.6931	43	0.7001	66	0.6922

models trained on the original dataset performed better than those trained on the augmented dataset.

The results of the retraining and evaluation over 10 iterations for the best model of each network architecture (models 12, 19 and 21) for evaluating the reproducibility and selecting the best model can be consulted in tables B.4, B.5 and B.6 of appendix B. ResNet-50 was not retrained because its performance was much lower than other architectures. The mean area under the ROC curve of the 3D stacked CNN model (model 12), VGG-like model (model 19) and Inception-v2-like model (model 21) was 0.8822, 0.8760 and 0.8690, respectively.

The 3D stacked CNN model (model 12) showed consistently the best performance on the iterated cross-validation. A random version (one of the 10 iterations) of this model was selected to be evaluated on the test set. It is out of scope to prove the superiority of a model in relation to the others in a population of individuals with pulmonary nodules. Therefore, hypothesis testing was not performed. The superiority of the 3D stacked CNN model (model 12) should be interpreted as strictly concerning the cross-validation dataset.

A visual representation of the best model is fiven in figure 5.2.

Table 5.2: Area under the ROC curve on cross-validation for the main models

	AUC ROC curve					
Model (No.)	**F1**	**F2**	**F3**	**F4**	**Mean**	**SD**
Stacked 3D CNN (12)	0.7917	0.9000	0.8750	0.9790	0.8864	0.0772
Stacked 3D CNN + Augm. (22)	0.7333	0.8500	0.8417	0.9371	0.8405	0.0835
VGG-like (19)	0.7333	0.9250	0.9417	0.9161	0.8790	0.0977
VGG-like + Augm. (25)	0.7000	0.7833	0.8667	0.9301	0.8200	0.1001
Inception-v2-like (21)	0.7250	0.9083	0.8917	0.9650	0.8725	0.1032
Inception-v2-like + Augm. (27)	0.7333	0.8333	0.8417	0.8741	0.8206	0.0608
ResNet-50 pre-trained (28)	0.7167	0.8083	0.8500	0.7203	0.7738	0.0662

5.1.3 Learning curves

The figure 5.1 shows the evolution of training for the best model (Stacked 3D CNN) during the cross-validation. For each fold, the top chart shows the training loss and the validation loss. The bottom chart presents the training and the validation accuracy. In the first fold, the model suffers overfitting right after the first epoch. In the remaining folds, the overfitting starts later.

5.2 Evaluation on the test set

5.2.1 ROC curve

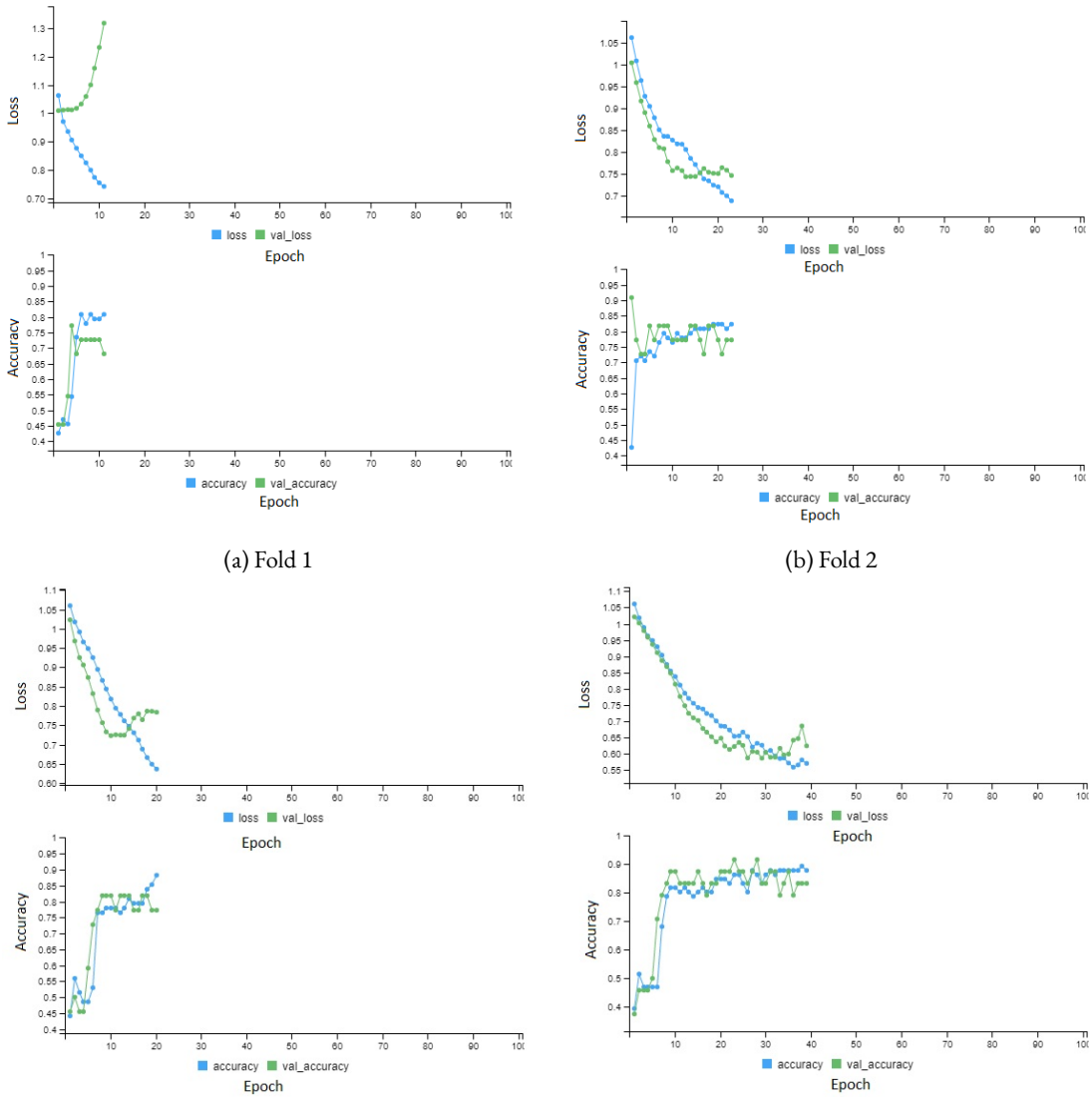
Since the best model has actually four versions (one by fold of cross-validation), an ensemble classifier was built and evaluated on test set. That ensemble classifier obtained an area under the ROC curve of 0.8385, 95% CI: 0.6455-1.0000 (DeLong) on the test set. The ROC curve is represented in figure 5.3.

5.2.2 Performance metrics

For a decision threshold that maximizes the Youden index

A decision threshold of 0.5039 was obtained on the cross-validation for the ensemble model based on the criterion which maximizes the Youden index.

The confusion matrix for the test set is presented in the table 5.3. From 23 PET images of pulmonary nodules, 4 were true positives, 13 were true negatives, 6 were false negatives. There was no false positive. Therefore, the ensemble model had a sensibility of 40.0%, a specificity of



(a) Fold 1

(b) Fold 2

(c) Fold 3

(d) Fold 4

Figure 5.1: Learning curves of the best model (Stacked 3D CNN) on cross-validation.

100.0% and an accuracy of 73.91% for this threshold, on test set.

		Actual class	
		Malignant	Benign
Predicted class	Malignant	4	0
	Benign	6	13

Table 5.3: Confusion matrix for the pulmonary nodules classification on test set by the ensemble model based on a threshold that maximizes the Youden index

.

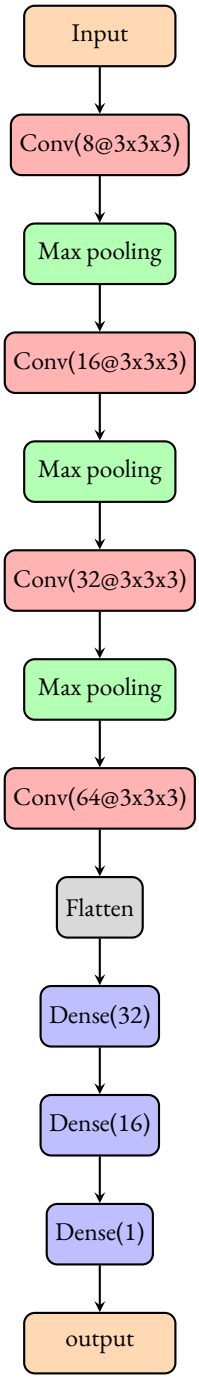


Figure 5.2: Network architecture of the best model. Each convolutional layer and the two first dense layers has a leaky ReLU activation function. Convolutions are performed with strides of (1, 1, 1) and no padding. Max-pooling layers have pool size of (2, 2, 2) and strides of (2, 2, 2).

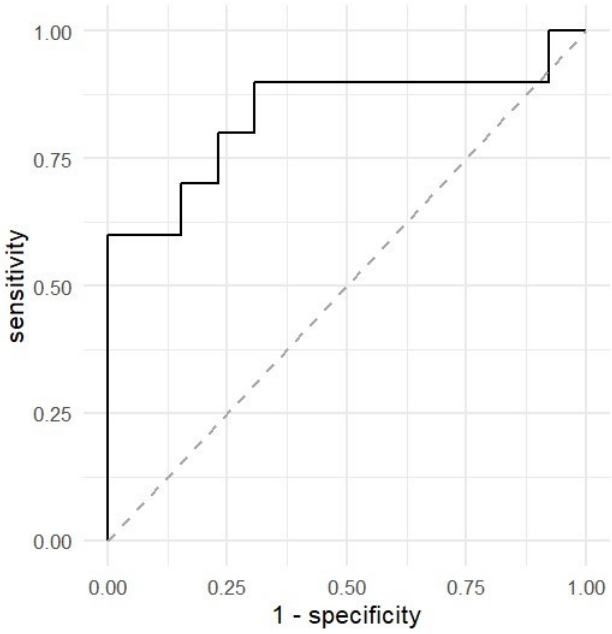


Figure 5.3: ROC curve of the ensemble model on the test set.

For a decision threshold that ensures a minimum sensitivity of 95%

A decision threshold of 0.3149, ensuring a minimum sensitivity of 95% on cross-validation was derived from that dataset.

The corresponding confusion matrix is on the table 5.4. From 23 PET images of pulmonary nodules, 8 were true positives, 9 were true negatives, 2 were false negatives and 4 were false positives. Therefore, the ensemble model had a sensibility of 80.00%, a specificity of 69.23% and an accuracy of 73.91% for this threshold, on test set.

		Actual class	
		Malignant	Benign
Predicted class	Malignant	8	4
	Benign	2	9

Table 5.4: Confusion matrix for the pulmonary nodules classification on test set by the ensemble model based on a threshold that ensures a minimum sensitivity of 95%

5.2.3 Comparison between the CNN ensemble model and the SUVmax

The figure 5.4 shows a comparison of the ROC curves between the bwSUVmax and the CNN model. Since the ROC curves cross each other at various points, a paired comparison with the Venkatraman test was applied to evaluate the equivalency of the curves rather than the

area under the curve. The test statistic (E) was 22 and the two-side P-value was 0.7995, based on 2000 permutations.

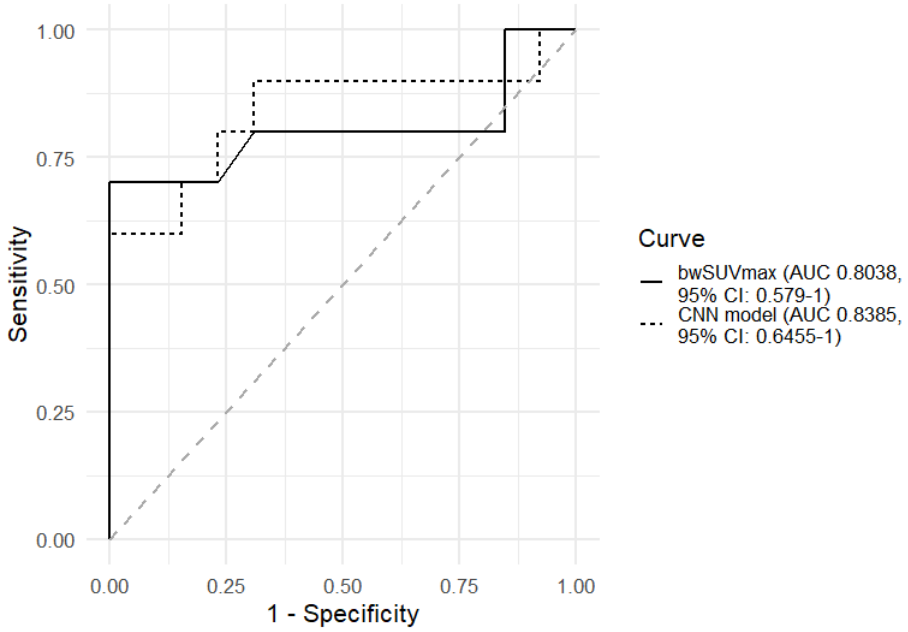


Figure 5.4: Comparison of the ROC curve between of the best CNN model and the SUVmax on the test set.

This page is intentionally left blank

Chapter 6

Discussion

In this dissertation, a 3D CNN model for classification of solid pulmonary nodules was developed from an annotated dataset of PET images specifically created for that purpose. This classification task aimed to differentiate between benign and malignant nodules. To best of my knowledge, this is the first study that addresses the creation of a deep learning model for classification of indeterminate pulmonary nodules, using PET images as inputs.

The only attempts of using machine learning models for differential diagnosis of indeterminate pulmonary nodules addressed classical methods and handcrafted imaging features as inputs, in some cases, combined with non-imaging risk factors (Herder et al., 2005; Y. Yang et al., 2018; S. Chen et al., 2019; Teramoto et al., 2019; H.-Y. Guo et al., 2020).

Y. Han et al. (2021) trained several classical machine learning models and a 2D CNN pretrained (VGG16) for distinguishing the histologic subtype of pulmonary lesions in patients already diagnosed with a lung cancer, from a dataset of 1419 PET/CT fusion images. The deep learning model obtained a area under the ROC curve of 0.903. Despite the use of a deep learning algorithm, it should be noted that the classification problem is not the same as in the current dissertation because it only includes malignant lesions. Additionally, the type of images also includes the CT information, unlike the present work which only uses PET data.

The final model of the current dissertation yielded an area under the ROC curve of 0.8385 (95% CI: 0.6455-1.000) on the test set. Named as 3D stacked CNN model, it has four 3D convolutional layer, three 3D max-pooling layers and fully connected network with 3 layers. Its has relatively simple architecture network when compared to recent types of networks published, which are more complex and deeper (Khan et al., 2020). Since the inputs of the 3D CNN are volumes, it receives information of whole nodule, unlike a 2D CNN that receives information from some slices intersecting the nodule, for this reason a 3D CNN was preferred. However, a 3D CNN has the cost of a higher number of parameters and higher risk of overfitting. As such, the capacity of the model was carefully adjusted to the problem and size of dataset. Several

regularization methods were also applied, such as early stopping and L2 regularization.

Secondarily, the probabilistic predictions were converted in the target classes by determining an optimum threshold. Its determination was performed in the cross-validation rather than in the test set to prevent overfiting to the latter and to simulate a real scenario of classification with the test set. Two approaches were used. The Youden's index and a preassigned value for sensitivity of at least 95%, both yielding an accuracy of 73.91%. However, the sensitivity obtained with the second method in the test set was much more favourable (80% *vs.* 40%). This is explained by the characteristics of each method and by the variance associated to reduced size of the test set (23 images). The specificity of the second method of threshold moving was 73.91%. Youden's index is popular in Medicine. Originally, it was assumed that the resulting threshold imposed equal costs for false positives and false negatives, but more recently was demonstrated that the costs depends on the prevalence of disease and equal cost only occurs in a balanced dataset (Smits, 2010). A threshold that maximizes the specificity, setting a minimum value sensitivity of 95%, can be a more appropriate approach for the current problem because a greater cost is placed on false negatives than on false positives, being assumed that the cost of missing a malignant lesion is higher than the cost of additional investigations and psychological distress caused by a false positive if the ultimate goal is to maximize the overall survival.

The performance of the 3D CNN model was compared with the SUVmax of the nodules. The model had an area under the ROC curve higher than the SUVmax in the test set (0.8385 *vs.* 0.8038). However, the equivalency between the two ROC curves was not rejected by a hypothesis test that compared their shape. Because the test set was not sized to ensure an adequate statistical power to the applied test, this negative result requires confirmation in well-powered studies.

Other types of 3D CNNs were also proposed, achieving a slightly lower area under the ROC curve than the 3D stacked CNN in the cross-validation. These networks were inspired by VGG16 and Inception-v2. They are deeper and have some features that make them more efficient, such as factorization of convolutions, introduction of the sparsity in the network or $1 \times 1 \times 1$ convolutions.

Transfer learning had a lower classification performance in cross-validation than the models trained with random weight initialization. This could be explained by difference between the source domain were the CNN was pre-trained (ImageNet) and the target domain, by 2D architecture requiring 2D inputs, or by the type of pre-trained network (ResNet).

Deep learning models usually need to be trained in a big dataset to prevent overfitting. However, building an annotated dataset in medical imaging is a time-consuming and a labor-intensive task. Furthermore, the particularity of the task and the imaging modality involved imply that the number of cases available to be included may be limited. The dataset created in this

dissertation is relatively small. Even though, a model was successfully trained and regularized. The main factor that limited the size of the dataset was the number of eligible cases available to be included, despite the efforts undertaken to select, gather, preprocess and annotate the data.

Data augmentation with translations, rotations and Gaussian noise injection was also an approach applied with the aim of improving the generalization of the models. Data augmentation assumes that more information can be extracted from the original training dataset through augmentations in order to reduce the difference between this dataset and validation or test set (Shorten & Khoshgoftaar, 2019). However, the models trained with data augmentation had a performance in the cross-validation consistently lower than when trained in the original training dataset. It was out scope of the phase of model selection to make statistical inference of differences between models in the cross-validation to the population, being unknown the meaning of those differences. It is also unknown their cause. It is hypothesized that the size of original dataset is insufficient for the augmentations to produce any effect, or the type or the parameters of the transformations are not the most appropriate to lead to an improvement of classification performance in this specific type of image data and problem, or the factor of augmentation is insufficient. The effect of each type of transformation on the performance of the models was not assessed, but the type of transformations applied was class-preserving.

High-quality and representative datasets are essential for developing machine learning models and for ensuring they have acceptable generalization performance on unseen cases. Although this is a retrospective study, the target population to participate in the study was accurately defined by a set of objective criteria. As consequence, the model built is only valid in this population unless a subsequent validation in others population occurs in future. That is, the model built is valid for a population of individuals who underwent a 2-[^{18}F]FDG PET/CT to clarify an indeterminate solid pulmonary nodule with > 8 mm and no antecedent of malignant disease. The performance of the model is unknown in subsolid or $\leq$ 8 mm nodules, or in those patients with antecedents of malignancy (excepting the cases referred to in section 3.1).

The quality of a dataset also depends on the quality of the ground truth. Predictive modelling for diagnosis purposes follows the same principles of the diagnosis tests regarding obtaining an unbiased ground truth (Moons et al., 2014). A proof about the presence or absence of the target disease should be obtained without knowledge of the index test and vice-versa (Weinstein et al., 2005; Moons et al., 2014). Similarly, the ground truth should not contain information from the data where a predictive model will be built, otherwise the model will have an optimistic performance (incorporation bias) (Moons et al., 2014). In the current study, the incorporation bias was prevented by using the result of the histopathological or cytopathological examination of a specimen obtained by biopsy or surgical excision or, alternatively, a follow-up period with CT. Therefore, there was a differential verification of the disease status. The histopathological

characterization of the lesion was the main method to obtain the ground truth, representing 62.8% of the cases. The nodule status was determined by the cytopathological examination in one case. The CT imaging follow-up was the method to obtain the definitive diagnosis in the remaining cases, with 85% of the patients having a follow-up time of at least 2 years. Surgical resection is the gold standard for definitive diagnosis of pulmonary nodules (Ricciardi et al., 2020), that is an unbiased ground truth. The biopsy also provides a direct evidence of malignancy, but there is a risk of non-specific benign changes are false negatives (Laurent et al., 2003). To eliminate that risk of bias in the biopsy, only definitive evidence of a benign pathology was consider (on first or repeated biopsies), otherwise the follow-up criterion was applied. Imaging follow-up provides an indirect, but still strong, evidence of the status of the nodule, leading to a low risk of bias in the ground-truth. The defined follow-up criteria ensured that a malignant tumor is missed in < 1% of cases, according to the previous literature (Callister et al., 2015).

This study has some limitations. The model was built in a relatively small dataset. Despite the efforts of regularization, it is unknown its performance in a larger dataset. Also, the test set was small, so the generalization performance is highly dependent of the data split. One of the three PET scanner only contributed with 14 images to the dataset. It is unknown how the model generalize in a PET scanner basis and for new patients images to this scanner.

Because this is a retrospective study, the decision of performing a PET/CT exam or a biopsy or excision of the pulmonary nodule, as well as the duration of follow-up period was at the discretion of the attending physician. The decision criteria may have changed over time, as part of the evolution of knowledge in this area, and according to attending physician, resulting in a selection and partial verification biases (Schmidt & Factor, 2013). This is inferred by the growing number of patients who each year met the established criteria to be included in the sample, being particularly evident when comparing the periods before and from 2016.

When multiple nodules were present, the dataset only included the most suspicious nodule each patient, instead of all the nodules, but in practice it is important to know the status of all of them.

The input image data stores SUV by voxel. SUV has been popularized, but another less used measure was claimed to be more accurate, the standardized uptake value normalized by the lean body mass (SUL) (O et al., 2016), once the lean and the fat tissues have different metabolic profiles. Image data were not recalculated to show SUL because the DICOM files from one of the PET scanners did not have the height data recorded.

The model with the best average performance in the cross-validation may not be the model with the best generalization performance on unseen images because the high number of generated models may cause overfitting to the validation data. Therefore, if other than the best model was selected during the cross-validation, namely by a method as proposed by Ng (1997),

a better generalization performance might have been reached.

The following proposals are suggested as future work:

- Evaluate the proposed model in a larger dataset, preferably collected prospectively from multiple centres and PET/CT scanners, and eventually retrain it in those data;
- Train a CNN model that considers not only the PET data, but also the low-dose CT data (obtained from the same exam) and non-imaging risk factors;
- Apply a method that handles potential overfitting to the validation set at the model selection phase as that proposed by Ng (1997) or using nested cross-validation (Wainer & Cawley, 2021);
- Characterize the model performance according to the histopathological type of lung cancer;
- Approach the problem as a multi-class task by training a network that aims to predict the histopathological type of the pulmonary nodule;
- Make the CNN model explainable, since clinical decision support systems need to make traceable decisions to gain trust of the physicians (Singh et al., 2020);
- Predict probability estimates rather than the target class by calibrating a CNN, since this can be useful for the clinical decision. Although the neural networks are known to be poorly calibrated, there are post-processing methods to alleviate this problem as the temperature scaling (C. Guo et al., 2017).

In conclusion, all objectives proposed in this dissertation were reached. A 3D CNN model for classification of indeterminate solid pulmonary nodules was successfully developed from an annotated dataset of 2-[^{18}F]FDG PET images that was created for that propose aiming to solve a real-world problem.

This page is intentionally left blank

Appendix A

Supplemental material about the database of pulmonary nodules

Table A.1: Tabular data

Feature	Short name	Type	Scale	Encoding / Measure
Pseudonymized ID	id	qualitative	nominal	—
Age	age	numeric, discrete	ratio	years
Sex	sex	qualitative	nominal	0 - male, 1 - female
Smoking habits	smoking	qualitative	nominal	0 - no smoking history, 1 - current or previous smoking history
Time since quitting smoking	exsmoking_time	numeric, discrete	ratio	years
Pack-year	pack_year	numeric, continuous	ratio	pack-years

(Continued on next page)

Table A.1 – continued from previous page

Feature	Short name	Type	Scale	Encoding / Measure
Occupational exposition to carcinogens	exposition	qualitative	nominal	0 - no exposition, 1 - with exposition
Emphysema or pulmonary fibrosis	lung_disease	qualitative	nominal	0 - no disease, 1- a disease is present
Family history of lung cancer	fam_lung_ca	qualitative	nominal	0 - no familiar history, 1 - with familiar history
Nodule location	location_nod	qualitative	nominal	0 - right upper lobe, 1 - right middle lobe, 2 - right lower lobe, 3 - left upper lobe, 4 - left lower lobe
Nodule diameter	diam_nod	numeric, discrete	ratio	millimeter
Nodule spiculation	spiculation	qualitative	nominal	0 - no spiculation, 1 - spiculation is present
Pleural indentation	indentation	qualitative	nominal	0 - absent, 1 - present
Nodule Multiplicity	nod_multiplicity	numeric, discrete	Ratio	dimensionless

(Continued on next page)

Table A.1 – continued from previous page

Feature	Short name	Type	Scale	Encoding / Measure
maximum standardized uptake value normalized by the body mass (SUVmax) of the nodule	suvmax_nod	numeric, continuous	ratio	dimensionless
Mediastinal or hilar lymphadenopathy	adenopathy	qualitative	nominal	0 - no lymphadenopathy, 1- with lymphadenopathy
Mediastinal or hilar lymphadenopathy diameter	diam_adenop	numeric, discrete	ratio	millimeter
SUVmax of the more intense lymphadenopathy	suvmax_adenop	numeric, continuous	ratio	dimensionless
Scanner model	scanner_model	qualitative	nominal	0 - Siemens Biograph 6, 1 - GE Discovery LS, 2 - GE Discovery IQ
Relative path for cropped PET images	path_crop_img	qualitative	text	—

(Continued on next page)

Table A.1 – continued from previous page

Feature	Short name	Type	Scale	Encoding / Measure
Multiclass target	multiclass_target	qualitative	nominal	0 - benign, 1 - adenocarcinoma, 2 - squamous cell carcinoma, 3 - small cell lung cancer, 4 - large cell carcinoma, 5 - carcinoid tumor, 6 - metastasis, 7 - other
Binary target	binary_target	qualitative	nominal	0 - benign tumor, 1 - malignant tumor

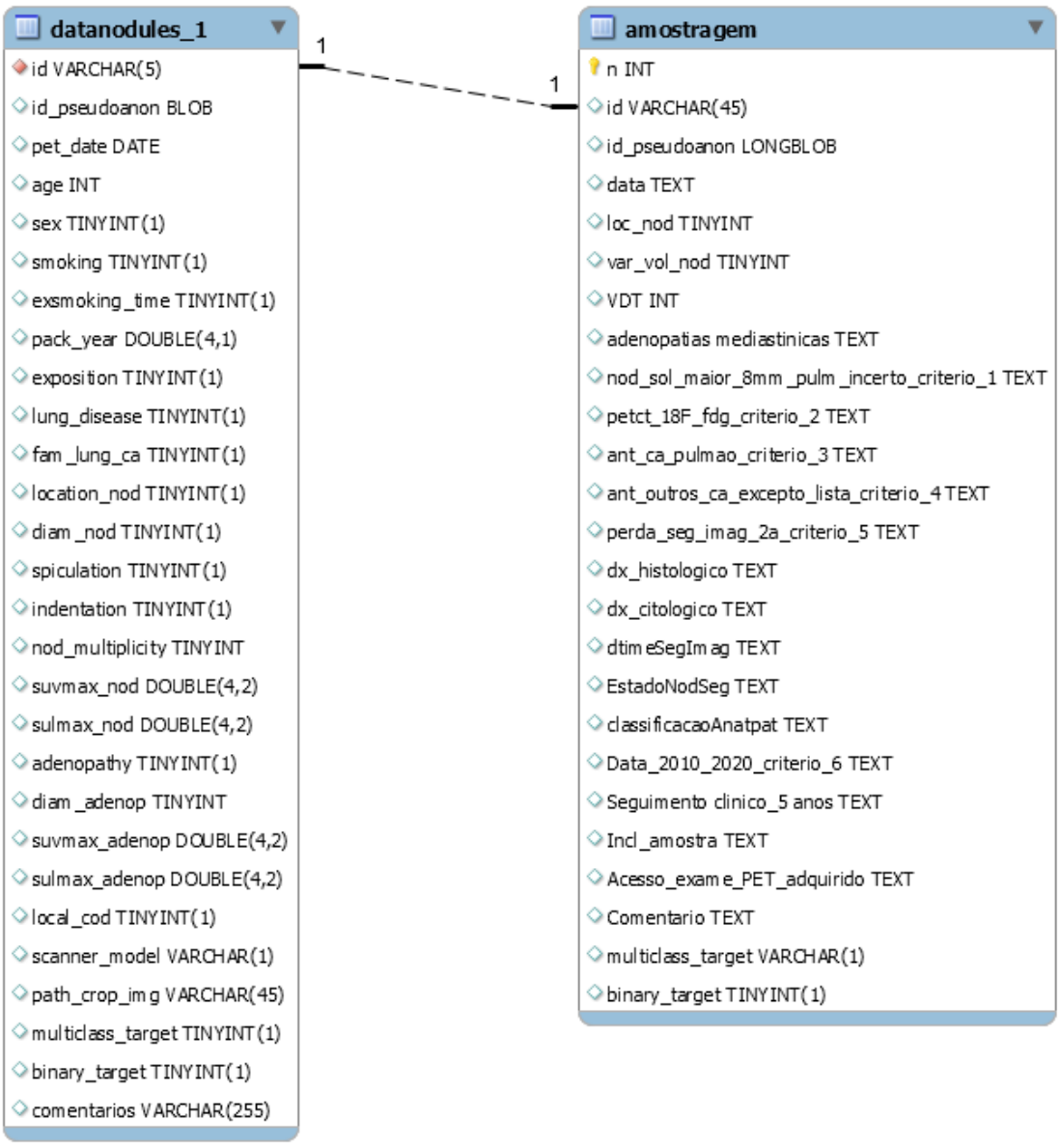


Figure A.1: Pulmonary nodules MySQL database for tabular data.

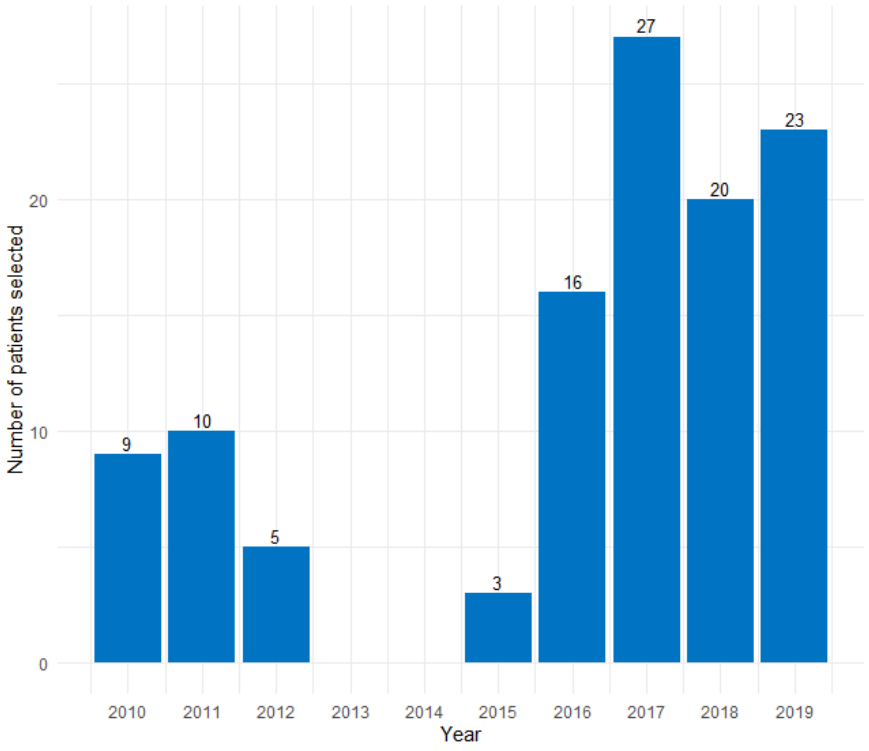


Figure A.2: Patients selected to be included in the dataset per year.

Appendix B

Supplemental tables with procedures and results about the classification task

Table B.1: List of experiments

No.	Batch size	Architecture	σ	Initialization	LR	Regularizer
1	68	conv(8,5,5,5) + mpool + conv(16,5,5,5) + mpool + flatten + fcn(24,16,1)	ReLU	Xavier	0.0005	—
2	68	conv(8,5,5,5) + mpool + conv(16,5,5,5) + mpool + flatten + fcn(24,16,1)	LeakyReLU (α=0.3)	Xavier	0.0005	—
3	68	conv(8,5,5,5) + mpool + conv(16,5,5,5) + mpool + flatten + fcn(24,16,1)	LeakyReLU (α=0.3)	He	0.0005	—
4	68	conv(8,5,5,5) + mpool + conv(16,5,5,5) + mpool + flatten + fcn(32,16,1)	LeakyReLU (α=0.3)	He	0.0005	—
5	68	conv(8,3,3,3) + mpool + conv(16,3,3,3) + mpool + flatten + fcn(32,16,1)	LeakyReLU (α=0.3)	Xavier	0.0005	—
6	68	conv(8,3,3,3) + mpool + conv(16,3,3,3) + mpool + flatten + fcn(32,16,1)	LeakyReLU (α=0.3)	He	0.0005	—
7	68	conv(6,3,3,3) + mpool + conv(16,3,3,3) + mpool + flatten + fcn(24,16,1)	LeakyReLU (α=0.3)	He	0.0005	—

(Continued on next page)

Table B.1 – continued from previous page

No.	Batch size	Architecture	σ	Initialization	LR	Regularizer
8	68	conv(8,3,3,3) + mpool + conv(16,3,3,3) + mpool + conv(32,3,3,3) + mpool + flatten + fcn(32,16,1)	LeakyReLU (α=0.3)	He	0.0005	—
9	16	conv(8,3,3,3) + mpool + conv(16,3,3,3) + mpool + conv(32,3,3,3) + mpool + flatten + fcn(32,16,1)	LeakyReLU (α=0.3)	He	0.0005	—
10	68	conv(8,3,3,3) + mpool + conv(16,3,3,3) + mpool + conv(32,3,3,3) + mpool + conv(64,3,3,3) + flatten + fcn(32,16,1)	LeakyReLU (α=0.3)	He	0.0005	—
11	68	conv(8,3,3,3) + mpool + conv(16,3,3,3) + mpool + conv(32,3,3,3) + mpool + conv(64,3,3,3) + flatten + fcn(32,16,1)	LeakyReLU (α=0.3)	He	0.001	—
12	68	conv(8,3,3,3) + mpool + conv(16,3,3,3) + mpool + conv(32,3,3,3) + mpool + conv(64,3,3,3) + flatten + fcn(32,16,1)	LeakyReLU (α=0.3)	He	0.001	L2(0.00098)
13	68	conv(8,3,3,3) + overlap mpool + conv(16,3,3,3) + conv(16,3,3,3) + overlap mpool + conv(32,3,3,3) + conv(32,3,3,3) + conv(32,3,3,3) + overlap mpool + flatten + fcn(16,16,1)	LeakyReLU (α=0.3)	He	0.0005	L2(0.004) and dropout(0.5)
14	68	conv(8,3,3,3) + mpool + conv(16,3,3,3) + conv(16,3,3,3) + mpool + conv(32,3,3,3) + conv(32,3,3,3) + conv(32,3,3,3) + mpool + flatten + fcn(16,16,1)	LeakyReLU (α=0.3)	He	0.0005	L2(0.004) and dropout(0.5)

(Continued on next page)

Table B.1 – continued from previous page

No.	Batch size	Architecture	σ	Initialization	LR	Regularizer
15	68	conv(8,3,3,3) + overlap mpool + conv(16,3,3,3) + conv(16,3,3,3) + overlap mpool + conv(32,3,3,3) + conv(32,3,3,3) + conv(32,3,3,3) + overlap mpool + flatten + fcn(16,1)	LeakyReLU (α=0.3)	He	0.0005	L2(0.004) and dropout(0.5)
16	68	conv(8,3,3,3) + overlap mpool + conv(16,3,3,3) + conv(16,3,3,3) + overlap mpool + conv(32,3,3,3) + conv(32,3,3,3) + conv(32,3,3,3) + overlap mpool + gap + fcn(16,1)	LeakyReLU (α=0.3)	He	0.001	L2(0.002) and dropout(0.5)
17	68	conv(8,3,3,3) + overlap mpool + conv(16,3,3,3) + conv(16,3,3,3) + overlap mpool + conv(32,3,3,3) + conv(32,3,3,3) + conv(32,3,3,3) + overlap mpool + flatten + fcn(8,8,1)	LeakyReLU (α=0.3)	He	0.0005	L2(0.003) and dropout(0.5)
18	68	conv(8,3,3,3) + overlap mpool + conv(16,3,3,3) + conv(16,3,3,3) + overlap mpool + conv(32,3,3,3) + conv(32,3,3,3) + conv(32,3,3,3) + overlap mpool + flatten + fcn(4,4,1)	LeakyReLU (α=0.3)	He	0.002	L2(0.002) and dropout(0.45)
19	68	conv(8,3,3,3) + overlap mpool + conv(16,3,3,3) + conv(16,3,3,3) + overlap mpool + conv(32,3,3,3) + conv(32,3,3,3) + conv(32,3,3,3) + overlap mpool + flatten + fcn(1)	LeakyReLU (α=0.3)	He	0.0005	L2(0.002)

(Continued on next page)

Table B.1 – continued from previous page

No.	Batch size	Architecture	σ	Initialization	LR	Regularizer
20	68	conv(8,3,3,3) + overlap mpool + conv(16,3,3,1) + conv(16,1,1,3) + conv(16,3,3,1) + conv(16,1,1,3) + overlap mpool + conv(32,3,3,1) + conv(32,1,1,3) + conv(32,3,3,1) + conv(32,1,1,3) + conv(32,3,3,1) + conv(32,1,1,3) + overlap mpool + flatten + fcn(4,1)	LeakyReLU (α=0.3)	He	0.0005	L2(0.001)
21	68	conv(8,3,3,3) + mpool + Inception + Inception + Reduction + Inception + Inception + Reduction + gap + fcn(1)	LeakyReLU (α=0.3)	He	0.0005	L2(0.0006)
22	8	conv(8,3,3,3) + mpool + conv(16,3,3,3) + mpool + conv(32,3,3,3) + mpool + conv(64,3,3,3) + flatten + fcn(32,16,1)	LeakyReLU (α=0.3)	He	0.0001	L2(0.03) and data augmentation
23	8	conv(8,3,3,3) + mpool + conv(16,3,3,3) + mpool + conv(32,3,3,3) + mpool + conv(64,3,3,3) + flatten + fcn(32,16,1)	LeakyReLU (α=0.3)	He	0.0001	L2(0.01) and data augmentation
24	8	conv(8,3,3,3) + overlap mpool + conv(16,3,3,3) + conv(16,3,3,3) + overlap mpool + conv(32,3,3,3) + conv(32,3,3,3) + conv(32,3,3,3) + overlap mpool + flatten + fcn(1)	LeakyReLU (α=0.3)	He	0.0001	L2(0.02) and data augmentation

(Continued on next page)

Table B.1 – continued from previous page

No.	Batch size	Architecture	σ	Initialization	LR	Regularizer
25	8	conv(8,3,3,3) + overlap mpool + conv(16,3,3,3) + conv(16,3,3,3) + overlap mpool + conv(32,3,3,3) + conv(32,3,3,3) + conv(32,3,3,3) + overlap mpool + flatten + fcn(1)	LeakyReLU (α=0.3)	He	0.0001	L2(0.06) and data augmentation
26	16	conv(8,3,3,3) + mpool + Inception + Inception + Reduction + Inception + Inception + Reduction + gap + fcn(1)	LeakyReLU (α=0.3)	He	0.001	L2(0.01) and data augmentation
27	8	conv(8,3,3,3) + mpool + Inception + Inception + Reduction + Inception + Inception + Reduction + gap + fcn(1)	LeakyReLU (α=0.3)	He	0.0001	L2(0.04) and data augmentation
28	68	ResNet50 (base) + gap + fcn (8,1)	ReLU	He	5×10^{-7}	Transfer learning and dropout(0.5)

Table B.2: Epoch with the minimum validation loss by fold for the different trained models obtained by early stopping. Val. Loss - Validation loss

	F1		F2		F3		F4	
Model	**Epoch**	**Val. Loss**	**Epoch**	**Val. Loss**	**Epoch**	**Val. Loss**	**Epoch**	**Val. Loss**
1	10	0.6446	23	0.3992	22	0.4002	50	0.4372
2	8	0.6445	24	0.3563	21	0.4112	44	0.4132
3	3	0.6533	13	0.4083	7	0.3981	19	0.3567
4	3	0.6684	14	0.3648	7	0.4117	25	0.3564
5	17	0.6498	26	0.4145	28	0.4320	33	0.3902
6	3	0.6685	22	0.4660	12	0.3973	21	0.3505
7	3	0.6665	12	0.4122	8	0.3958	21	0.3666
8	13	0.6541	26	0.5010	28	0.3927	63	0.3294
9	3	0.6565	10	0.4961	9	0.3895	38	0.4013
10	3	0.6785	20	0.4693	18	0.4063	41	0.2893
11	1	0.6793	13	0.4289	10	0.4039	31	0.2883
12	1	1.0159	13	0.7510	10	0.7286	32	0.6241
13	43	1.8607	40	1.6354	58	1.5334	100	1.4203
14	43	1.8074	37	1.7258	50	1.5368	82	1.3931
15	5	1.8613	30	1.5219	20	1.4974	66	1.3220
16	22	1.2583	63	0.9716	44	1.0213	38	0.7917
17	29	1.5197	40	1.3042	51	1.1974	100	1.1235
18	56	0.9040	28	0.9644	62	0.6708	94	0.6830
19	5	1.2068	19	0.9084	12	0.8852	33	0.8948
20	6	1.2002	21	0.9245	20	0.9123	48	0.8748
21	10	1.1227	28	0.8556	18	0.8627	41	0.8205
22	78	0.8269	48	0.7109	23	0.6632	100	0.6013
23	23	1.4317	88	1.0219	38	0.9898	39	0.8546
24	10	1.1456	11	1.1024	17	1.1171	66	0.7851
25	34	0.8372	19	0.6767	15	0.5610	64	0.4629
26	7	0.8722	91	0.6840	15	0.5507	16	0.5710
27	34	0.8392	29	0.7389	71	0.6039	48	0.5697
28	80	0.6910	49	0.6931	43	0.7001	66	0.6922

Table B.3: Area under the ROC curve on cross-validation for the different models. SD - standard deviation.

	AUC ROC curve					
Model	**F1**	**F2**	**F3**	**F4**	**Mean**	**SD**
1	0.6833	0.9083	0.8667	0.8881	0.8366	0.1036
2	0.6833	0.9250	0.8750	0.9161	0.8499	0.1131
3	0.6833	0.8833	0.8750	0.9510	0.8482	0.1151
4	0.6917	0.9167	0.8750	0.9371	0.8551	0.0970
5	0.6917	0.9083	0.8333	0.9161	0.8374	0.1040
6	0.6833	0.8417	0.8917	0.9510	0.8419	0.1148
7	0.6833	0.9083	0.8833	0.9441	0.8548	0.1170
8	0.7083	0.8583	0.8833	0.9650	0.8538	0.1071
9	0.6917	0.8667	0.8583	0.8881	0.8262	0.0905
10	0.7667	0.8583	0.8583	0.9720	0.8638	0.0841
11	0.7917	0.9000	0.8750	0.9720	0.8847	0.0744
12	0.7917	0.9000	0.8750	0.9790	0.8864	0.0772
13	0.7333	0.8750	0.8667	0.8881	0.8408	0.0722
14	0.7000	0.7833	0.8667	0.9161	0.8165	0.0951
15	0.6917	0.8333	0.8750	0.9231	0.8308	0.0997
16	0.7250	0.9000	0.7583	0.8601	0.8108	0.0827
17	0.7250	0.8917	0.8833	0.9021	0.8505	0.0840
18	0.7583	0.8583	0.9250	0.8741	0.8539	0.0698
19	0.7333	0.9250	0.9417	0.9161	0.8790	0.0977
20	0.7583	0.9000	0.8750	0.9371	0.8676	0.0772
21	0.7250	0.9083	0.8917	0.9650	0.8725	0.1032
22	0.7333	0.8500	0.8417	0.9371	0.8405	0.0835
23	0.6750	0.8417	0.8583	0.8462	0.8053	0.0871
24	0.7000	0.6750	0.8500	0.8182	0.7608	0.0862
25	0.7000	0.7833	0.8667	0.9301	0.8200	0.1001
26	0.6833	0.8417	0.8833	0.8252	0.8084	0.0869
27	0.7333	0.8333	0.8417	0.8741	0.8206	0.0608
28	0.7167	0.8083	0.8500	0.7203	0.7738	0.0662

Table B.4: Area under the ROC curve of the best stacked 3D model (model 12) over 10 iterations of cross-validation.

	AUC ROC curve					
Iteration	**F1**	**F2**	**F3**	**F4**	**Mean**	**SD**
1	0.7917	0.9000	0.8583	0.9790	0.8823	0.0784
2	0.7917	0.9000	0.8750	0.9790	0.8864	0.0772
3	0.7917	0.9000	0.8750	0.9650	0.8829	0.0717
4	0.7917	0.9000	0.8750	0.9650	0.8829	0.0717
5	0.7917	0.9000	0.8583	0.9720	0.8805	0.0756
6	0.7917	0.9000	0.8667	0.9720	0.8826	0.0749
7	0.7917	0.9000	0.8667	0.9720	0.8826	0.0749
8	0.7917	0.9000	0.8750	0.9720	0.8847	0.0744
9	0.7917	0.9000	0.8500	0.9720	0.8784	0.0765
10	0.7917	0.9000	0.8583	0.9650	0.8788	0.0728
Mean	**0.7917**	**0.9000**	**0.8658**	**0.9713**	**0.8822**	**0.0748**

Table B.5: Area under the ROC curve of the best VGG-like model (model 19) over 10 iterations of cross-validation.

	AUC ROC curve					
Iteration	**F1**	**F2**	**F3**	**F4**	**Mean**	**SD**
1	0.7333	0.9250	0.9167	0.9371	0.8780	0.0968
2	0.7333	0.9250	0.9417	0.9091	0.8773	0.0969
3	0.7333	0.9250	0.9333	0.9231	0.8787	0.0970
4	0.7333	0.9250	0.9333	0.9301	0.8804	0.0981
5	0.7333	0.9250	0.9333	0.9161	0.8769	0.0960
6	0.7333	0.9250	0.925	0.8881	0.8679	0.0914
7	0.7333	0.9333	0.9167	0.9301	0.8784	0.0969
8	0.7333	0.9250	0.9083	0.9301	0.8742	0.0944
9	0.7333	0.9167	0.9333	0.9021	0.8714	0.0929
10	0.7333	0.9250	0.9167	0.9301	0.8763	0.0954
Mean	**0.7333**	**0.9250**	**0.9258**	**0.9196**	**0.8760**	**0.0956**

Table B.6: Area under the ROC curve of the best Inception-v2-like model (model 21) over 10 iterations of cross-validation.

	AUC ROC curve					
Iteration	**F1**	**F2**	**F3**	**F4**	**Mean**	**SD**
1	0.7250	0.9083	0.8917	0.9510	0.8690	0.0992
2	0.7250	0.9083	0.8917	0.9650	0.8725	0.1032
3	0.7250	0.9083	0.8917	0.9650	0.8725	0.1032
4	0.725	0.9083	0.8917	0.9510	0.8690	0.0992
5	0.7250	0.9083	0.8917	0.9301	0.8638	0.0938
6	0.7250	0.9083	0.8917	0.9510	0.8690	0.0992
7	0.7250	0.9083	0.8917	0.9371	0.8655	0.0955
8	0.7250	0.9083	0.8917	0.9371	0.8655	0.0955
9	0.7250	0.9083	0.8917	0.9580	0.8708	0.1012
10	0.7250	0.9083	0.8917	0.9650	0.8725	0.1032
Mean	**0.7250**	**0.9083**	**0.89170**	**0.9510**	**0.8690**	**0.0992**

References

3D Slicer. (2020 (accessed January 26, 2020)). https://www.slicer.org.

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Zheng, X. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems.* Retrieved from https://www.tensorflow.org/ (Software available from tensorflow.org)

Alberg, A. J., Brock, M. V., Ford, J. G., Samet, J. M., & Spivack, S. D. (2013, May 01). Epidemiology of lung cancer: Diagnosis and management of lung cancer, 3rd ed: American college of chest physicians evidence-based clinical practice guidelines. *CHEST*, *143*(5), e1S-e29S. Retrieved from https://doi.org/10.1378/chest.12-2345 doi: 10.1378/chest.12-2345

Allaire, J. (n.d.). *Tensorflow for r*. https://tensorflow.rstudio.com/. (Accessed: 2021-04-10)

Allaire, J., & Chollet, F. (2019). keras: R interface to 'keras' [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=keras (R package version 2.2.5.0)

Allaire, J., & Tang, Y. (2019). tensorflow: R interface to 'tensorflow' [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=tensorflow (R package version 2.0.0)

Anaconda software distribution. (2020). Anaconda Inc. Retrieved from https://docs.anaconda.com/

Bailay, D. L., Humm, J. L., Todd-Pokropek, A., & van Aswegen, A. (2015). *Nuclear medicine physics*. Vienna: INTERNATIONAL ATOMIC ENERGY AGENCY. Retrieved from https://www.iaea.org/publications/10368/nuclear-medicine-physics

Bradley, A. P. (1997). The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, *30*(7), 1145 - 1159. Retrieved from http://www.sciencedirect.com/science/article/pii/S0031320396001422 doi: https://doi.org/10.1016/S0031-3203(96)00142-2

Button, K. S., Ioannidis, J. P. A., Mokrysz, C., Nosek, B. A., Flint, J., Robinson, E. S. J., &

Munafò, M. R. (2013, May 01). Power failure: why small sample size undermines the reliability of neuroscience. *Nature Reviews Neuroscience*, *14*(5), 365-376. Retrieved from https://doi.org/10.1038/nrn3475 doi: 10.1038/nrn3475

Callister, M. E. J., Baldwin, D. R., Akram, A. R., Barnard, S., Cane, P., Draffan, J., ... on behalf of the British Thoracic Society Standards of Care Committee, B. T. S. P. N. G. D. G. (2015). British thoracic society guidelines for the investigation and management of pulmonary nodules: accredited by nice. *Thorax*, *70*(Suppl 2), ii1–ii54. Retrieved from https://thorax.bmj.com/content/70/Suppl_2/ii1 doi: 10.1136/thoraxjnl-2015-207168

Castiglioni, I., Gallivanone, F., Soda, P., Avanzo, M., Stancanello, J., Aiello, M., ... Salvatore, M. (2019). AI-based applications in hybrid imaging: how to build smart and truly multi-parametric decision models for radiomics. *European Journal of Nuclear Medicine and Molecular Imaging*, *46*(13), 2673-2699. Retrieved from https://doi.org/10.1007/s00259-019-04414-4 doi: 10.1007/s00259-019-04414-4

Chen, L., Zhou, Z., Sher, D., Zhang, Q., Shah, J., Pham, N.-L., ... Wang, J. (2019, mar). Combining many-objective radiomics and 3d convolutional neural network through evidential reasoning to predict lymph node metastasis in head and neck cancer. *Physics in Medicine & Biology*, *64*(7), 075011. Retrieved from https://doi.org/10.1088%2F1361-6560%2Fab083a doi: 10.1088/1361-6560/ab083a

Chen, S., Harmon, S., Perk, T., Li, X., Chen, M., Li, Y., & Jeraj, R. (2019). Using neighborhood gray tone difference matrix texture features on dual time point pet/ct images to differentiate malignant from benign fdg-avid solitary pulmonary nodules. *Cancer Imaging*, *19*(1), 56. Retrieved from https://doi.org/10.1186/s40644-019-0243-3 doi: 10.1186/s40644-019-0243-3

Choi, H., & Jin, K. H. (2018). Predicting cognitive decline with deep learning of brain metabolism and amyloid imaging. *Behavioural Brain Research*, *344*, 103 - 109. Retrieved from http://www.sciencedirect.com/science/article/pii/S0166432818301013 doi: https://doi.org/10.1016/j.bbr.2018.02.017

Chollet, F., & Allaire, J. (2018). Deep learning with r..

Chollet, F. e. a. (2015). *Keras*. https://github.com/fchollet/keras. GitHub.

Comninos, P. (2006). Three-dimensional transformations. In *Mathematical and computer programming techniques for computer graphics* (pp. 225–252). London: Springer London. Retrieved from https://doi.org/10.1007/978-1-84628-292-8_7 doi: 10.1007/978-1-84628-292-8_7

Curtis, F. E., & Scheinberg, K. (2020). *Adaptive stochastic optimization*.

de Groot, P. M., Wu, C. C., Carter, B. W., & Munden, R. F. (2018). The epidemiology of

lung cancer. *Translational Lung Cancer Research*, *7*(3). Retrieved from http://tlcr.amegroups.com/article/view/21996

DeLong, E. R., DeLong, D. M., & Clarke-Pearson, D. L. (1988). Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach. *Biometrics*, *44*(3), 837–845. Retrieved from http://www.jstor.org/stable/2531595

Digital imaging and communication in medicine. (n.d.). https://www.dicomstandard.org/. (Accessed: 2021-04-11)

Ding, Y., Sohn, J. H., Kawczynski, M. G., Trivedi, H., Harnish, R., Jenkins, N. W., ... Franc, B. L. (2019). A deep learning model to predict a diagnosis of alzheimer disease by using 18f-fdg pet of the brain. *Radiology*, *290*(2), 456-464. Retrieved from https://doi.org/10.1148/radiol.2018180958 (PMID: 30398430) doi: 10.1148/radiol.2018180958

Dumoulin, V., & Visin, F. (2016). *A guide to convolution arithmetic for deep learning.*

Ehteshami Bejnordi, B., Veta, M., Johannes van Diest, P., van Ginneken, B., Karssemeijer, N., Litjens, G., ... the CAMELYON16 Consortium (2017, 12). Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer. *JAMA*, *318*(22), 2199-2210. Retrieved from https://doi.org/10.1001/jama.2017.14585 doi: 10.1001/jama.2017.14585

Elia, S., Loprete, S., De Stefano, A., & Hardavella, G. (2019). Does aggressive management of solitary pulmonary nodules pay off? *Breathe*, *15*(1), 15–23. Retrieved from https://breathe.ersjournals.com/content/15/1/15 doi: 10.1183/20734735.0275-2018

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, *542*(7639), 115-118. Retrieved from https://doi.org/10.1038/nature21056 doi: 10.1038/nature21056

Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., ... Dean, J. (2019). A guide to deep learning in healthcare. *Nature Medicine*, *25*(1), 24-29. Retrieved from https://doi.org/10.1038/s41591-018-0316-z doi: 10.1038/s41591-018-0316-z

Fedorov, A., Beichel, R., Kalpathy-Cramer, J., Finet, J., Fillion-Robin, J.-C., Pujol, S., ... Kikinis, R. (2012). 3D Slicer as an image computing platform for the Quantitative Imaging Network. *Magnetic Resonance Imaging*, *30*(9), 1323 - 1341. Retrieved from http://www.sciencedirect.com/science/article/pii/S0730725X12001816 (Quantitative Imaging in Cancer) doi: https://doi.org/10.1016/j.mri.2012.05.001

Glorot, X., & Bengio, Y. (2010, 13–15 May). Understanding the difficulty of training deep feedforward neural networks. In Y. W. Teh & M. Titterington (Eds.), (Vol. 9, pp. 249–256). Chia Laguna Resort, Sardinia, Italy: JMLR Workshop and Conference Proceedings. Retrieved from http://proceedings.mlr.press/v9/glorot10a.html

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. (http://www.deeplearningbook.org)

Goodfellow, I., Warde-Farley, D., Mirza, M., Courville, A., & Bengio, Y. (2013, 17–19 Jun). Maxout networks. In S. Dasgupta & D. McAllester (Eds.), *Proceedings of the 30th international conference on machine learning* (Vol. 28, pp. 1319–1327). Atlanta, Georgia, USA: PMLR. Retrieved from http://proceedings.mlr.press/v28/goodfellow13.html

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., ... Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, *77*, 354 - 377. Retrieved from http://www.sciencedirect.com/science/article/pii/S0031320317304120 doi: https://doi.org/10.1016/j.patcog.2017.10.013

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). *On calibration of modern neural networks.*

Guo, H.-Y., Lin, J.-T., Huang, H.-H., Gao, Y., Yan, M.-R., Sun, M., ... Yang, X.-N. (2020, Jan 01). Development and validation of a ¹⁸f-fdg pet/ct-based clinical prediction model for estimating malignancy in solid pulmonary nodules based on a population with high prevalence of malignancy. *Clinical Lung Cancer*, *21*(1), 47-55. Retrieved from https://doi.org/10.1016/j.cllc.2019.07.014 doi: 10.1016/j.cllc.2019.07.014

Han, J., Kamber, M., & Pei, J. (2011). *Data mining: Concepts and techniques* (3rd ed.). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Han, Y., Ma, Y., Wu, Z., Zhang, F., Zheng, D., Liu, X., ... Guo, X. (2021, Feb 01). Histologic subtype classification of non-small cell lung cancer using pet/ct images. *European Journal of Nuclear Medicine and Molecular Imaging*, *48*(2), 350-360. Retrieved from https://doi.org/10.1007/s00259-020-04771-5 doi: 10.1007/s00259-020-04771-5

Hatt, M., Le Rest, C. C., Tixier, F., Badic, B., Schick, U., & Visvikis, D. (2019). Radiomics: Data are also images. *Journal of Nuclear Medicine*, *60*(Supplement 2), 38S-44S. Retrieved from http://jnm.snmjournals.org/content/60/Supplement_2/38S.abstract doi: 10.2967/jnumed.118.220582

He, K., Zhang, X., Ren, S., & Sun, J. (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.),

Computer vision – eccv 2014 (pp. 346–361). Cham: Springer International Publishing.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Delving deep into rectifiers: Surpassing human-level performance on imagenet classification.*

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 ieee conference on computer vision and pattern recognition (cvpr)* (p. 770-778). doi: 10.1109/CVPR.2016.90

Herder, G. J., van Tinteren, H., Golding, R. P., Kostense, P. J., Comans, E. F., Smit, E. F., & Hoekstra, O. S. (2005). Clinical prediction model to characterize pulmonary nodules: Validation and added value of 18 f-fluorodeoxyglucose positron emission tomography. *Chest*, *128*(4), 2490 - 2496. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0012369215526627` doi: https://doi.org/10.1378/chest.128.4.2490

Huang, G., Liu, Z., & Weinberger, K. Q. (2016). Densely connected convolutional networks. *CoRR*, *abs/1608.06993*. Retrieved from `http://arxiv.org/abs/1608.06993`

Huang, Y., Xu, J., Zhou, Y., Tong, T., Zhuang, X., & t. A. D. N. I. A. (2019). Diagnosis of alzheimer's disease via multi-modality 3d convolutional neural network. *Frontiers in Neuroscience*, *13*, 509. Retrieved from `https://www.frontiersin.org/article/10.3389/fnins.2019.00509` doi: 10.3389/fnins.2019.00509

Ioffe, S., & Szegedy, C. (2015). *Batch normalization: Accelerating deep network training by reducing internal covariate shift.*

Jin Huang, & Ling, C. X. (2005). Using auc and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, *17*(3), 299-310. doi: 10.1109/TKDE.2005.50

Kermany, D. S., Goldbaum, M., Cai, W., Valentim, C. C. S., Liang, H., Baxter, S. L., ... Zhang, K. (2018, Feb 22). Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, *172*(5), 1122-1131.e9. Retrieved from `https://doi.org/10.1016/j.cell.2018.02.010` doi: 10.1016/j.cell.2018.02.010

Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020, Dec 01). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, *53*(8), 5455-5516. Retrieved from `https://doi.org/10.1007/s10462-020-09825-6` doi: 10.1007/s10462-020-09825-6

Kingma, D. P., & Ba, J. (2017). *Adam: A method for stochastic optimization.*

Kirienko, M., Sollini, M., Silvestri, G., Mognetti, S., Voulaz, E., Antunovic, L., ... Chiti, A. (2018). Convolutional Neural Networks Promising in Lung Cancer T-Parameter Assessment on Baseline FDG-PET/CT. *Contrast Media & Molecular Imaging*, *2018*, 6. Retrieved from `10.1155/2018/1382309`

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th international joint conference on artificial intelligence - volume 2* (p. 1137–1143). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 25* (pp. 1097–1105). Curran Associates, Inc. Retrieved from http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Kukačka, J., Golkov, V., & Cremers, D. (2017). *Regularization for deep learning: A taxonomy.*

Laurent, F., Montaudon, M., Latrabe, V., & Bégueret, H. (2003). Percutaneous biopsy in lung cancer. *European Journal of Radiology*, *45*(1), 60-68. Retrieved from https://www.sciencedirect.com/science/article/pii/S0720048X02002863 doi: https://doi.org/10.1016/S0720-048X(02)00286-3

Le, Q. V., Ngiam, J., Chen, Z., Chia, D., Koh, P. W., & Ng, A. Y. (2010). Tiled convolutional neural networks. In *Proceedings of the 23rd international conference on neural information processing systems - volume 1* (p. 1279–1287). Red Hook, NY, USA: Curran Associates Inc.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436-444. Retrieved from https://doi.org/10.1038/nature14539 doi: 10.1038/nature14539

Lin, E., & Alavi, A. (2009). PET and PET/CT: A Clinical Guide, 2nd ed. *American Journal of Neuroradiology*, *30*(9), E142–E142. doi: 10.3174/ajnr.a1720

Lin, M., Chen, Q., & Yan, S. (2014). Network in network. *CoRR*, *abs/1312.4400*.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, *42*, 60 - 88. Retrieved from http://www.sciencedirect.com/science/article/pii/S1361841517301135 doi: https://doi.org/10.1016/j.media.2017.07.005

Liu, M., Cheng, D., Yan, W., & the Alzheimer's Disease Neuroimaging Initiative. (2018). Classification of alzheimer's disease by combination of convolutional and recurrent neural networks using fdg-pet images. *Frontiers in Neuroinformatics*, *12*, 35. Retrieved from https://www.frontiersin.org/article/10.3389/fninf.2018.00035 doi: 10.3389/fninf.2018.00035

Lu, L., Shin, Y., Su, Y., & Karniadakis, G. E. (2020). *Dying relu and initialization: Theory and numerical examples.*

López-Ratón, M., Rodríguez-Álvarez, M. X., Cadarso-Suárez, C., & Gude-Sampedro, F.

(2014). Optimalcutpoints: An r package for selecting optimal cutpoints in diagnostic tests. *Journal of Statistical Software, Articles*, *61*(8), 1–36. Retrieved from https://www.jstatsoft.org/v061/i08 doi: 10.18637/jss.v061.i08

MacMahon, H., Naidich, D. P., Goo, J. M., Lee, K. S., Leung, A. N. C., Mayo, J. R., ... Bankier, A. A. (2017). Guidelines for Management of Incidental Pulmonary Nodules Detected on CT Images: From the Fleischner Society 2017. *Radiology*, *284*(1), 228-243. Retrieved from https://doi.org/10.1148/radiol.2017161659 (PMID: 28240562) doi: 10.1148/radiol.2017161659

Mahsereci, M., Balles, L., Lassner, C., & Hennig, P. (2017). Early stopping without a validation set. *CoRR*, *abs/1703.09580*. Retrieved from http://arxiv.org/abs/1703.09580

Manzanera, O. M., Meles, S. K., Leenders, K. L., Renken, R. J., Pagani, M., Arnaldi, D., ... Maurits, N. M. (2019). Scaled subprofile modeling and convolutional neural networks for the identification of parkinson's disease in 3d nuclear imaging data. *International Journal of Neural Systems*, *29*(09), 1950010. Retrieved from https://doi.org/10.1142/S0129065719500102 (PMID: 31046514) doi: 10.1142/S0129065719500102

Milletari, F., Navab, N., & Ahmadi, S. (2016, Oct). V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3d vision (3dv)* (p. 565-571). doi: 10.1109/3DV.2016.79

Moons, K. G. M., de Groot, J. A. H., Bouwmeester, W., Vergouwe, Y., Mallett, S., Altman, D. G., ... Collins, G. S. (2014, 10). Critical appraisal and data extraction for systematic reviews of prediction modelling studies: The charms checklist. *PLOS Medicine*, *11*(10), 1-12. Retrieved from https://doi.org/10.1371/journal.pmed.1001744 doi: 10.1371/journal.pmed.1001744

Ng, A. Y. (1997). Preventing" overfitting" of cross-validation data. In *Icml* (Vol. 97, pp. 245–253).

Nobashi, T., Zacharias, C., Ellis, J. K., Ferri, V., Koran, M. E., Franc, B. L., ... Davidzon, G. A. (2019, Oct 28). Performance comparison of individual and ensemble cnn models for the classification of brain 18f-fdg-pet scans. *Journal of Digital Imaging*. Retrieved from https://doi.org/10.1007/s10278-019-00289-x doi: 10.1007/s10278-019-00289-x

O, J. H., Lodge, M. A., & Wahl, R. L. (2016). Practical PERCIST: A Simplified Guide to PET Response Criteria in Solid Tumors 1.0. *Radiology*. doi: 10.1148/radiol.2016142043

Patel, V. K., Naik, S. K., Naidich, D. P., Travis, W. D., Weingarten, J. A., Lazzaro, R., ... Raoof, S. (2013, Mar 01). A practical algorithmic approach to the diagnosis and management of solitary pulmonary nodules: Part 1: Radiologic characteristics and imaging modalities. *CHEST*, *143*(3), 825-839. Retrieved from https://doi.org/10.1378/chest.12

-0960 doi: 10.1378/chest.12-0960

Pau, G., Fuchs, F., Sklyar, O., Boutros, M., & Huber, W. (2010). Ebimage—an r package for image processing with applications to cellular phenotypes. *Bioinformatics*, *26*(7), 979–981. doi: 10.1093/bioinformatics/btq046

Prechelt, L. (2012). Early stopping — but when? In G. Montavon, G. B. Orr, & K.-R. Müller (Eds.), *Neural networks: Tricks of the trade: Second edition* (pp. 53–67). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from https://doi.org/10.1007/978-3-642-35289-8_5 doi: 10.1007/978-3-642-35289-8_5

R Core Team. (2019). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from https://www.R-project.org/

Rajpurkar, P., Irvin, J., Ball, R. L., Zhu, K., Yang, B., Mehta, H., ... Lungren, M. P. (2018, 11). Deep learning for chest radiograph diagnosis: A retrospective comparison of the chexnext algorithm to practicing radiologists. *PLOS Medicine*, *15*(11), 1-17. Retrieved from https://doi.org/10.1371/journal.pmed.1002686 doi: 10.1371/journal.pmed.1002686

Rao, C., & Liu, Y. (2020). Three-dimensional convolutional neural network (3d-cnn) for heterogeneous material homogenization. *Computational Materials Science*, *184*, 109850. Retrieved from http://www.sciencedirect.com/science/article/pii/S0927025620303414 doi: https://doi.org/10.1016/j.commatsci.2020.109850

Rawat, W., & Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, *29*(9), 2352-2449. (PMID: 28599112) doi: 10.1162/neco_a_00990

Reginelli, A., Capasso, R., Petrillo, M., Rossi, C., Faella, P., Grassi, R., ... Bouros, D. (2019). Looking for Lepidic Component inside Invasive Adenocarcinomas Appearing as CT Solid Solitary Pulmonary Nodules (SPNs): CT Morpho-Densitometric Features and 18-FDG PET Findings. *BioMed Research International*, *2019*, 1-9. Retrieved from https://doi.org/10.1155/2019/7683648 doi: doi.org/10.1155/2019/7683648

Ricciardi, S., Davini, F., Manca, G., De Liperi, A., Romano, G., Zirafa, C. C., & Melfi, F. (2020, Sep 01). Radioguided surgery, a cost-effective strategy for treating solitary pulmonary nodules: 20-year experience of a single center. *Clinical Lung Cancer*, *21*(5), e417-e422. Retrieved from https://doi.org/10.1016/j.cllc.2020.02.026 doi: 10.1016/j.cllc.2020.02.026

Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, J.-C., & Müller, M. (2011). proc: an open-source package for r and s+ to analyze and compare roc curves. *BMC Bioinformatics*, *12*, 77.

Ruder, S. (2017). *An overview of gradient descent optimization algorithms.*

Ruilong, Z., Daohai, X., Li, G., Xiaohong, W., Chunjie, W., & Lei, T. (2017). Diagnostic value of 18F-FDG-PET/CT for the evaluation of solitary pulmonary nodules: a systematic review and meta-analysis. *Nuclear Medicine Communications*, *38*(1), 67-75. Retrieved from https://www.ingentaconnect.com/content/wk/numec/2017/00000038/00000001/art00010 doi: doi:10.1097/MNM.0000000000000605

Schmidt, R. L., & Factor, R. E. (2013, Apr 01). Understanding sources of bias in diagnostic accuracy studies. *Archives of Pathology & Laboratory Medicine*, *137*(4), 558-565. Retrieved from https://doi.org/10.5858/arpa.2012-0198-RA doi: 10.5858/arpa.2012-0198-RA

Shen, W.-C., Chen, S.-W., Wu, K.-C., Hsieh, T.-C., Liang, J.-A., Hung, Y.-C., ... Kao, C.-H. (2019). Prediction of local relapse and distant metastasis in patients with definitive chemoradiotherapy-treated cervical cancer by deep learning from [18f]-fluorodeoxyglucose positron emission tomography/computed tomography. *European Radiology*, *29*(12), 6741-6749. Retrieved from https://doi.org/10.1007/s00330-019-06265-x doi: 10.1007/s00330-019-06265-x

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, *6*(1), 60. Retrieved from https://doi.org/10.1186/s40537-019-0197-0 doi: 10.1186/s40537-019-0197-0

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In Y. Bengio & Y. LeCun (Eds.), *3rd international conference on learning representations, ICLR 2015, san diego, ca, usa, may 7-9, 2015, conference track proceedings.* Retrieved from http://arxiv.org/abs/1409.1556

Singh, A., Sengupta, S., & Lakshminarayanan, V. (2020). Explainable deep learning models in medical image analysis. *Journal of Imaging*, *6*(6). Retrieved from https://www.mdpi.com/2313-433X/6/6/52 doi: 10.3390/jimaging6060052

Smits, N. (2010, Sep 30). A note on youden's jand its cost ratio. *BMC Medical Research Methodology*, *10*(1), 89. Retrieved from https://doi.org/10.1186/1471-2288-10-89 doi: 10.1186/1471-2288-10-89

Sokolova, M., Japkowicz, N., & Szpakowicz, S. (2006). Beyond accuracy, f-score and roc: A family of discriminant measures for performance evaluation. In A. Sattar & B.-h. Kang (Eds.), *Ai 2006: Advances in artificial intelligence* (pp. 1015–1021). Berlin, Heidelberg: Springer Berlin Heidelberg.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learn-*

ing Research, *15*, 1929-1958. Retrieved from http://jmlr.org/papers/v15/srivastava14a.html

Sung, H., Ferlay, J., Siegel, R. L., Laversanne, M., Soerjomataram, I., Jemal, A., & Bray, F. (2021). Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians*, *71*(3), 209-249. Retrieved from https://acsjournals.onlinelibrary.wiley.com/doi/abs/10.3322/caac.21660 doi: https://doi.org/10.3322/caac.21660

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... Rabinovich, A. (2014). *Going deeper with convolutions.*

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015). Rethinking the inception architecture for computer vision. *CoRR*, *abs/1512.00567*. Retrieved from http://arxiv.org/abs/1512.00567

Tajbakhsh, N., Shin, J. Y., Gurudu, S. R., Hurst, R. T., Kendall, C. B., Gotway, M. B., & Liang, J. (2016, May). Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE Transactions on Medical Imaging*, *35*(5), 1299-1312. doi: 10.1109/TMI.2016.2535302

Teramoto, A., Tsujimoto, M., Inoue, T., Tsukamoto, T., Imaizumi, K., Toyama, H., ... Fujita, H. (2019). Automated classification of pulmonary nodules through a retrospective analysis of conventional ct and two-phase pet images in patients undergoing biopsy. *Asia Oceania journal of nuclear medicine & biology*, *7*(1), 29-37. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/30705909 (30705909[pmid])

Teymurazyan, A., Riauka, T., Jans, H.-S., & Robinson, D. (2013, Jun 01). Properties of noise in positron emission tomography images reconstructed with filtered-backprojection and row-action maximum likelihood algorithm. *Journal of Digital Imaging*, *26*(3), 447-456. Retrieved from https://doi.org/10.1007/s10278-012-9511-5 doi: 10.1007/s10278-012-9511-5

Togo, R., Hirata, K., Manabe, O., Ohira, H., Tsujino, I., Magota, K., ... Shiga, T. (2019). Cardiac sarcoidosis classification with deep convolutional neural network-based features using polar maps. *Computers in Biology and Medicine*, *104*, 81 - 86. Retrieved from http://www.sciencedirect.com/science/article/pii/S0010482518303640 doi: https://doi.org/10.1016/j.compbiomed.2018.11.008

ur Rehman, S., Tu, S., Waqas, M., Huang, Y., ur Rehman, O., Ahmad, B., & Ahmad, S. (2019). Unsupervised pre-trained filter learning approach for efficient convolution neural network. *Neurocomputing*, *365*, 171 - 190. Retrieved from http://www.sciencedirect.com/science/article/pii/S0925231219309981 doi: https://doi.org/10.1016/j.neucom.2019.06.084

Ushey, K., Allaire, J., & Tang, Y. (2020). reticulate: Interface to 'python' [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=reticulate (R package version 1.16)

Vapnik, V. N. (1999, September). An overview of statistical learning theory. *Trans. Neur. Netw.*, *10*(5), 988–999. Retrieved from https://doi.org/10.1109/72.788640 doi: 10.1109/72.788640

Venkatraman, E. S., & Begg, C. B. (1996, 12). A distribution-free procedure for comparing receiver operating characteristic curves from a paired experiment. *Biometrika*, *83*(4), 835-848. Retrieved from https://doi.org/10.1093/biomet/83.4.835 doi: 10.1093/biomet/83.4.835

Wainer, J., & Cawley, G. (2021). Nested cross-validation when selecting classifiers is overzealous for most practical applications. *Expert Systems with Applications*, *182*, 115222. Retrieved from https://www.sciencedirect.com/science/article/pii/S0957417421006540 doi: https://doi.org/10.1016/j.eswa.2021.115222

Wang, H., Zhou, Z., Li, Y., Chen, Z., Lu, P., Wang, W., ... Yu, L. (2017). Comparison of machine learning methods for classifying mediastinal lymph node metastasis of non-small cell lung cancer from 18f-fdg pet/ct images. *EJNMMI Research*, *7*(1), 11. Retrieved from https://doi.org/10.1186/s13550-017-0260-9 doi: 10.1186/s13550-017-0260-9

Wang, Q., Ma, Y., Zhao, K., & Tian, Y. (2020, Apr 12). A comprehensive survey of loss functions in machine learning. *Annals of Data Science*. Retrieved from https://doi.org/10.1007/s40745-020-00253-5 doi: 10.1007/s40745-020-00253-5

Weinstein, S., Obuchowski, N. A., & Lieber, M. L. (2005, Jan 01). Clinical evaluation of diagnostic tests. *American Journal of Roentgenology*, *184*(1), 14-19. Retrieved from https://doi.org/10.2214/ajr.184.1.01840014 doi: 10.2214/ajr.184.1.01840014

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of Big Data*, *3*(1), 9. Retrieved from https://doi.org/10.1186/s40537-016-0043-6 doi: 10.1186/s40537-016-0043-6

Woodard, G. A., Jones, K. D., & Jablons, D. M. (2016). Lung cancer staging and prognosis. In K. L. Reckamp (Ed.), *Lung cancer: Treatment and research* (pp. 47–75). Cham: Springer International Publishing. Retrieved from https://doi.org/10.1007/978-3-319-40389-2_3 doi: 10.1007/978-3-319-40389-2_3

Yang, C.-K., Yeh, J. C.-Y., Yu, W.-H., Chien, L.-I., Lin, K.-H., Huang, W.-S., & Hsu, P.-K. (2019, Jun 13). Deep convolutional neural network-based positron emission tomography analysis predicts esophageal cancer outcome. *Journal of clinical medicine*, *8*(6),

844. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/31200519 doi: 10.3390/jcm8060844

Yang, Y., Feng, X., Chi, W., Li, Z., Duan, W., Liu, H., ... Liu, B. (2018). Deep learning aided decision support for pulmonary nodules diagnosing: a review. *Journal of Thoracic Disease*, *10*(7). Retrieved from http://jtd.amegroups.com/article/view/19479

Yee, E., Popuri, K., Beg, M. F., & the Alzheimer's Disease Neuroimaging Initiative. (2020). Quantifying brain metabolism from fdg-pet images into a probability of alzheimer's dementia score. *Human Brain Mapping*, *41*(1), 5-16. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1002/hbm.24783 doi: 10.1002/hbm.24783

Ypsilantis, P.-P., Siddique, M., Sohn, H.-M., Davies, A., Cook, G., Goh, V., & Montana, G. (2015, 09). Predicting response to neoadjuvant chemotherapy with pet imaging using convolutional neural networks. *PLOS ONE*, *10*(9), 1-18. Retrieved from https://doi.org/10.1371/journal.pone.0137036 doi: 10.1371/journal.pone.0137036

Zhang, G. P. (2000, Nov). Neural networks for classification: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *30*(4), 451-462. doi: 10.1109/5326.897072

Zhou, Z., Chen, L., Sher, D., Zhang, Q., Shah, J., Pham, N., ... Wang, J. (2018, July). Predicting lymph node metastasis in head and neck cancer by combining many-objective radiomics and 3-dimensioal convolutional neural network through evidential reasoning*. In *2018 40th annual international conference of the ieee engineering in medicine and biology society (embc)* (p. 1-4). doi: 10.1109/EMBC.2018.8513070