

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
PRÓ-REITORIA DE PESQUISA, PÓS-GRADUAÇÃO E INOVAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Raul Victor Medeiros da Nóbrega

**Lung Nodule Malignancy Classification in Chest Computed
Tomography Images Using Transfer Learning And Convolutional
Neural Networks**

Fortaleza – CE

19 de setembro de 2018

Dados Internacionais de Catalogação na Publicação
Instituto Federal do Ceará - IFCE
Sistema de Bibliotecas - SIBI
Ficha catalográfica elaborada pelo SIBI/IFCE, com os dados fornecidos pelo(a) autor(a)

Nóbrega, Raul.

Lung Nodule Malignancy Classification in Chest Computed Tomography Images Using Transfer Learning And Convolutional Neural Networks / Raul Nóbrega. - 2018.
81 f. : il. color.

Dissertação (Mestrado) - Instituto Federal do Ceará, Mestrado em Ciência da Computação, Campus Fortaleza, 2018.

Orientação: Prof. Dr. Pedro Pedrosa Rebouças Filho .

1. Visão Computacional. 2. Engenharia Biomédica. 3. Ferramenta de Auxílio ao diagnóstico Médico. 4. Redes Neurais Convolutivas. I. Título.

CDD 004

Raul Victor Medeiros da Nóbrega

Lung Nodule Malignancy Classification in Chest Computed Tomography Images Using Transfer Learning And Convolutional Neural Networks

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Linha de Pesquisa: Inteligência Artificial

Supervisor: Prof. Dr. Pedro Pedrosa Rebouças Filho

Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)

Pró-Reitoria de Pesquisa, Pós-Graduação e Inovação (PRPI)

Programa de Pós-Graduação em Ciência da Computação (PPGCC)

Fortaleza – CE

19 de setembro de 2018



Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)
Pró-Reitoria de Pesquisa, Pós-Graduação e Inovação (PRPI)
Programa de Pós-Graduação em Ciência da Computação (PPGCC)

Raul Victor Medeiros da Nóbrega

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação, sendo aprovada pela Coordenação do Programa de Pós-Graduação em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará e pela banca examinadora:

**Supervisor: Prof. Dr. Pedro Pedrosa Rebouças
Filho**

Instituto Federal de Educação, Ciência e Tecnologia do
Ceará (IFCE)

Prof. Dr. Amauri Holanda de Souza Júnior
Instituto Federal de Educação, Ciência e Tecnologia do
Ceará (IFCE)

Prof. Dr. Tiago Carneiro Pessoa
Instituto Federal de Educação, Ciência e Tecnologia do
Ceará (IFCE)

Prof. Dr. Antônio Carlos da Silva Barros
Universidade da Integração Internacional da Lusofonia
Afro-Brasileira (UNILAB)

Fortaleza – CE
19 de setembro de 2018

Acknowledgements

To my parents (Maria Inês Medeiros and Aluizio Moises de Medeiros), for giving me the love, confidence and structure to reach my dreams.

To my supervisor Prof. Dr. Pedro Pedrosa Rebouças Filho for his dedication, commitment and patience.

To the professors and colleagues of the Programa de Pós-Graduação em Ciência da Computação do Instituto Federal do Ceará (PPGCC-IFCE) who contributed directly to my professional and personal development.

To the Instituto Federal de Educação, Ciências e Tecnologia do Ceará (IFCE) and all the professionals that compose it for providing an adequate environment for the intellectual and social development of students.

The research institution CAPES that allowed me to see in science not only a source of income, but my mission in this life.

*“Disciplina é liberdade
Compaixão é fortaleza
Ter bondade é ter coragem.”
(Renato Russo)*

Abstract

Lung cancer corresponds to 26% of all deaths due to cancer in 2017, accounting for more than 1.5 million deaths globally. Considering this challenging situation, several computer-aided diagnosis(CAD) systems have been developed to detect lung cancer at early stages, which increases the patients' survival rate. Motivated by the success of deep learning in natural and medical image classification tasks, the main objective of this work is to explore the performance of deep transfer learning from non-medical images on the lung nodules malignancy classification task. More specifically, this task is represented in this work by a binary classification problem using the Lung Image Database Consortium and Image Database Resource Initiative (LIDC/IDRI) dataset. To accomplish that, the methodology applied in this work is divided into 4 stages. First, the LIDC/IDRI has its 1018 chest Computed Tomography (CT) exams and medical annotations processed. Then, several Convolutional Neural Networks (CNN), such as VGG16, VGG19, MobileNet, Xception, InceptionV3, ResNet50, Inception-ResNet-V2, DenseNet169, DenseNet201, NASNetMobile and NASNetLarge models, are built, trained on ImageNet dataset, converted into feature extractors and applied on the LIDC/IDRI nodule images. Second, each set of deep features is submitted to 10-Fold Cross Validations with Naive Bayes, Multilayer Perceptron (MLP), Support Vector Machine (SVM), K-Nearest Neighbors (KNN) and Random Forest (RF) classifiers. Third, each Cross Validation average result has its evaluation metrics Accuracy (ACC), Area Under the Curve (AUC), True Positive Rate (TPR), Precision (PPV), and F1-Score computed and compared. Thus, according to the results, the Deep Feature Extractor based on ResNet50 model combined with a SVM classifier (using a Radial Basis Function (RBF) kernel), achieve an AUC metric of 93.19% (the highest value not only among the evaluated combinations, but also among the related works evaluated), a TPR of 85.38%, an ACC of 88.41%, a PPV of 73.48% and a F1-Score of 78.83%. Based on these results, deep transfer learning from non-medical images proves to be a relevant strategy to extract representative features for lung nodule malignancy classification in chest CT images. Overall, the combination of ResNet50 and SVM-RBF achieve satisfactory results and is eligible to be integrated into a CAD system.

Keywords: Lung Nodule Malignancy Classification, Convolutional Neural Networks, Transfer Learning, Computer-Aided Diagnoses, Computed Tomography.

List of Figures

Figure 1 – Architecture of LeNet-5 Convolutional Neural Network (LECUN et al., 1998).	33
Figure 2 – Illustration of a 2D Max pooling operation (KARPATHY; LI, 2017)	34
Figure 3 – The Rectified Linear Unit (ReLU) activation function (https://goo.gl/FLHUrZ)	35
Figure 4 – The illustration in item (a) represents the conventional information flow during a forward pass of a network and in the item (b) this flow is interrupted (set to zero) at some neuron, this process is called dropout (SRIVASTAVA et al., 2014).	36
Figure 5 – The effective receptive field of a stack of 2 3x3 convolutions (SIMONYAN; ZISSERMAN, 2014)	37
Figure 6 – Two Inception architecture illustrations. In item (a) the naive Inception Module; In item (b) the Inception Module with dimensionality reduction.	38
Figure 7 – Batch normalizing transform, applied to activation x over a mini-batch (Ioffe; Szegedy, 2015).	39
Figure 8 – An architecture illustration of an Inception Module with asymmetric convolutions (SZEGEDY et al., 2016).	40
Figure 9 – An example of a Residual block architecture (WU; ZHONG; LIU, 2017).	41
Figure 10 – An illustration of a general architecture layout when combining Residual Connections with Inception modules (LÄNGKVIST; KARLSSON; LOUTFI, 2014).	42
Figure 11 – Illustration of a general architecture layout when combining Residual Connections with Inception modules (HUANG et al., 2017).	44
Figure 12 – Illustration of a general architecture layout when combining Residual Connections with Inception modules (ZOPH; LE, 2016).	45
Figure 13 – In yellow the sequence of steps performed to process the LIDC/IDRI dataset. The blue branch outlines the procedures to build a deep feature extractor. The green branch is how the proposed approach is evaluated.	47
Figure 14 – Illustration of the first step of medical annotation processing, where X , Y , Z are the centroid coordinates; R the radius; and L the malignancy level of a given nodule.	49
Figure 15 – Illustration of the medical annotations processing second step, the treatment of redundant annotations.	49
Figure 16 – Illustration of the volumes of interest extraction process.	49
Figure 17 – Converting a extracted volume of interest into a 3 channel image.	50
Figure 18 – Illustration of the nodule malignancy levels distribution before and after the reduction process.	51

Figure 19 – Examples of non-medical images from the reference dataset (used to train 11 CNNs explored in this work).	53
Figure 20 – Illustration of the first two steps of a CNN. In the first, the input image is submitted to a sequence of Convolutional and Subsampling (Pooling) layers. In the second, the results of the first step are resized and concatenated into a single list, called here 'deep features'	54
Figure 21 – Illustration of the process used to transform a CNN into a feature extractor.	54
Figure 22 – Illustration of a deep feature extraction process in a given dataset. . . .	55
Figure 23 – Illustration of the 10-Fold Cross Validation process in a shuffled dataset.	56
Figure 24 – Illustration of a confusion matrix for the lung nodule malignancy classification task performed in this work.	57
Figure 25 – Prediction probability histogram of a binary problem. Where TP, TN, FP and FN are true positive, true negative, false positive and false negative, respectively. T is the threshold between what is classified as class 0 and class 1.	58
Figure 26 – Illustration of a receiver operating characteristic curve (goo.gl/oLLnBz).	59
Figure 27 – Comparison of the 5 experiments with highest average between AUC, ACC, TPR, F-Score and PPV evaluation metrics. Each of these experiments is identified by a different color and represents the combination of a CNN-based feature extractor and a machine learning technique. . . .	63
Figure 28 – Comparison between the 5 experiments with the highest average of the AUC and TPR evaluation metrics. Each of these experiments is identified by a different color and represents the combination of a CNN-based feature extractor and a machine learning technique.	64
Figure 29 – The average execution time (in milliseconds) of each Deep feature extractor explored in this work.	65
Figure 30 – The values of AUC, ACC, TPR, F-Score, and PPV metrics from the selected combination.	65
Figure 31 – List of related works sorted by the number of nodules evaluated.	66
Figure 32 – The results from the proposed approach and related works are compared in a bar chart using the AUC, ACC, TPR, metrics. Each one of these metrics is indicated by a color.	67

List of Tables

Table 1 – Accuracy (ACC), Area Under Curve (AUC), F-Score, Sensitivity (TPR), Positive Predictive Value (PPV) for the evaluated combinations between deep feature extractors and classifiers.	62
---	----

List of abbreviations and acronyms

CNN	Convolutional Neural Networks
SVM	Support Vector Machine
MLP	Multilayer Perceptron
RF	Random Forest
CV	Computer Vision
kNN	k-Nearest Neighbor
LBP	Local Binary Patterns
CAD	Computer-aided diagnosis
PPV	Positive Predictive Value
TPR	True Positive Rate
ACC	Accuracy
CT	Computer Tomography
AUC	Area Under ROC-Curve
DIP	Digital Image Processing
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
SCM	Structural Co-occurrence Matrix

Contents

1	INTRODUCTION	19
1.1	Motivation	20
1.2	Related Works	21
1.3	Objectives	22
1.3.1	Secondary Objectives	22
1.4	Contributions	23
1.5	Publications	23
1.6	Outline	23
2	CLASSICAL MACHINE LEARNING TECHNIQUES	25
2.1	Naive Bayes Classifier	25
2.2	<i>k</i>-Nearest Neighbor (kNN)	26
2.3	Support Vector Machines (SVM)	27
2.4	Random Forest	29
2.5	Multilayer Perceptron (MLP)	30
3	CONVOLUTIONAL NEURAL NETWORK	33
3.1	Lenet-5	33
3.1.1	Convolutional Layers	33
3.1.2	Pooling Layers	34
3.2	AlexNet	34
3.2.1	ReLU Nonlinearity	35
3.2.2	Dropout	35
3.3	VGG	36
3.3.1	Factorized Convolutions	37
3.4	Inception	38
3.4.1	Inception Module	38
3.4.2	Batch-normalization	39
3.4.3	Asymmetric Convolutions	39
3.5	ResNet	40
3.5.1	Residual Block	41
3.6	Inception-ResNet	41
3.6.1	Inception Module with Residual connections	41
3.7	Xception	42
3.7.1	Depthwise Separable Convolutions	42
3.8	MobileNet	43

3.8.1	Width Multipliers	43
3.8.2	Resolution Multipliers	43
3.9	DenseNet	44
3.9.1	Dense Blocks	44
3.10	NASNet	45
3.10.1	NASNet search space	45
4	METHODOLOGY	47
4.1	Processing the LIDC/IDRI database	48
4.1.1	Medical Annotations Processing	48
4.1.2	Problem Reduction	50
4.2	Convolutional Neural Networks as feature extractors	51
4.2.1	Architectures Construction and Initialization	51
4.2.2	Architectures Training	52
4.2.3	Converting CNNs into feature extractors	52
4.3	Evaluating the proposed approach	53
4.3.1	Extracting features from LIDC/IDRI database	53
4.3.2	Malignancy Classification	55
4.3.3	Evaluation metrics	56
4.3.4	Hardware Settings	59
5	RESULTS	61
5.1	Evaluating Explored Combinations	61
5.2	Comparison of the proposed approach with related works	65
6	CONCLUSION	69
6.1	Future Works	70
	BIBLIOGRAPHY	71
	APPENDIX	77
	APPENDIX A – THE LUNG IMAGE DATABASE CONSORTIUM (LIDC) AND IMAGE DATABASE RESOURCE INITIATIVE (IDRI)	79

1 Introduction

The integration between computation and medicine research fields enables the development of tools to aid the diagnosis of diseases and to monitor their treatments. Among these tools, the ones that process medical images using Computer Vision systems stand out (SILVA, 2004; LI; LIU; FOX, 2005; FELIX, CORTEZ; HOLANDA, 2010; REBOUÇAS FILHO, 2013).

The use of medical images is widely disseminated for the acquisition and representation of information about the human body. Further, Computed Tomography (CT) is one of the most used techniques to acquire these images (HERMAN, 2009; REBOUÇAS FILHO, 2013).

Computed Tomography consists in constructing a model of the body under analysis, through a mapping of the radioactive attenuation coefficient that transcends it. Developed by Godfrey Hounsfield (1973), the tomography is the equipment that makes the CT images acquisition. Such a machine computes images in cross sections, performing, on them, extremely complex mathematical operations, thus generating exams with great accuracy and in a timely manner to medical applications (KALENDER, 2006; REBOUÇAS FILHO, 2010).

Computer Vision systems are commonly based on Digital Image Processing and Machine Learning algorithms (REBOUÇAS FILHO, 2013). Moreover, these systems are applied to the detection, recognition, and classification of objects, shapes, and scenes (GONZALEZ; WOODS, 2010). More specifically, Computer Vision systems are represented by five consecutive steps: acquisition, preprocessing, segmentation, attribute extraction, and interpretation (GONZALEZ; WOODS, 2010; REBOUÇAS FILHO, 2013).

The first step, the acquisition, is defined as the process of capturing the image that you want to analyze. (GONZALEZ; WOODS, 2010). The main purpose of the preprocessing stage is to perform enhancement, noise attenuation or image restoration operations, generally applying Digital Image Processing techniques (ALEXANDRIA, 2005; GONZALEZ; WOODS, 2010). The segmentation stage consists of dividing the image into regions of interest (REBOUÇAS FILHO, 2010). Next, the feature extraction stage aims to identify a set of information describing each segmented object in the previous step (NIXON, AGUADO, 2012). Finally, the interpretation step consists in assigning meaning to a set of recognized objects by analyzing their previously extracted characteristics (SONKA; HLAVAC; BOYLE, 2008). In the context of Artificial Intelligence this stage is known as classification.

This dissertation, in turn, focuses on the feature extraction and classification steps

of a Computer Vision system that aims to aid in the diagnosis of malignancy of pulmonary nodules in CT images.

These stages are essential in a Computer Vision system that aims at assigning diagnosis of pulmonary nodules.

The classification accuracy is a critical factor for medical applications. Because, assigning a malignant nodule as a non-malignant one or vice versa could, not only the reduce the reliability (and consequently the adoption) of such system, but also compromise the specialist physician analysis.

In this sense, to perform lung nodule malignancy classification task, several feature extractors and machine learning techniques are implemented and analyzed in this dissertation.

1.1 Motivation

Although lung cancer mortality rate reduced 17% (from 2002 to 2014) among females and 43% (from 1990 to 2014) among males (SIEGEL; MILLER; JEMAL, 2017), lung cancer still accounts for 1.59 million deaths worldwide (HUSSEIN et al., 2017), corresponding to 26% of all deaths due to cancer in 2017 (SIEGEL; MILLER; JEMAL, 2017). Also, since more than 50% of lung cancer patients are diagnosed at a distant stage, for which the five-year survival rate is only 4% (HUSSEIN et al., 2017), lung cancer is considered an aggressive disease (SHEN et al., 2017).

However, to address this high mortality rate, pulmonologist and radiologists started to use chest CT images to diagnose lung cancers at an earlier stage (SIEGEL; MILLER; JEMAL, 2017). The use of these images have proved to be a valuable source of information, reducing lung cancer mortality by up to 20% in a high-risk group (MARCUS et al., 2016; SIEGEL; MILLER; JEMAL, 2017).

Moreover, most of modern lung Computer-Aided Diagnosis (CAD) systems are based on Computer Vision systems and are used to perform an automatic assessment of pulmonary nodules in chest CT images, and consequently, support medical decision-making. The ultimate goal of such lung CAD systems is to automatically discriminate between malignant and non-malignant nodules (SHEN et al., 2015).

In order to achieve this, such systems need to extract representative imaging biomarkers at a macroscopic level in chest CT images to perform a proper classification. The rising quantity and quality of public-available datasets made it possible for data-driven methods, e.g., Convolutional Neural Network (CNN), learn how to extract representative features from raw data (LECUN et al., 1990).

However, to perform proper training on CNNs deep models a large amount of

data is necessary, but this remains a challenge in the medical domain (SHIN et al., 2016; CHRISTODOULIDIS et al., 2017). To overcome this, a strategy called transfer learning suggests that features learned to solve a specific task can be useful for tasks from other related domains (PAN; YANG, 2010; SHIE et al., 2015; NÄPPI et al., 2016).

Finally, motivated by the success of deep transfer learning in computer-aided diagnosis systems (SHIE et al., 2015; HUYNH; LI; GIGER, 2016; HWANG et al., 2016; NÄPPI et al., 2016; SHIN et al., 2016; CHRISTODOULIDIS et al., 2017; ESTEVA et al., 2017; HUSSEIN et al., 2017; LITJENS et al., 2017), this work addresses the challenge of lung nodules malignancy classification in chest CT images using deep features.

1.2 Related Works

In order to extract representative imaging biomarkers at a macroscopic level in chest CT images, several methods have been proposed. Overall, these extractors can be divided into two categories: hand-crafted, and data-driven.

Several studies investigating a wide variety of hand-crafted features extractors for lung nodule classification have been carried out. Among the explored extractors are:

- Local Binary Pattern (LBP) (HAN et al., 2015; SHEN et al., 2015; SHEN et al., 2017; WEI et al., 2018),
- Gabor (HAN et al., 2015; WEI et al., 2018),
- Histogram of Oriented Gradients (HOG) (SHEN et al., 2015; SHEN et al., 2017; DHARA et al., 2016),
- Gray-Level Co-occurrence Matrix (GLCM) (HAN et al., 2015; DHARA et al., 2016; WEI et al., 2017; WEI et al., 2018; SERGEEVA et al., 2016),
- Structural co-occurrence matrix (SCM) (RODRIGUES et al., 2018) and many others.

Besides, Aerts et al. (2014) approach combines hand-crafted with morphologic features. More specifically, in Shen et al. (2017) implementation of this method, they compute first-order statistics, shape and size features, textural features, and wavelet features, which account for 319 features, and then, select the top 54 features using mRMR score (PENG; LONG; DING, 2005). These features are then used to train an SVM classifier.

However, with the advent of deep learning, there has been a rapid rise in the use of data-driven methods for medical images processing. For example, Shen et al. (2015) train a CNN with 3D patches of the LIDC/IDRI nodules images. These patches are generated by cropping Volumes Of Interest (VOIs) of a given nodule with three different resolutions, and then, reshaping each one to 32x32x32 pixels. After that, this trained CNN is transformed into a feature extractor and applied on all patches. Next, the features extracted from a

given nodule are concatenated, composing a new dataset. Finally, they train an SVM and a Random Forest classifier on this dataset.

Later, Shen et al. (2017) present the Multi-Scale Convolutional Neural Networks, (MCNN), which can be roughly described as a 6-layer CNN where the max-pooling layers are replaced by the proposed multi-crop layers. This multi-crop layer performs a depthwise concatenation of 3 max-pooling operations applied on concentric crops of the input feature maps. Inspired by the performance of dual path network (DPN) on ImageNet (DENG et al., 2009; SHEN et al., 2017; ZHU et al., 2017) proposed an approach based on a gradient boosting machine (GBM) with 3D DPN.

Moreover, Hussein et al. (2017) uses malignancy, calcification, sphericity, margin, lobulation, spiculation and texture medical annotations as labels to fine-tune 3D CNNs pre-trained on Sports-1M dataset (KARPATHY et al., 2014).

1.3 Objectives

The primary goal of this work is to explore deep learning models as feature extractors to tackle the challenging problem of lung nodule malignancy classification in chest CT images.

1.3.1 Secondary Objectives

More specifically, the following procedures are performed in this work:

- Process the exams and medical annotations from the Lung Image Database Consortium image collection (LIDC-IDRI)(ARMATO et al., 2011) to compose a set of lung nodules images and their respective malignancy information.
- Extract deep features from target dataset, using 11 different CNNs models pre-trained on ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) such as: VGG16 (SIMONYAN; ZISSERMAN, 2014), VGG19 (SIMONYAN; ZISSERMAN, 2014), MobileNet (HOWARD et al., 2017), Xception (CHOLLET, 2017), InceptionV3 (SZEGEDY et al., 2016), ResNet50 (HE et al., 2016), Inception-ResNet-V2 (SZEGEDY et al., 2017), DenseNet169, DenseNet201 (HUANG et al., 2017) and NASNetMobile and NASNetLarge (ZOPH et al., 2017).
- Apply 5 classification methods, such as: Naive Bayes (THEODORIDIS; KOUTROUMBAS, 2008), Multilayer Perceptron (MLP) (HAYKIN, 2008), Support Vector Machine (SVM) (VAPNIK, 1998), Nearest Neighbours (KNN) (COVER; HART, 1967) and Random Forest (RF) (LIAW; WIENER et al., 2002), for each extracted features set.

- Compare the results among themselves and with related works through 5 evaluation metrics: Accuracy (ACC), Area Under the Curve (AUC), True Positive Rate (TPR), Precision (PPV) and F1-Score.

1.4 Contributions

The main contributions of this dissertation are:

- a detailed methodology for processing the exams and medical annotations from the LIDC/IDRI database;
- the exploration of several CNN architectures as feature extractors (based on the Transfer of Learning concept);
- the results of the experiments performed in this work were submitted to an extensive statistical analysis, thus serving as baseline results for future works;
- the proposition of an effective approach for lung nodule malignancy classification in chest Computed Tomography images.

1.5 Publications

During this masters project, the following scientific papers were accepted and published, namely:

- (IEEE CBMS 2018) - Lung Nodule Classification via Deep Transfer Learning in CT Lung Images - 1^o Author - Qualis B1 (CC);
- (IEEE Access 2018) - Health of Things Algorithms for Malignancy Level Classification of Lung Nodules - 2^o Author - Qualis A2 (Eng. IV);
- (IJCNN 2018) - Localization of Mobile Robots with Topological Maps and Classification with Reject Option using Convolutional Neural Networks in Omnidirectional Images - 2^o Author - Qualis A1 (CC);
- (IEEE CBMS 2017) - Segmentation and Visualization of the Lungs in Three Dimensions Using 3D Region Growing and Visualization Toolkit in CT Examinations of the Chest - 1^o Author - Qualis B1 (CC);
- (ISDA 2017) - Lung Segmentation in Chest Computerized Tomography Images Using the Border Following Algorithm - 3^o Author - Qualis B1 (CC).

1.6 Outline

In addition to this introductory chapter, which contains the motivation (Section 1.1), related works (Section 1.2), objectives (Section 1.3), contributions (Section 1.4) and scientific production (Section 1.5), this dissertation has another 5 chapters. Chapter

2 presents machine learning methods consolidated in the literature, which are used to evaluate the proposed approach. Chapter 3 brings the main architectures of Convolutional Neural Networks and their fundamental concepts. Chapter 4 outlines the methodology used in this research, explaining the steps of processing the database (Section 4.1), elaborating the proposed approach (Section 4.2) and its respective evaluation (Section 4.3). Chapter 5 compares the obtained results between themselves and with related works. Finally, Chapter 6 contains the study conclusions and proposals for future works.

2 Classical Machine Learning Techniques

In this chapter, the supervised classifiers used in this dissertation are presented, Naive Bayes (Section 2.1), K-Nearest Neighbors (Section 2.2), Random Forest (Section 2.3), Support Vector Machine (Section 2.4), and Multi-layer Perceptron (Section 2.5) .

2.1 Naive Bayes Classifier

In machine learning, there are a set of simple probabilistic classifiers known as Naive Bayes. Based on Bayes' theorem, these classifiers assume class-conditional independence among features. In simple terms, a Naive Bayes classifier presumes that, for a given class, the probability of presence (or absence) of a given feature is unrelated to the presence of any other feature (THEODORIDIS; KOUTROUMBAS, 2008). More formally, Bayes' theorem states the following relationship:

$$P(c|x_1, \dots, x_k) = \frac{P(c) \cdot P(x_1, \dots, x_k|c)}{P(x_1, \dots, x_k)}, \quad (2.1)$$

where c is the class variable, $P(c)$ can be interpreted as the relative frequency of class in the training set (Prior probability) and k is the number of feature vector elements.

Further, Based on the features independence premise it can be stated that:

$$P(x_1, \dots, x_k|c) = \prod_{i=1}^k P(x_i|c). \quad (2.2)$$

Hence, Bayes' theorem can be rewired as:

$$P(c|x_1, \dots, x_k) = \frac{P(c) \cdot \prod_{i=1}^k P(x_i|c)}{P(x_1, \dots, x_k)}. \quad (2.3)$$

However, since $P(f_1, \dots, f_k)$ is constant for all possible classes, the previous equation can be further simplified into the Maximum A Posterior (MAP) decision rule, which is mathematically described as:

$$P(c|x_1, \dots, x_k) \propto P(c) \cdot \prod_{i=1}^k P(x_i|c) \quad (2.4)$$

$$\hat{c} = \underset{c}{\operatorname{argmax}} P(c) \cdot \prod_{i=1}^k P(x_i|c). \quad (2.5)$$

This means that, for example, to correctly classify a nodule image, its necessary to perform 3 steps. First, for each class, compute the product of the features occurrence probability (likelihood, eq. 2.3) . Then, multiply these likelihoods by the relative frequency (in the training set) of their respective class (prior). Finally, select the class with higher posterior (eq. 2.5).

However, in this dissertation it is assumed that the likelihood or probability density function of each feature (eq. 2.3) trends to a normal distribution distribution. This assumption is inspired by the central limit theorem, which states that the probability density function of the normalized sum of random independent variables approaches the Gaussian function (THEODORIDIS; KOUTROUMBAS, 2008). Thus, the likelihood of each feature in a Gaussian Naive Bayes classifier can formally described as:

$$P(x_i|c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} \cdot \exp\left(-\frac{(x_i - \mu_c)^2}{2\sigma_c^2}\right) \quad (2.6)$$

where, x_i is a given feature, μ_c and σ_c represent the mean and standard deviation of a given class and are computed from the instances in training set.

2.2 k -Nearest Neighbor (kNN)

The k -Nearest Neighbors (KNN) is a supervised machine learning method based on a non-parametric and instance-based learning algorithm. (DUDA; HART; STORK, 2012). By making no explicit assumptions about the data probability density function, KNN prevents, at the same time, the process of learning this distribution and the potential dangers of mismodeling it.

However, it is worth noting that, such minimal training/learning comes with considerable costs regarding memory demand and computing time. Memory demand is a critical factor because this classifier has to store a potentially high number of instances (samples with its features and labels). Also, as the whole dataset is required to predict the class of a given observation, computing time is also a critical factor for KNN algorithm.

The KNN inference process can be roughly described as majority voting process among the K training samples closest to the unseen instance, where the vote of a given training sample is its on label/class and close means lower value on a given dissimilarity metric, e.g. Euclidean, Mahalanobis, Minkowsky, Hamming distances. Euclidean distance, for example, is one of the most popular dissimilarity metrics and can be formulated as:

$$d(X, X^i) = \sqrt{(x_1 - x_1^i)^2 + (x_2 - x_2^i)^2 + \dots + (x_p - x_p^i)^2} \quad (2.7)$$

where $\mathbf{X}$ and $\mathbf{X}^i$ are two arbitrary points/instances in a p dimensional space and d is a dissimilarity metric function.

Formally, given that, $\mathbf{X}$ is an unseen observation, and $K \in \mathbb{N}^*$, the KNN classification algorithm can be divided into three stages.

In the First, KNN computes the distance between $\mathbf{X}$ and each training sample (the K training samples with lowest values of distance belong to a set $\mathbf{A}$).

In the Second, for each possible class, the conditional probability is estimated. Specifically, the probability of an unseen observation $\mathbf{X}$ been from class c , is given by the rate of samples from the subset $\mathbf{A}$ (with K elements) that have a label equals to c . This estimation process can be mathematically described as:

$$P(c) = \frac{1}{K} \cdot \sum_{i \in \mathbf{A}} I(y_i, c) \quad (2.8)$$

Where, c is a given class of the problem, y_i a given $\mathbf{A}$ sample label, $\mathbf{A}$ the K closest training samples and I is an indicator function that returns 1 when its inputs are equal and 0 otherwise.

Finally, the unseen instance is labeled with the most likely class (highest value).

2.3 Support Vector Machines (SVM)

In order to fundamentally understand the Support Vector Machines (SVM) classifier, it is important to grasp the concepts of Hyperplane Classifier, Maximal Margin Classifier and Soft Margin Classifier.

First, suppose that each instance of a given problem can be represented as a point $\mathbf{X}$ in a p dimensional space, where each dimension/axis of this space is associated with a feature of that problem. This point can be formulated as:

$$\mathbf{X} = (x_1, x_2, \dots, x_p) \quad (2.9)$$

Second, also suppose there is a $p-1$ dimensional hyperplane that perfectly separates a set of instances of a given problem by their labels. Its mathematical definition is given by the equation:

$$\beta_0 + \beta_1 \alpha_1 + \beta_2 \alpha_2 + \dots + \beta_p \alpha_p = 0. \quad (2.10)$$

where β are the hyperplane coefficients, $\alpha_1, \dots, \alpha_p \in P_O$ and P_O is given a point crossed by the hyperplane.

So, according to the relative position of a given point $\mathbf{X}$ to a given hyperplane, it is possible to define an indicator function that uses such hyperplane as threshold. This

decision rule is given by:

$$y = \begin{cases} +1, & \text{if } \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p > 0 \\ -1, & \text{if } \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p < 0 \end{cases}. \quad (2.11)$$

where the points on the positive side of the hyperplane are associated with a positive label ($y=+1$), and the opposite is also valid.

Then, the Hyperplane Classifier uses equation 2.11 as an estimation process, where the points are unseen samples, and their relative position to the hyperplane defines from which class they belong.

However, if a set of samples are linearly separable (perfectly separated using a hyperplane), an infinite number of such hyperplanes can exist. To select the best hyperplanes among those a method called Maximal Margin Classifier was proposed.

The main intuition behind the Maximal Margin Classifier is that the maximal margin hyperplane, which is farthest one from all training observations, is the optimal decision. Where ‘farthest’ stands for the highest sum up of the perpendicular distance from all training samples to a given hyperplane. And, the smallest of these perpendicular distances is known as the margin.

Therefore, find this best separating hyperplane can be formulated as a maximization problem:

$$\underset{\beta_1, \dots, \beta_p}{\text{maximize}} \ M, \quad (2.12)$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}) \geq M, \forall i = 1, \dots, n. \quad (2.13)$$

Note that, M is the margin length, n is the number of training samples, $X_1, \dots, X_n \in R_p$ and class labels $y_1, \dots, y_n \in \{-1, +1\}$.

The resulting hyperplane of such optimization is known as the Maximal Margin hyperplane. To build the Maximal Margin Classifier, this optimal hyperplane has to be applied to the equation 2.11.

However, one of the main Maximal Margin Classifier disadvantages is its sensibility to individual observations.

To overcome that, the Soft Margin Classifier, also known as support vector classifier, adds a slack variable ϵ for each observation in the training set. The goal of such variable is to allow some samples to be incorrect classified if this increase the classifier’s margin.

Therefore, the optimization process to find the best Soft Margin Classifier to separate most of the training can be modeled as the following optimization problem:

$$\underset{\beta_1, \dots, \beta_p, \epsilon_1, \dots, \epsilon_n}{\text{maximize}} \quad M, \quad (2.14)$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}) \geq M(1 - \epsilon_i), \quad (2.15)$$

$$\epsilon_i \geq 0, \sum_{i=1}^n \epsilon_i \leq C, \quad (2.16)$$

where C is hyperparameter greater than zero.

One of the main limitations of the Support Vector Classifier is that its decision boundary is always linear. To extend its concepts to nonlinear separable problems the SVM is proposed. The main idea of the SVM classifier is to transform the input feature space in a specific way that a linear decision boundary (in this new space) can correctly separate the samples. As describe exactly how the support vector classifier is computed is beyond the scope of this dissertation, it turns out that SVM's decision rule can be mathematically described as:

$$f(X) = \beta_0 + \sum_{i=1}^n \alpha_i K(X^T, X'), \quad (2.17)$$

where α is an SVM coefficient learned during training and K is the target space transformation. For example, a linear kernel (transformation) can be expressed by:

$$K(X, X') = \sum_{j=1}^p X_j^T X'_j, \quad (2.18)$$

and the radial based transformation is defined as:

$$K(X, X') = \exp(-\gamma \sum_{j=1}^p (X_j^T - X'_j)^2). \quad (2.19)$$

2.4 Random Forest

Random forest, which was first proposed by Ho (1995), is an ensemble method that combines the estimations performed by a set of Decision Trees (DTs) (BREIMAN, 2001).

In general, DTs are a non-parametric supervised type of machine learning algorithm used for classification and regression tasks. In the classification context, their goal is estimate the label of a given instance using decision rules (thresholds) learned from the training data.

However, as finding the optimal decision tree is known to be an NP-complete problem (BREIMAN, 2001), practical TDs training algorithm produces a suboptimal solution and can be conceptually associated with a greedy search, where local decisions are made at each tree node. Another TDs drawback is their susceptibility to learn even the noise in the training set (overfit).

To reduce the variance of DTs predictions, Random Forest uses an ensemble approach called random Bagging. In this context, random Bagging is built and train several estimators (e.g., DTs) on different random subsets of the training set and then combine their outputs (e.g., compute the majority vote) to compose the final prediction.

More formally, It turns out that, each DT can be viewed as a weighted neighborhood (LIN; JEON, 2006), and is expressed as:

$$\hat{y} = \sum_{i=1}^n W(X_i, X'_i) y_i, \quad (2.20)$$

where X' is a unseen observation, $\hat{y}$ its estimated label, X_i is i 'th training sample, y_i its respective label and W a weight function.

This weight function changes according to the classification method used, for example, in Decision Trees context:

$$W(X_i, X'_i) = \begin{cases} \frac{1}{k} & , \text{ if } X_i \in K \\ 0 & , \text{ otherwise } \end{cases}, \quad (2.21)$$

where K are k points in the same leaf as X' . Note that, the only difference between this weight function and the one computed for k-NN is that K is a set composed of the k points closest to the unseen observation. Moreover, it is possible to extend the equation 2.21 to compose the Random Forest prediction one, which is given by:

$$\hat{y} = \frac{1}{m} \sum_{j=1}^m \sum_{i=1}^n W_j(X_i, X'_i) y_i, \quad (2.22)$$

where m is the number of estimators (trees) used and W_j their respective weight functions.

2.5 Multilayer Perceptron (MLP)

Multilayer Perceptron (MLP) (HAYKIN, 2008) is a non-parametric supervised type of machine learning for classification. Moreover, this classifier is also an extension of the Perceptron classifier for nonlinear separable datasets.

Introduced by ROSEMBLATT (1958), Perceptron algorithm computes a weighted sum of the inputs and applies the result to an activation function. Abstractly, each weight

indicates the relevance of a given feature to predict a given class. More formally, Perceptron prediction can be expressed as the following equation:

$$\hat{y} = \varphi \left(\sum_{i=1}^p x_i w_i + b \right), \quad (2.23)$$

where φ is an activation function (e.g., logistic/softmax), x is an input vector, p is the number of features of a given input, w is the vector of weights, which combined with b , are Perceptron learnable parameters. Further, the training process of such classifier is an iterative process, given by:

$$w_i(t+1) = w_i(t) + \eta \cdot (y - \hat{y}(t)) \cdot x_i \quad \forall i = 1, \dots, p, \quad (2.24)$$

in which t is a given training time step, y is the true label, $\hat{y}$ is the predicted label, and η is the learning rate. However, despite been a simple and powerful algorithm, Perceptron is not able to perform nonlinear classification.

To overcome that, MLP was proposed. MLP is a computational graph composed of a stack of perceptron layers. Each of these layers (except the last one), has their logistic activation functions (used to build a linear classifier) replaced by another nonlinear function, such as sigmoid function (used to perform a nonlinear transformation on the feature space).

More formally, the decision rule of this classifier can be mathematically formulated as:

$$\hat{y}_{l_k} = \begin{cases} \varphi_{0_k}(\sum_{i=1}^p x_i \cdot w_{0_{k_i}} + b_{0_k}) & \text{if } l = 0 \\ \varphi_{l_k}(\sum_{i=1}^D \hat{y}_{l-1_i} \cdot w_{l_{k_i}} + b_{k_l}) & \text{otherwise} \end{cases}, \quad (2.25)$$

where x is the input vector of p dimensions, l is a given layer index, D is the number of neurons in the layer $l - 1$, and φ is the nonlinear function. It is worth noting that, in general, MLP learnable parameters (weights and bias) are initialized with random values close to zero.

Further, to improve its performance, MLP generally uses an iterative process mathematically, expressed by the equation:

$$\hat{w}_{l_{k_i}}(t+1) = \begin{cases} w_{0_{k_i}}(t) + \eta \cdot \delta_{1_i}(t) \cdot x_k(t) & \text{if } l = 0 \\ w_{l_{k_i}}(t) + \eta \cdot \delta_{l+1_i}(t) \cdot \hat{y}_{l_k}(t) & \text{otherwise} \end{cases}, \quad (2.26)$$

where η is the learning rate and $\delta_{l_k}(t)$ is the local gradient of the k 'th neuron of the l 'th

layer. This local gradient is given by:

$$\delta_{l_k}(t) = \begin{cases} \varphi'_{l_k}[u_{l_k}(t)] \cdot \frac{1}{2M} \sum_{o=1}^M [y_{l_o}(t) - \hat{y}_{l_o}(t)]^2 & \text{if } l = z \\ \varphi'_{l_k}[u_{l_k}(t)] \cdot \sum_{j=1}^J w_{l_{k_j}}(t) \cdot \delta_{l+1_j}(t) & \text{otherwise} \end{cases}, \quad (2.27)$$

where z is the last layer, M is the number of neurons in the last layer(z) and J is the total number of neurons in the $l + 1$ 'th layer.

Note that, the derivative $\varphi'_i[u_{l_i}(t)]$ has different values depending on the activation function chosen, for example:

$$\varphi'_{l_k}[u_{l_k}(t)] = \begin{cases} y_{l_k}(t)[1 - y_{l_k}(t)], & \text{if } \textit{logistic/sigmoid} \\ \frac{1}{2}[1 - (y_{l_k}(t))^2], & \text{if } \textit{hyperbolic tangent} \end{cases}. \quad (2.28)$$

3 Convolutional Neural Network

In this chapter, a brief overview about the essential building blocks of modern CNN architectures is presented.

3.1 Lenet-5

Inspired by Artificial Neural Networks (HAYKIN, 2008), Neocognitron (FUKUSHIMA; MIYAKE, 1982) architectures, back-propagation (BENGIO; GOODFELLOW; COURVILLE, 2015) optimization strategy and shared-weights concept (also named local connectivity (RUMELHART; HINTON; WILLIAMS, 1986); and motivated by the need to solve real world image-recognition problems without a computationally expensive preprocessing stage or even requiring complex and time consuming architecture engineering, LeCun et al. (1998) developed a pioneering 7-level convolutional network to classify hand-written numbers on checks images (LECUN et al., 1990). This architecture, presented in Figure 1, was named as LeNet-5 and among its main contributions to deep learning research field are: the introduction of convolutional and subsampling layers.

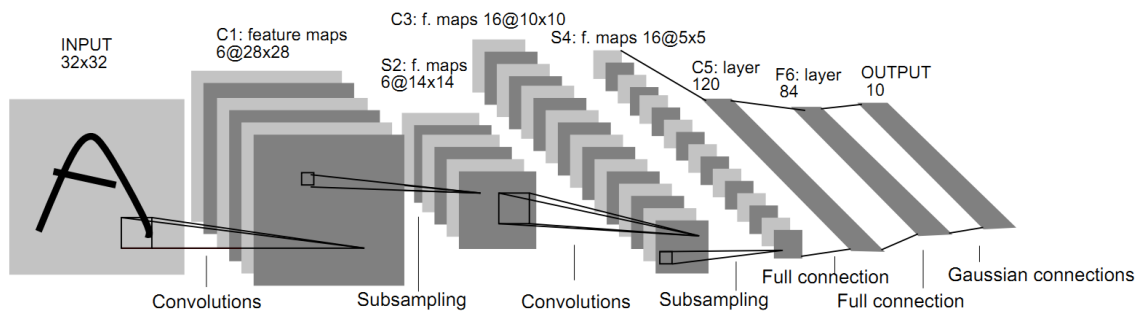


Figure 1 – Architecture of LeNet-5 Convolutional Neural Network (LECUN et al., 1998).

3.1.1 Convolutional Layers

Despite been considered a universal function approximator (HAYKIN, 2008), fully connected layers, with enough discriminative power to solve (even simple) computer vision tasks, usually, have an infeasible amount of free parameters to allow a proper generalization of the knowledge.

To address this issue, convolutional layers use a restricted connection-scheme between inputs and weights. More specifically, differing from the fully connected layer strategy, in which, each neuron has a weighted connection with each input (e.g., each pixel),

in a convolutional layer, each neuron only has weighted connections with a subregion of the input (local receptive field) (LECUN et al., 1998).

Moreover, instead of performing an inner product between all inputs and weights (which is possible in fully connected layers) in convolutional layers, as the name convolutional suggests, each neuron perform an operation that is equivalent to a convolution (the kernel to be applied over the image is composed of the neuron weights). Furthermore, as in the fully connected layers, the output of the convolution is used to a squashing function (e. g. logistic function, hyperbolic tangent) and named as activation map or even feature map (LECUN et al., 1998).

3.1.2 Pooling Layers

As the classical convolutional layer does not reduce the input size, as the fully connected one, an extra step may be required to downsample the convolution output, preferably without adding any new parameters to the model. Moreover, another motivation to use this downsampling stage is based on the assumption that features relative position is more relevant than their explicit location.

To overcome this limitation, the Pooling Layer is introduced. A pooling layer performs a summary of adjacent inputs based on an aggregation rule, e.g, the max pooling operation outputs (BENGIO; GOODFELLOW; COURVILLE, 2015) the maximum input value within a rectangular neighborhood. A graphic example of a max pooling layer operation is presented in Figure 2.

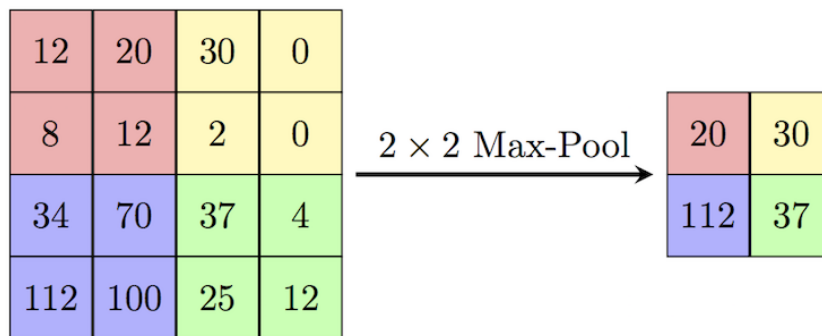


Figure 2 – Illustration of a 2D Max pooling operation (KARPATHY; LI, 2017)

3.2 AlexNet

Despite of been developed in the late 80's CNNs only become popular in 2012 when Krizhevsky, Sutskever e Hinton (2012) propose a CNN model named AlexNet. This deep model won, by a large margin, the Large-Scale Visual Recognition Challenge (ILSVRC) (RUSSAKOVSKY et al., 2015) object detection task, an important benchmark to computer

vision techniques (DENG et al., 2009; KRIZHEVSKY; SUTSKEVER; HINTON, 2012). It worth noting that, the attention given to this ILSVRC competition is mostly due to its dataset size and complexity. ImageNet, the name of this dataset, has 1000 categories and more than 1.2 million images (DENG et al., 2009).

Thus, since AlexNet breakthrough all the subsequent ILSVRC winner used CNN-based models. However, despite such enthusiasm, AlexNet model has a similar architecture to its predecessor, LeNet-5 (LECUN et al., 1998) (basically composed of convolutional, pooling and fully connected layers).

Further, among the most notable contributions of AlexNet paper to the research field of CNNs are: Rectified Linear Unit (ReLU), and DropOut.

3.2.1 ReLU Nonlinearity

Based on previous studies (NAIR; HINTON, 2010; GLOROT; BORDES; BENGIO, 2011), ReLU function, which can be mathematically defined as $g(z) = \max(0, z)$, was chosen as AlexNet neurons nonlinearity. A graphic representation of the ReLU function is presented in Figure 3. This choice was mostly motivated by Nair and Hinton (NAIR; HINTON, 2010) experiments, which indicate that the use of ReLUs significantly speed up the training convergence process, when compared with other nonlinearities functions, such as: Logistic or Hyperbolic tangent.

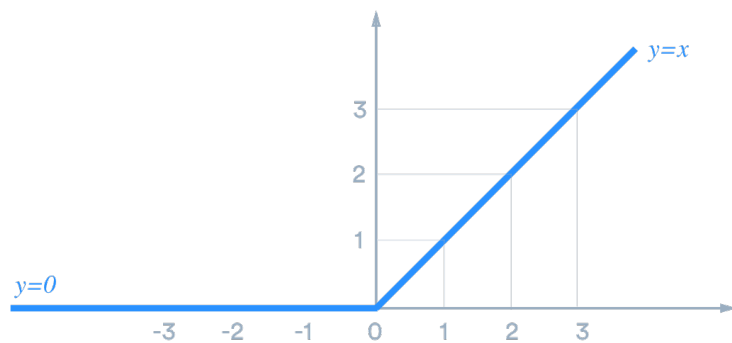


Figure 3 – The Rectified Linear Unit (ReLU) activation function (<https://goo.gl/FLHUrZ>)

3.2.2 Dropout

A common and successful strategy to deal with overfitting is combine the predictions of different models (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). However, this seems unfeasible for large neural networks, since the training process of each model can take up to several days.

To overcome this issue, an efficient regularization strategy called “dropout” (HINTON et al., 2012) provides an approximation of ensembling an exponentially number of models. Dropout, illustrated in Figure 4, can be roughly described as a technique where,

for a given network input, the output each neuron has a probability of been set to zero, usually 50%.

According to this idea, these “dropped out” (set to zero) neurons do not contribute to the network forward pass and, consequently, do not have their weights updated during the gradient back-propagation.

As a result of dropout, every input presented is processed by a different architecture. Moreover, Hinton et al. (2012) argues that this uncertainty, reduces co-adaptations of neurons and forces the network to learn a more robust set of features, acting as a powerful regularization strategy for CNNs.

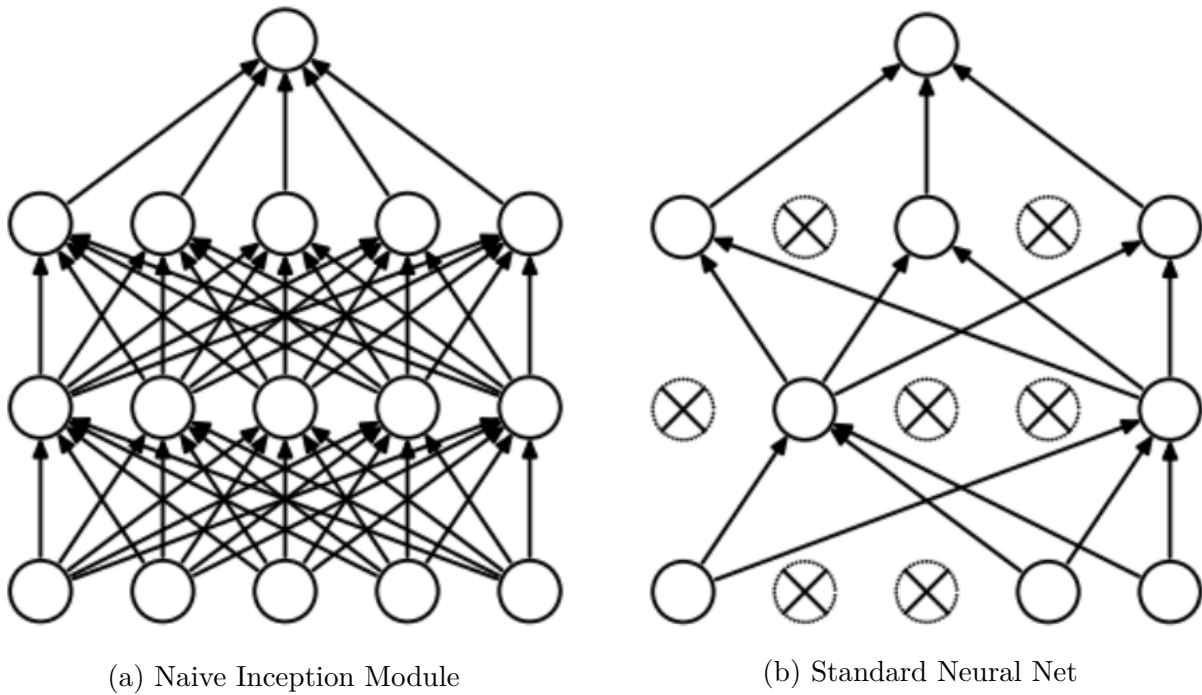


Figure 4 – The illustration in item (a) represents the conventional information flow during a forward pass of a network and in the item (b) this flow is interrupted (set to zero) at some neuron, this process is called dropout (SRIVASTAVA et al., 2014).

3.3 VGG

Experiments performed in Oxford Visual Geometry Group (VGG) paper (SIMONYAN; ZISSERMAN, 2014), including the winner model of ILSVRC 2014 (RUSAKOVSKY et al., 2015) detection and localization tasks, demonstrated that, for all architectures evaluated (11, 13, 16, or 19 layers), there is a positive correlation between the models depth (number of layers) increase and its accuracy on ImageNet Dataset. Such deep increase, without overfitting the model, was only possible through a regularization strategy named Factorized Convolutions.

3.3.1 Factorized Convolutions

Inspired by the Zeiler e Fergus (2014) paper, which used smaller receptive window (kernel/filter) size than AlexNet (e.g., in the first convolution layer (ZEILER; FERGUS, 2014) network used 7×7 kernels whereas AlexNet used 11×11 kernels), VGG network used very small (3×3) kernels in all its convolution layers.

There are three important aspects to answer: why small kernels enabled VGG network to overpower most of the previous models?

First, in a convolution, a given point in the activation map is only influenced by a connected input subregion, e.g., a 3×3 convolution is influenced by a 3×3 window of the input. Moreover, if a second 3×3 convolution is performed, but now over first activation map, each point of the new activation map is influenced by a 3×3 window of the first one. However, as each point of the first activation map is combining a 3×3 information from the input, a 3×3 combination of these first activation points is influenced by a 5×5 information from the input. Thus, if this process is repeated once more, each output point of three-layer 3×3 convolution stack is influenced by a 7×7 subregion of the input. In other words, a three-layer 3×3 convolution stack has the same effective receptive field of a single 7×7 convolution. An example of this equivalency process is illustrated in Figure 5.

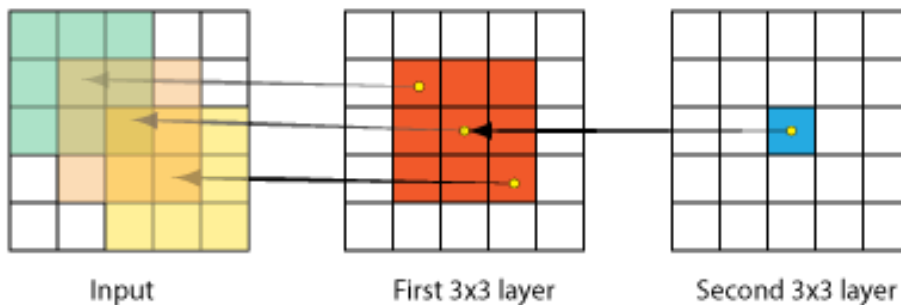


Figure 5 – The effective receptive field of a stack of 2 3×3 convolutions (SIMONYAN; ZISSERMAN, 2014)

Second, a three-layer 3×3 convolution stack has fewer parameters than a single 7×7 convolutional layer. Third, as each layer has its nonlinearity, a stack of conv. layers have more discriminative power than just a single layer.

Therefore, the VGG Networks has become very popular not only because of their good results in ILSVRC 2014 (RUSSAKOVSKY et al., 2015), but also for their uniform architecture. However, accounting for 140 million parameters, VGG can be a bit challenging for embedded systems.

3.4 Inception

Since LeNet5, the building blocks of deep neural architectures has been the same (convolutional layers, pooling layers, and fully connected layers), however this change when Google Brain team released the GoogleNet (SZEGEDY et al., 2015). Inception, how is also called GoogleNet, set state-of-the-art results on the ILSVRC 2014 (RUSSAKOVSKY et al., 2015) detection task, achieving a top-5 error rate of 6.67%.

Further, among the Inception models contributions to the deep learning research field are Inception Module, Batch-Normalization, and Asymmetric Convolutions.

3.4.1 Inception Module

Design a CNN architecture usually consists of choosing which operations use (1×1 conv., 3×3 conv., 5×5 conv. or pooling) at each layer. So, why not perform all operation at once, and then concatenate their outputs, taking advantage of multi-level feature extraction? An illustration of this parallel architecture is shown in Figure (a).

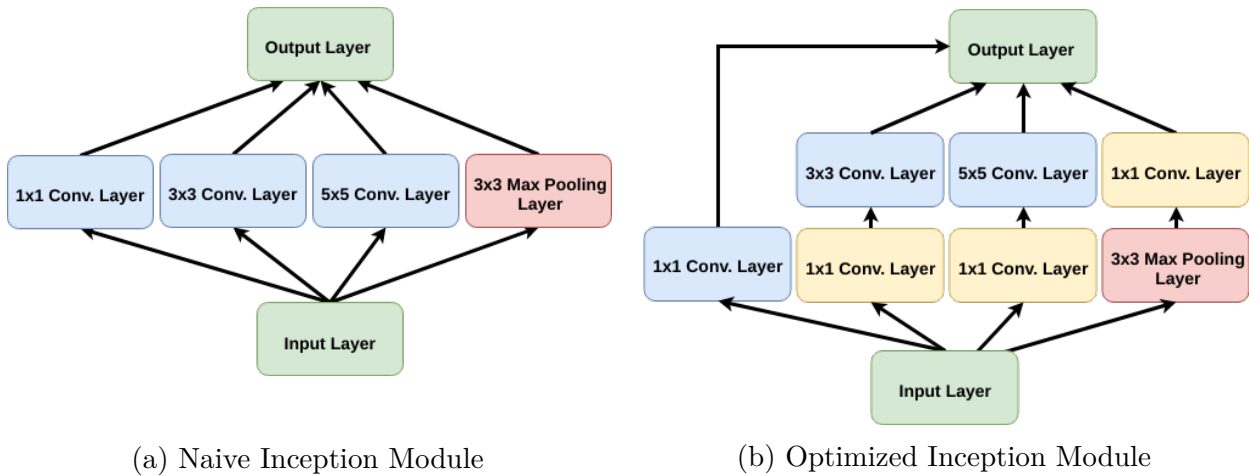


Figure 6 – Two Inception architecture illustrations. In item (a) the naive Inception Module; In item (b) the Inception Module with dimensionality reduction.

The main disadvantage of this naive Inception Module (parallel approach previously described) is that after a few layers like this the amount of parameter added to the network would probably make it overfit during training.

However, to address this parameter inefficiency, Google Brain performed 1×1 convolution to compressed the inputs of the convolution layers and the output of the pooling layers. An illustration of the Inception Module is presented in Figure 6 (b). This simple adaptation preserved most of the parallel approach accuracy and significantly reduced the amount of free parameter.

3.4.2 Batch-normalization

One of the most challenging issues in deep networks optimization is the Internal Covariate Shift (Ioffe; Szegedy, 2015), that is, roughly, a change in intermediate layers inputs distribution at training time. This happens because the inputs of a given layer are affected by all previous layers weights, which usually change over the optimization. Moreover, as the layers weights need to adapt to a new input distribution at each weight update a negative effect appears, the model's convergence became slower.

To address this Internal Covariate Shift issue, Ioffe e Szegedy (2015) proposed Batch-normalization (BN), a new mechanism that incorporates normalizing activations in the architecture itself.

With only two extra parameters per activation, BN normalizes the outputs of the inner product between weights (plus bias) and inputs of a given layer, based on their mean and standard-deviation. The BN computations are mathematically described in Figure 7. This is equivalent to “whitening” the data, making a zero-centered distribution and normalizing its range throughout the batch. Moreover, among the benefits of BN are: the layer after a BN focuses on how to best associate inputs; BN has a powerful regularization effect and reduces the need for dropout.

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;	
Parameters to be learned: γ, β	
Output: $\{y_i = \text{BN}_{\gamma, \beta}(x_i)\}$	
$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i$	// mini-batch mean
$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2$	// mini-batch variance
$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$	// normalize
$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i)$	// scale and shift

Figure 7 – Batch normalizing transform, applied to activation x over a mini-batch (Ioffe; Szegedy, 2015).

3.4.3 Asymmetric Convolutions

Extending the concept of Factorized Convolutions, previously applied in VGG paper (SIMONYAN; ZISSERMAN, 2014), and also base on the Flattened CNNs (JIN; DUNDAR; CULURCIELLO, 2014), Google Brain team use Asymmetric Convolutions to propose a new version of the Inception Modules. Aiming to improve the Inception Module parameter efficiency, they realize that, rather than perform a $n \times n$ convolution it would

be better to apply a $n \times 1$ convolution followed by a $1 \times n$ convolution, thus, factoring it into two asymmetric convolutions. A generic architecture of an Inception Module with asymmetric convolutions is presented in Figure 8.

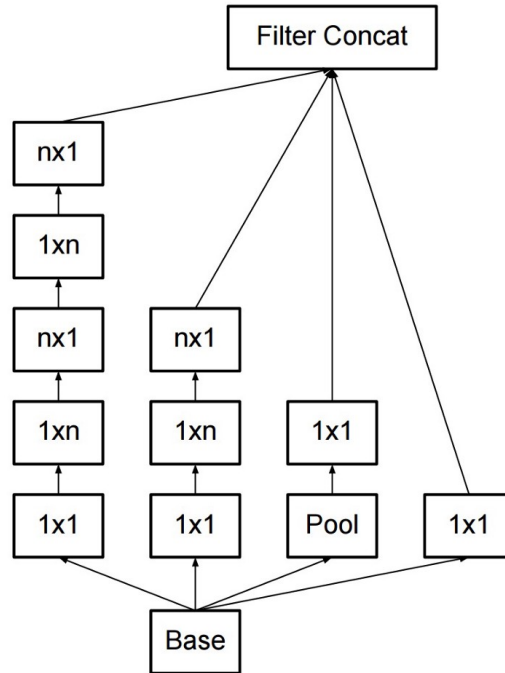


Figure 8 – An architecture illustration of an Inception Module with asymmetric convolutions (SZEGEDY et al., 2016).

3.5 ResNet

Despite VGG indications, pointing to a positive correlation between network depth and accuracy, further experiments, performed in Residual Neural Network (ResNet) paper (WU; ZHONG; LIU, 2017), do not support this view, at least for classical CNNs (networks only composed of convolutions, pooling, and fully-connected layers).

ResNet authors argue that very deep classical CNNs are harder to optimize, mostly due to the vanishing-gradient problem (BENGIO; SIMARD; FRASCONI, 1994). In practice, an 18-layer network has higher testing and training errors than a 34-layer network, even if these 16 additional layers only performed an identity mapping.

To overcome this issue and inspired by the Recurrent Neural Networks “skip connections” concept, ResNet authors introduced a novel deep neural network building block, the Residual Block.

3.5.1 Residual Block

In a Residual Block, the input flows for two branches, one connects the input to the merging stage, the other performs a stack of convolutions layers (usually 2 ou 3) preserving the input dimensions (e.g., if the input has C channels, so C filter are applied), then sends the results to the merging stage. Finally, both branches outputs are depthwise added up. Note that, the second branch last convolution layer do not have nonlinearities after the convolutions operations. These missing activation functions are applied to the merge stage output, composing the Residual Block output. An illustration of a generic version of this Residual Block is presented in Figure 9.

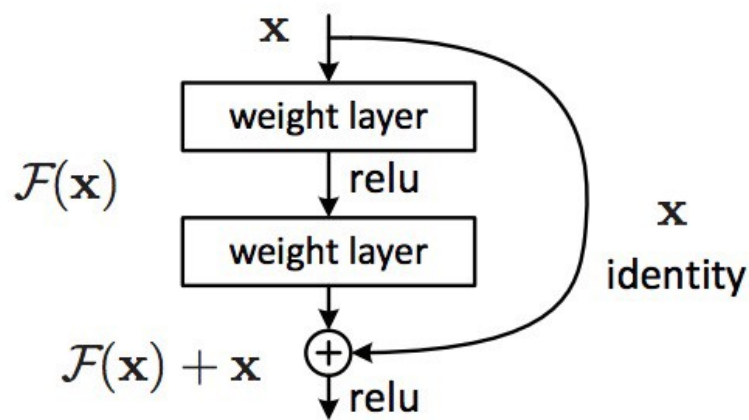


Figure 9 – An example of a Residual block architecture (WU; ZHONG; LIU, 2017).

With this simple strategy ResNet set state-of-art results in object detection, localization, and segmentation tasks on ILSVRC (RUSSAKOVSKY et al., 2015) and COCO 2015 (LIN et al., 2014), achieving a top-5 classification error rate of 3.57% (ILSVRC) and surpassing even the human-level performance on ImageNet dataset.

3.6 Inception-ResNet

In 2016, less than a year latter than ResNet, the Google Brain team performed architectures design experiments by combining the ideas of Residual Connections (WU; ZHONG; LIU, 2017) with the Inception Module (SZEGEDY et al., 2015). Among these experiments is Inception-ResNet-V2 model (LÄNGKVIST; KARLSSON; LOUTFI, 2014), which achieved a top-1 classification error of 16.5% and a top-5 of 3.1%, thus surpassing, ResNet and Inception results.

3.6.1 Inception Module with Residual connections

The only difference between a straightforward combination between an Inception Module with Residual Connections is a scaling layer at the Inception module output.

Apparently, without this scaling stage, the network starts, after a few thousands of iterations, to outputs only zeros. The authors performed several experiments to deal with that, but only scaling down, by a factor between 0.1 and 0.3, the inception output before adding them to inputs stabilized the optimization procedure. The generic illustration of the inception module with Residual Connections is presented in Figure 10.

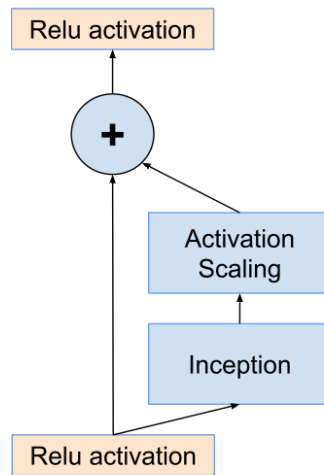


Figure 10 – An illustration of a general architecture layout when combining Residual Connections with Inception modules (LÄNGKVIST; KARLSSON; LOUTFI, 2014).

3.7 Xception

In 2016, François Chollet released the Extreme Inception (Xception) paper (CHOLLET, 2017), investigating the hypothesis that cross-channel correlations can be entirely decoupled from spatial correlations. To accomplish this, the author proposes the Depthwise Separable Convolution layer. It is worth noting that, with 36 Depthwise Separable Convolution layers structured into 14 modules (most of them with residual connections), Xception architecture achieved good results on JFT dataset (Hinton; Vinyals; Dean, 2015), a 350-million-images database with 17,000 possible labels, surpassing the performance of ResNet and Imagenet architectures.

3.7.1 Depthwise Separable Convolutions

The idea behind a 2D convolution layer is learned features directly from a 3D space (width, height, and channels). Thus, a convolution maps cross-channel and spatial correlations simultaneously.

Further, in an Inception Module, a convolution is factorized into a set of 1x1 convolutions, that are influenced specifically by cross-channel correlations, followed by

regular convolutions (e.g., 3x3 or 5x5 convolutions), which maps both spatial correlations and cross-channel.

However, in depth separable convolution, first a channel-wise convolution is performed, which maps only spatial correlations; then a set of pointwise convolutions (a 10 of 1x1 convolution) is computed, mapping the cross-channel correlations; finally, all convolutions outputs are depthwise concatenated. Hence, the spatial correlation is entirely decoupled from cross-channel correlations.

Among the benefits of the Depthwise Separable Convolutions strategy over the classic Inception Module are a reduction in the number of free parameters and more efficient execution. Both of these characteristics are valuable for mobile or even embedded applications.

3.8 MobileNet

In 2017, Google researchers designed and released a “mobile-first” architecture called MobileNet (HOWARD et al., 2017). This 28-layer network is essentially (except the first layer) build on depthwise separable convolutions, which were introduced by (HOWARD et al., 2017) in 2016, and achieved, in its baseline configuration, 99.1% of the Inception V3 (SZEGEDY et al., 2015) detection accuracy (on imageNet database) with 87.7% of fewer parameters and 88.6% fewer operations.

The MobileNet approach distinguishes from the Xception model for the introduction of two hyper-parameters (Width Multiplier, Resolution Multiplier), which allow the model to match, efficiently, mobile and embedded systems hardware constraints.

3.8.1 Width Multipliers

In order build a flexible and efficient architectures, MobileNets authors propose a hyper-parameter α , also called ‘width multiplier’. In MobileNets paper, this α parameter is a reduction factor in the number of channels-wise convolutions, in a depthwise separable convolution layers. Moreover, α almost quadratically affects the model’s number of parameters and computational cost. Note that $\alpha \in (0, 1]$, where 1 is the MobileNet’s baseline configuration.

3.8.2 Resolution Multipliers

The second hyper-parameter proposed is the resolution multiplier ρ . In practice, this ρ parameter adapts the network architecture to process different input resolution (width and height). For example, if ρ equal 1, the network input will be a 224×224 image,

if ρ equals 0.75, the input will be a 168×168 image, and so on. Note that, as $\alpha, \rho \in (0, 1]$ (been $\rho=1$ the baseline) and ρ also has a quadratically effect over the computational cost.

Thus, width multiplier and resolution hyper-parameters offer useful a trade-off between size, accuracy, and latency, when targeting a system with hardware limitations.

3.9 DenseNet

Dense Convolutional Network (DenseNet), proposed by Huang et al. (2017), extended the shorter connections concept, previously explored by (SZEGEDY et al., 2015), composing what is called Dense Block.

3.9.1 Dense Blocks

There are two main differences between DenseNet and ResNet architectures. First, rather than perform shorter connections between adjacent layers, DenseNet connects each layer to every other layer with the same input resolution (width and height). This set of interconnected layers are named Dense Block. An illustration of a Dense Block is presented in Figure 11. Second, every DenseNet layer performs a depthwise concatenation of the previous outputs instead of summing them.

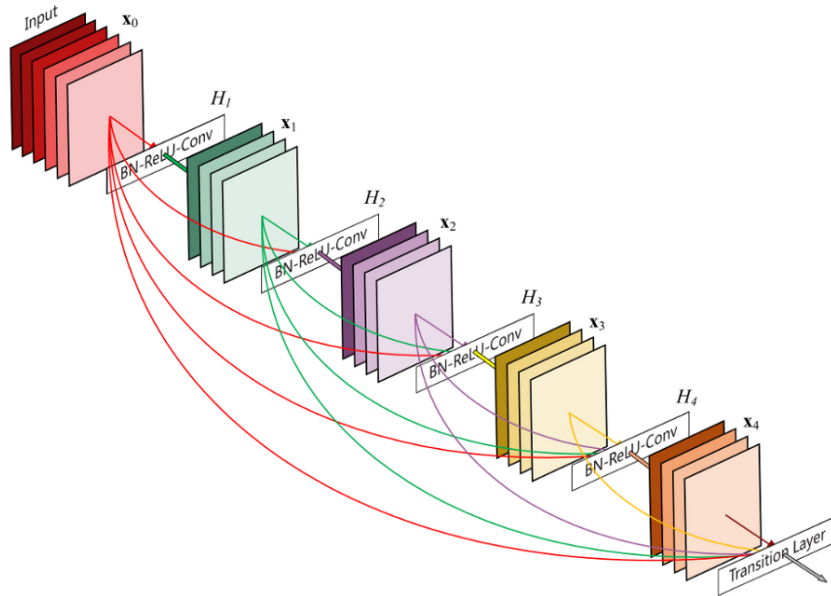


Figure 11 – Illustration of a general architecture layout when combining Residual Connections with Inception modules (HUANG et al., 2017).

Thus, as a result of this strategy, DenseNet slightly outperform ResNet accuracy results on ImageNet detection task, requiring approximately 55% fewer parameters and 50% fewer operations than ResNet.

3.10 NASNet

Based on the assumption that design accurate and efficient image classification architectures often are complex architecture engineering process Neural Architecture Search (NAS) (ZOPH; LE, 2016) paper explores the hypothesis that is possible to efficient learn a model architecture directly from a dataset of interest. Inspired by NAS experiments, Google Brain team proposed a novel paradigm of designing convolutional architectures, NASNet search space.

3.10.1 NASNet search space

Such paradigm constrain the wide search space, of combining several building blocks (e.g. identity, 1×3 then 3×1 conv., 1×7 then 7×1 conv., 5×5 max pooling, 1×1 conv., 3×3 conv., 3×3 depthwise-separable conv., etc.), to form a whole network, into a smaller one, searching for only two module architecture (Normal Cell and Reduction Cell). An illustration of the accurate (on ImageNet) modules found into the NASNet search space is presented in Figure 12. These modules are alternated and repeated several times to compose the final architecture.

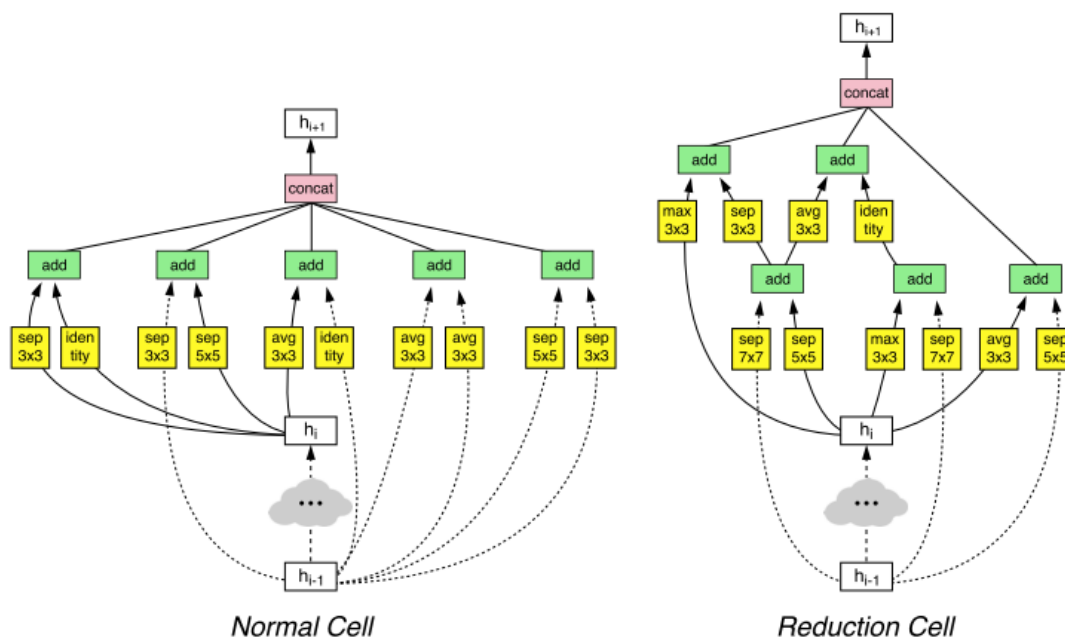


Figure 12 – Illustration of a general architecture layout when combining Residual Connections with Inception modules (ZOPH; LE, 2016).

Moreover, to speed up this search process, NASNet rely on the assumption that is possible to transfer a module learned on a small dataset, e.g., CIFAR-10, to a larger one, such ImageNet.

Thus, NASNet achieved state-of-the-art detection accuracy results on ImageNet dataset, specifically, 82.7% on top-1 and 96.2% on top-5. Further, in its lite version, NASNet matches the parameter efficiency of MobileNet, but is more accurate and faster. However, it is worth noting that, despite using 500 Nvidia K40 GPUs, the fastest architecture search took 4 days.

4 Methodology

This chapter describes the experiments developed to evaluate the performance of CNNs as feature extractors in the lung nodule malignancy classification task using chest CT images from the LIDC/IDRI database.

The remainder of this chapter is organized into 3 sections: database processing, explored approach implementation and its evaluation. Each of these is represented by a sequence of steps in the flowchart in Figure 13, wherein yellow are the steps of the first section, in blue those of the second and in green those of the third.

In the first section, the LIDC/IDRI database is filtered, extracted, and formatted. In the second, 11 CNN architectures are built, trained and transformed into feature extractors. In the third, each of the implemented extractors is applied to the LIDC/IDRI database, thus generating 11 sets of features. Then each of these sets is classified by 6 machine learning techniques. Finally, 5 evaluation metrics are computed for each classifier applied.

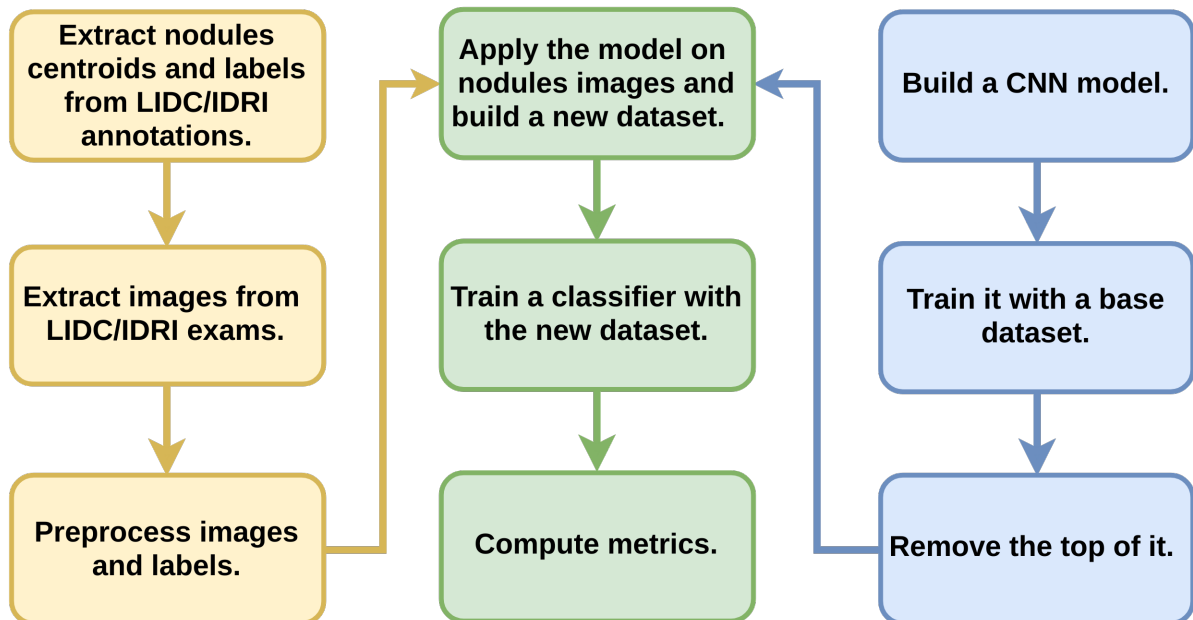


Figure 13 – In yellow the sequence of steps performed to process the LIDC/IDRI dataset. The blue branch outlines the procedures to build a deep feature extractor. The green branch is how the proposed approach is evaluated.

4.1 Processing the LIDC/IDRI database

The database processing (used in this work to represent the lung nodule malignancy classification task) can be divided into 2 stages: the processing of the exams and the reduction of the problem.

The LIDC/IDRI database, used in this work to represent the lung nodule malignancy classification task, is described in detail at Appendix A. The process of transforming this database into a binary problem can be divided into 2 stages: the medical annotations processing and the problem reduction.

4.1.1 Medical Annotations Processing

To perform the first stage, 5 steps are required. In the first, each of the 1018 LIDC/IDRI database exams, which is initially an unordered sequence of DICOM files, has its images sorted (as indicated in the file header) and converted into a single three-dimensional array. The elements of this matrix indicate the radiological density of the chest CT exam in question.

Also, there is an XML file for each CT exam, containing the medical annotations of 4 experienced radiologists. Among other information, each one of these physicians also marked the external contour coordinates and malignancy level of each nodule detected. Based on this information it is also possible to compute the centroid of the nodule in question and its largest radius (the largest Euclidean distance between the centroid and a contour point). Figure 14 shows the first step of the medical annotations processing.

However, it is common and even expected, to have nodules detected, segmented, and categorized by more than one radiologist. In addition, allowing two or more samples with a high degree of similarity to be at the same time in the training and test set reduces the database representativeness and, consequently, the reliability of the classification process.

In the second step, a strategy for merging these redundant annotations is proposed. First, the Euclidean distances between the centroids of all the annotations of a given exam are computed. Then, the centroids that have a distance smaller than 80% of their respective radius have their annotations combined. This process of merging the annotations consists of computing the midpoint between the centroids and the median of their malignancy levels. This process of handling redundant annotations is illustrated in Figure 15.

In the third step, after ensuring that there is only one annotation per nodule, the centroids are located in the three-dimensional matrix generated in the first step of this section. Then, the 16 voxels surrounding the centroids are copied and stored in a list. This process of extracting the volumes of interest is illustrated in Figure 16.

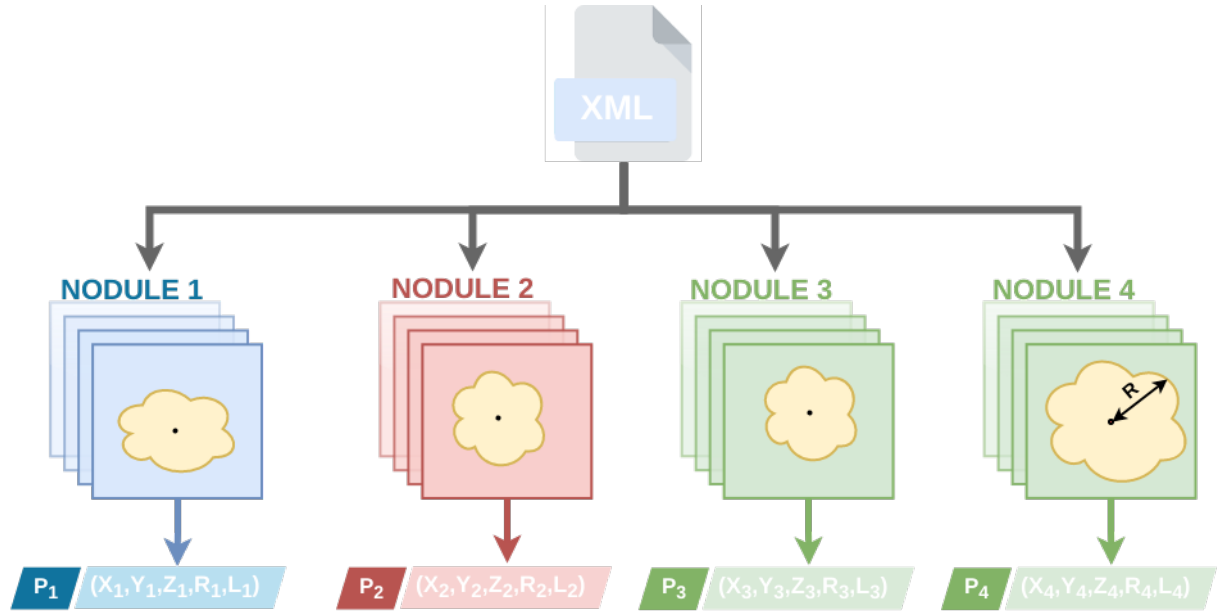


Figure 14 – Illustration of the first step of medical annotation processing, where X , Y , Z are the centroid coordinates; R the radius; and L the malignancy level of a given nodule.

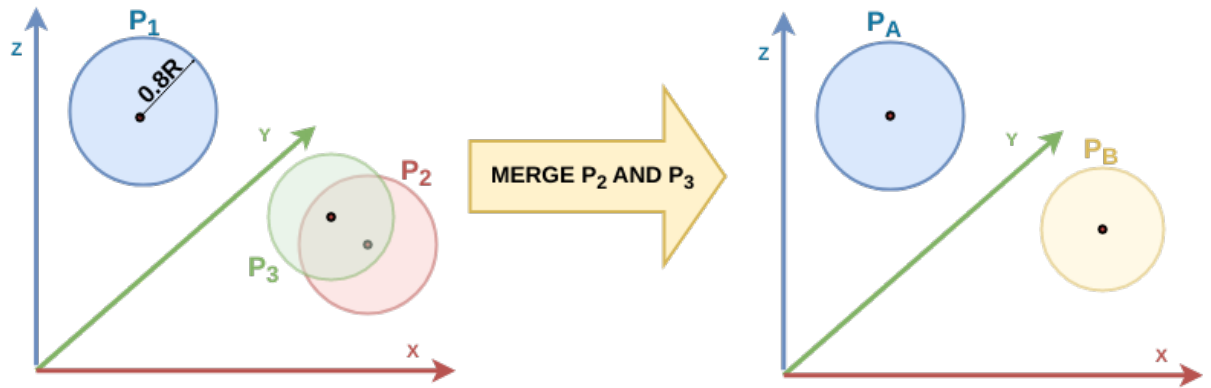


Figure 15 – Illustration of the medical annotations processing second step, the treatment of redundant annotations.

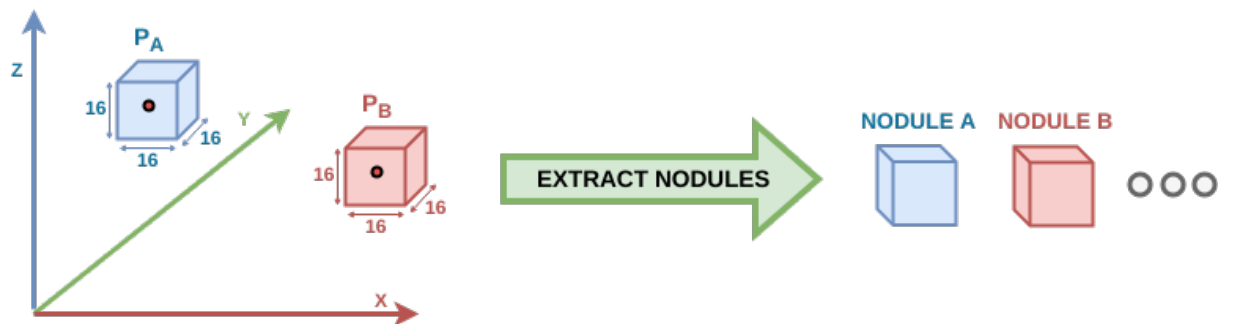


Figure 16 – Illustration of the volumes of interest extraction process.

In the fourth step, each one of these volumes of interest is sub-sampled and transformed into a 3-channel image. An illustration of this transformation is shown in

Figure 17.

This process is necessary due to the restrictions imposed by the hypothesis explored in this work. More specifically, to apply the concept of Transfer Learning, one must first understand how the "knowledge" of an Artificial Intelligence is represented. In the context of CNNs, this "knowledge" lies in the topology of architecture and its weights.

However, if Transfer Learning is the concept explored in this work, then the architecture used (which in this case represents its knowledge) must be preserved and then transferred. In practice, this implies that the network input can not change dimensions.

Thus, if the CNNs used in this work are trained with 3 channels images, the database preprocessing must restrict the nodules representation to an image with only 3 channels. That is, the 3 most representative plans of the extracted nodules volumes have to be selected. To meet these requirements, the sagittal, coronal and axial anatomical planes, which divide each dimension by half, are extracted. This process of selecting the planes is illustrated in Figure 17.

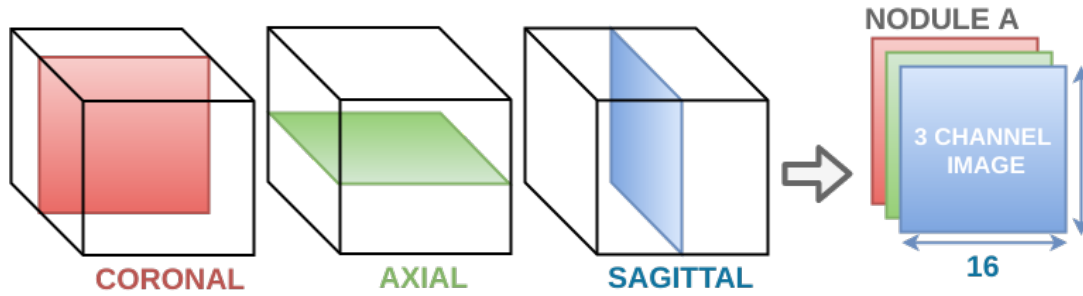


Figure 17 – Converting a extracted volume of interest into a 3 channel image.

In the fifth and last step of this stage, the collection of nodules images is submitted to a sequence of treatments: centralization, normalization, and formation. In the centralization, each channel of each image has its average value computed and, later, subtracted from its elements. In the Normalization, these channels are normalized by their respective standard deviations. Finally, the images are resized in height and width to meet the input resolutions specified by each architecture explored in this work. For example, the input resolution of InceptionV3, Xception, InceptionResNetV2 architectures is 299x299x3; the input resolution of MobileNet, ResNet50, VGG16, VGG19, NASNetMobile, DenseNet169 and DenseNet201 architectures is 244x244x3; and the input resolution of NASNetLarge architectures is 331x331x3.

4.1.2 Problem Reduction

During the LIDC/IDRI database design process, radiologists were instructed to assign a level of malignancy between 1 and 5 for each node detected. However, the vast majority of studies in the literature (using the LIDC/IDRI database) restrict the task of

lung nodule malignancy classification to a binary classification problem (DENG et al., 2009; AERTS et al., 2014; SHEN et al., 2015; HUSSEIN et al., 2017; SHEN et al., 2017; ZHU et al., 2017). Therefore, in order to maintain a methodological standard among the comparative works, a reduction procedure is applied. In it, the levels 1 and 2 are combined and labeled as non-malignant nodules, while levels 4 and 5 are combined and labeled as malignant nodules. The number of samples distribution before and after the reduction process is illustrated in Figure 18.

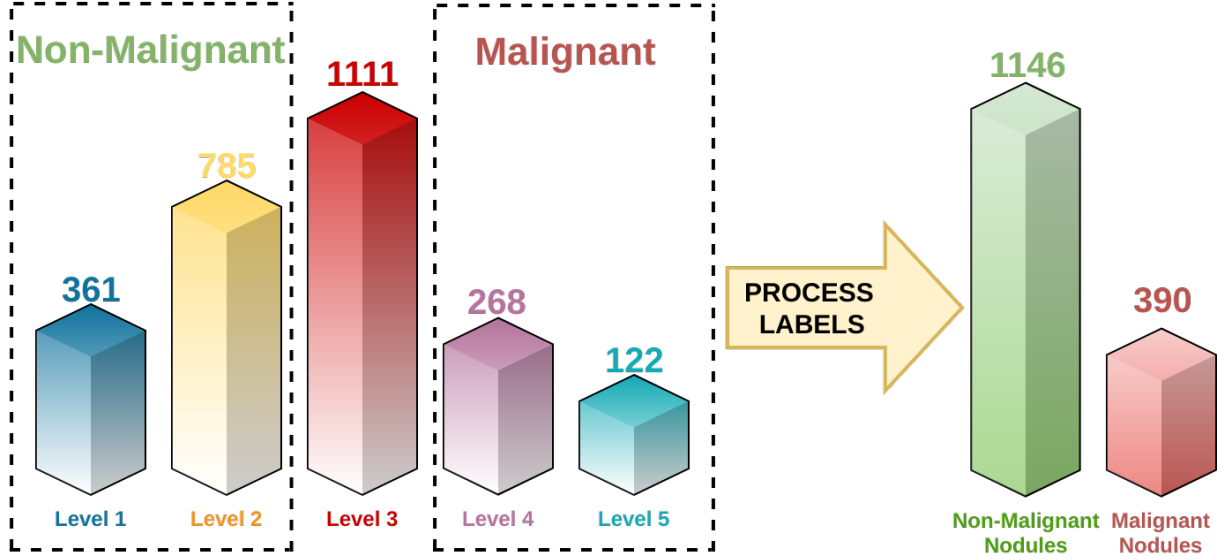


Figure 18 – Illustration of the nodule malignancy levels distribution before and after the reduction process.

4.2 Convolutional Neural Networks as feature extractors

The second section of this methodology details the architectures construction and initialization process, their training (with the reference database) and their transformations into feature extractors.

4.2.1 Architectures Construction and Initialization

Over the last few years, especially after 2012, the performance in the object detection task has significantly increased with the development of several CNN architectures.

However, since it is unfeasible to evaluate the hypothesis explored in this work in all architectures elaborated by the scientific community, 11 reputable architectures (many of them are winners of complex challenges such as ILSVRC (RUSSAKOVSKY et al., 2015) and COCO (LIN et al., 2014)) are selected to represent the CNN method class.

More specifically, the computational graph representing each one of the architectures VGG16, VGG19, MobileNet, Xception, InceptionV3, ResNet50, Inception-ResNet-V2,

DenseNet169, DenseNet201, NASNetMobile, and NASNetLarge is implemented and initialized as specified in their respective papers.

4.2.2 Architectures Training

The second stage of this section consists of training the 11 architectures contemplated in this work in the reference database, that is, the database from which the "knowledge" is transferred.

Based on the premise that the characteristics/patterns learned by a CNN to discriminate a set of classes are more generic than those learned to discriminate only a subset of those classes, the reference database used in this work should have as many classes as possible. Further, since all 11 architectures were elaborated with the objective of solving the problem represented by the ImageNet database, which consists of 1.2 million non-medical images and divided into 1000 categories, the training methodologies used by each of these architectures are documented in detail in their respective articles. Thus, the ImageNet database, which has some samples shown in Figure 19, is selected to be used as a reference database.

4.2.3 Converting CNNs into feature extractors

In the third and last step of this section, the CNNs trained in the reference dataset are converted into feature extractors. However, to accomplish this, it is necessary to understand the four transformations performed by these neural networks.

First, the CNN input image, that is, the image to be processed, is initially submitted to a sequence of non-linear transformations, which are defined according to the architecture used. More specifically, this process converts the input image into a set of small arrays (in none of the architectures used in this work the resolution of these arrays is larger than 11x11x1). Subsequently, each of these matrices is resized to a one-dimensional vector. Then, the set of these vectors is concatenated, thus forming a single vector. This vector, in turn, can be interpreted as the features extracted by CNN.

Finally, these feature vectors are submitted to one or more fully connected layers. In general, this process resembles the processing performed by the Perceptron or the MLP classifiers. It should be noted that the activation functions used in these layers are the ReLU or Sigmoid, for the first layers, and Softmax, for the last one (if the network has only one layer fully connected the activation function used is the Softmax).

However, as illustrated in Figure 21, the last fully-connected layers (used to train the network in the reference database) are removed from the model.

With that, when an image is submitted to this modified architecture, the result can no longer be interpreted as a set of probabilities of an input image belonging to a

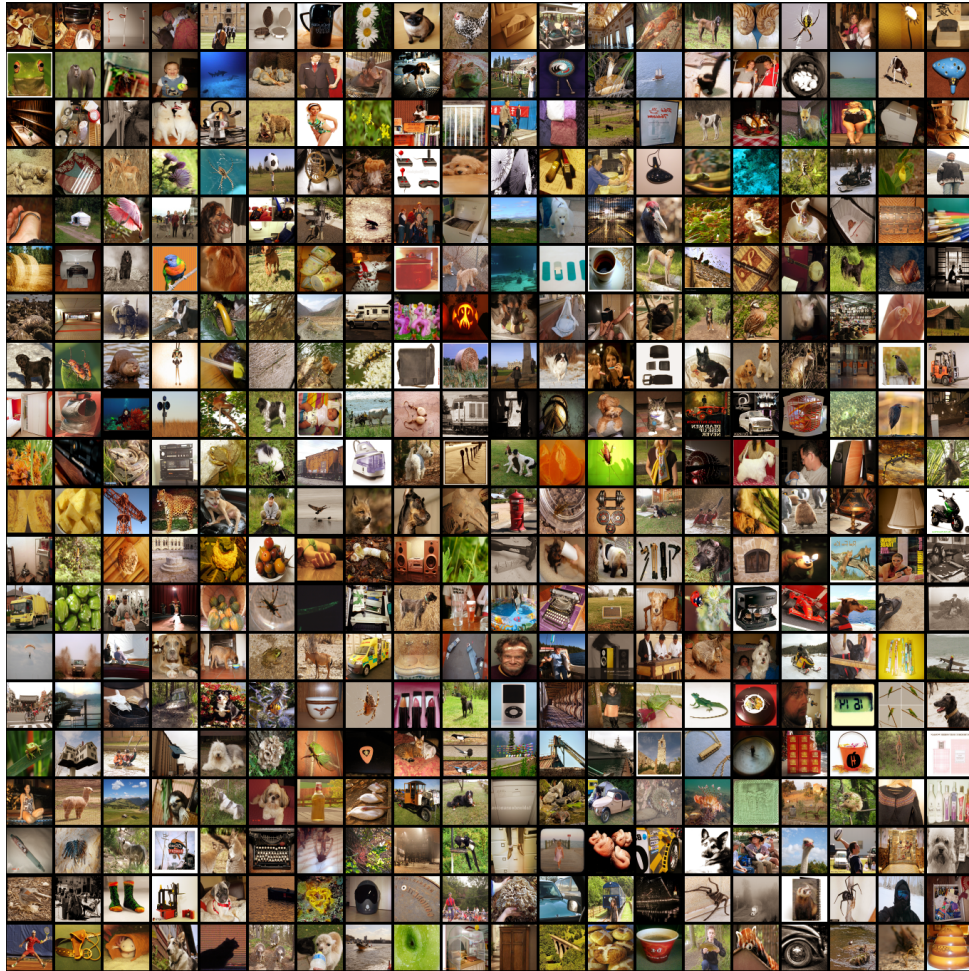


Figure 19 – Examples of non-medical images from the reference dataset (used to train 11 CNNs explored in this work).

given class of the reference dataset (ImageNet), but rather as an information vector, which would be used by fully connected layers to compose the class probabilities.

4.3 Evaluating the proposed approach

In this section, the methodology used to evaluate the performance of the 11 feature extractors implemented in this work is detailed.

4.3.1 Extracting features from LIDC/IDRI database

First, each of the architectures converted into a feature extractors is applied to the target database, that is, the LIDC/IDRI database after preprocessing performed in the first section of this chapter.

Furthermore, since most applied classifiers process the same image more than once (during their training), it is preferable to convert the whole target database into 11 new feature sets (one for each modified architecture) before training it. This avoids the

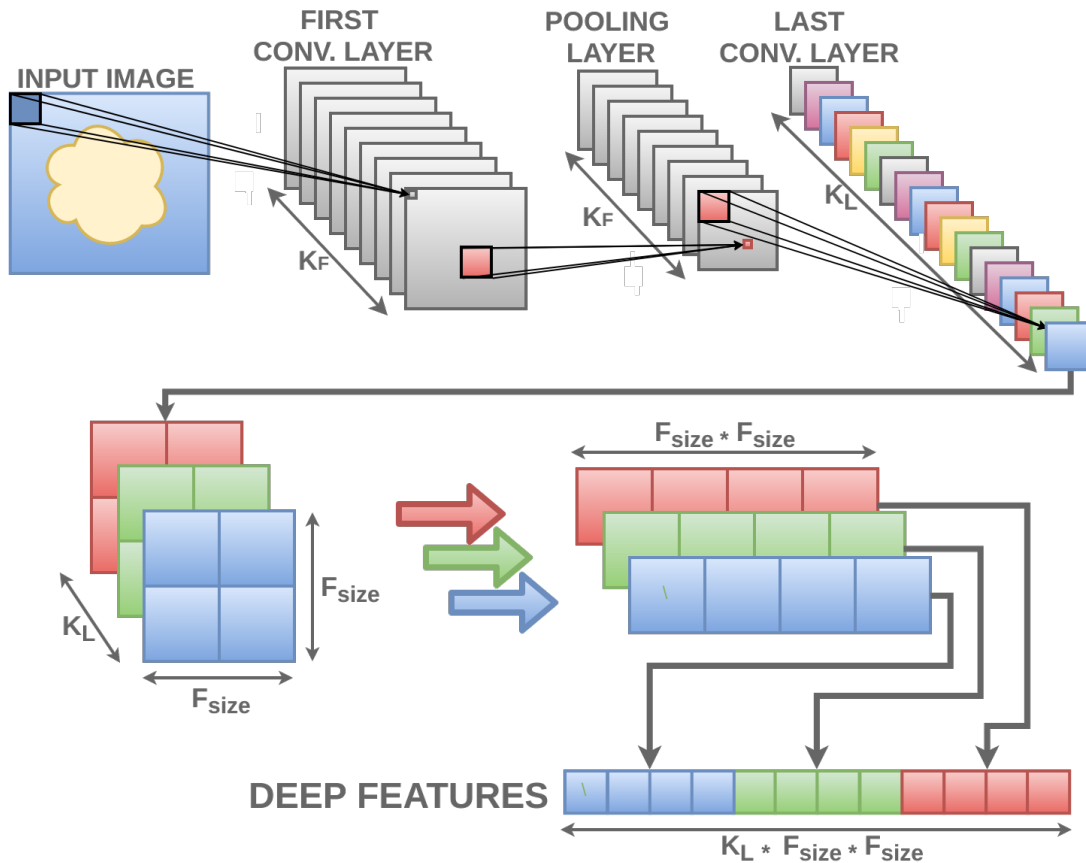


Figure 20 – Illustration of the first two steps of a CNN. In the first, the input image is submitted to a sequence of Convolutional and Subsampling (Pooling) layers. In the second, the results of the first step are resized and concatenated into a single list, called here 'deep features'

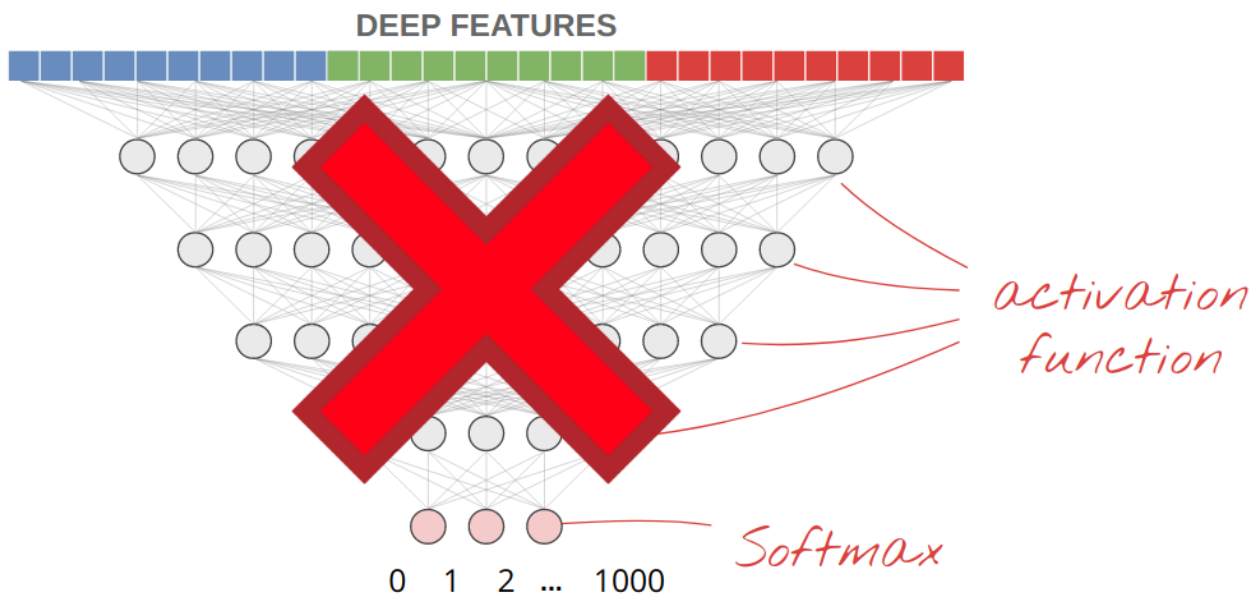


Figure 21 – Illustration of the process used to transform a CNN into a feature extractor.

computational cost of performing the extraction task more than once, thus reducing the time consumed during the evaluation process. Also, the extraction process performed for each explored architectures is illustrated in Figure 22.

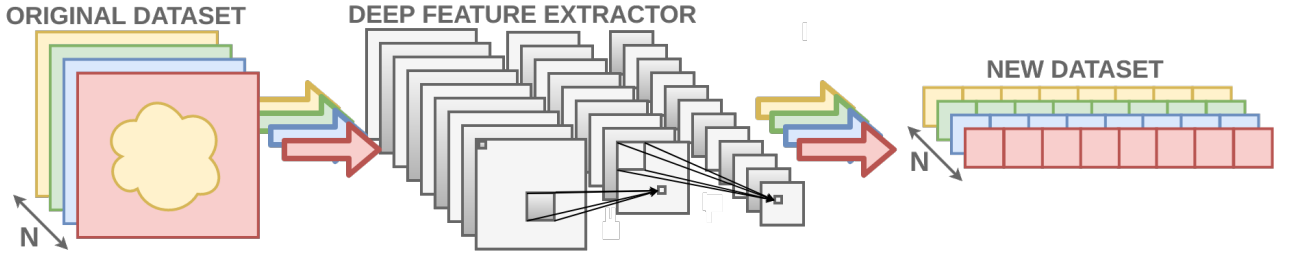


Figure 22 – Illustration of a deep feature extraction process in a given dataset.

4.3.2 Malignancy Classification

Next, 5 classification methods are applied to the 11 sets of extracted characteristics: Naive Bayes, KNN, SVM, MLP, and RF.

In the Naive Bayes, a Gaussian distribution is used as probability density function. In the KNN classifier the hyperparameter “number of neighbors” was chosen through a grid search, testing the odd values from 1 to 11.

Moreover, for the SVM, MLP and RF classifiers, the hyperparameters were chosen through a random search with 20 iterations over each fold of a 10-fold cross-validation process, which is illustrated in Figure 23 .

More specifically, the random search on SVM hyperparameters explored ‘C’ and ‘Y’ with 2^{-5} , 2^{-4} , 2^{-3} , ..., 2^{15} and 2^{-15} , 2^{-14} , ..., 2^3 values respectively (HSU et al., 2003).

Besides, the MLP was trained using the Adam algorithm (KINGMA; BA, 2014) with an initial learning rate of 5^{-4} , learning rate decay of 0.01, the maximum number of 1000 iterations, tolerance for the optimization of 1^{-4} and epsilon of 1^{-8} . Beyond that, a random search was performed to find the number of neurons used in the hidden layer. This search ranged from 2 to 1000 neurons.

Lastly, the random search for RF was used to find: which number of features to consider when looking for the best split (1, 2, 3, ... or 10); which tree maximum depth should be used (6 or none); which minimum number of samples required to split an internal node (1, 2, 3, ... or 10); which minimum samples per leaf should be used (1, 2, 3, ... or 10); whether bootstrap samples should be used when building trees or not, and which function should be used to measure the quality of a split (gini or entropy). Furthermore, RF train process used 3000 estimators.

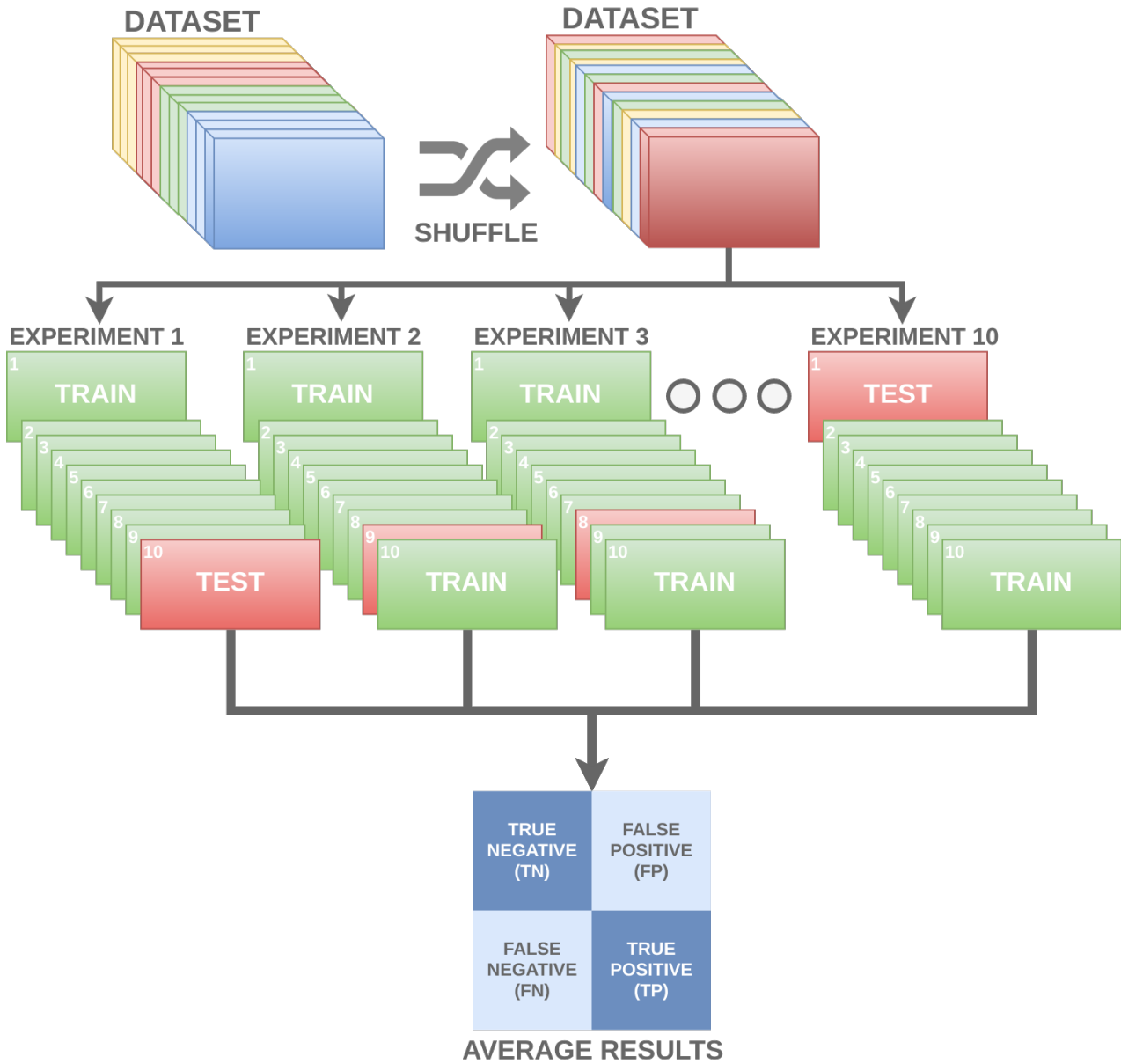


Figure 23 – Illustration of the 10-Fold Cross Validation process in a shuffled dataset.

4.3.3 Evaluation metrics

Finally, to numerically compare the classifications performances the evaluation metrics: accuracy (ACC), sensitivity (True Positive Rate, TPR), positive predictive value (PPV), F-Score, and the area under the ROC-curve (AUC), are computed. More specifically, ACC is the number of nodules correctly classified overall evaluated nodules; TPR is the rate of malignant nodules correctly classified; PPV is the number of malignant nodules correctly classified over the number of samples classified as malignant nodules; F-Score is the harmonic mean between TPR and PPV; and AUC is the area under the curve of TPR against the rate of non-malignant nodules correctly classified at various threshold configurations.

To accomplish this, a confusion matrix of each classification process is computed

using the average results from the 10-Fold Cross Validation. An example of this confusion matrix is illustrated in Figure 24, in which, true positive (TP) is the number of malignant nodules correctly classified; true negative (TN) is the number of the malignant nodules correctly classified; false positive (FP) is the number of non-malignant nodules wrongly classified; and false negative (FN) is the number of malignant nodules wrongly classified.

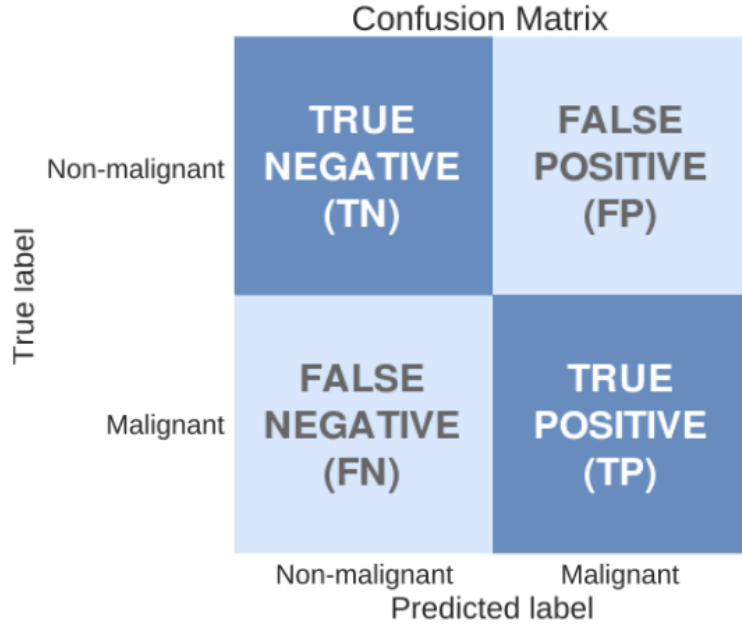


Figure 24 – Illustration of a confusion matrix for the lung nodule malignancy classification task performed in this work.

Based on the this average confusion matrix values the evaluation metrics ACC, TPR, PPV, and F-Score can be mathematically described as the following equations:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}, \quad (4.1)$$

$$TPR = \frac{TP}{TP + FN}, \quad (4.2)$$

$$PPV = \frac{TP}{TP + FP}, \quad (4.3)$$

$$F - Score = \frac{PPV \times TPR}{PPV + TPR}. \quad (4.4)$$

Moreover, to compute AUC evaluation metric 2 steps are required. In the first, the validation samples are divided into 2 groups, the non-malignant nodules and the malignant ones. Next, each of these sets is submitted to a prediction process of a given classifier

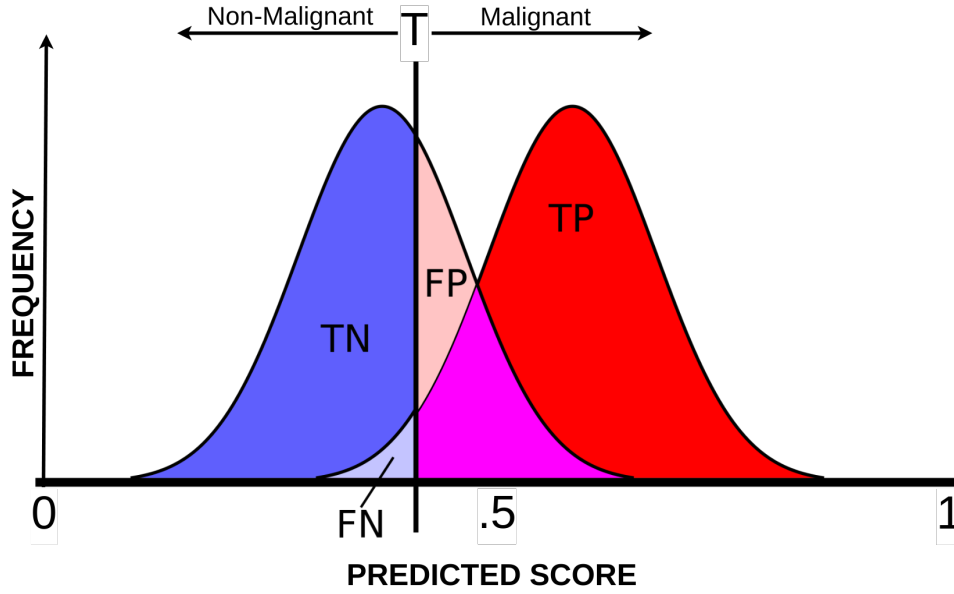


Figure 25 – Prediction probability histogram of a binary problem. Where TP, TN, FP and FN are true positive, true negative, false positive and false negative, respectively. T is the threshold between what is classified as class 0 and class 1.

and the correct class prediction scores are used to compute a histogram. Then, these 2 histograms are plotted into a graph, which has a sample illustrated in Figure 25.

In this graph, it is observed that, as the parameter T varies, the relation between the values of TP, TN, FP and FN changes, and consequently, the same happens with the TPR and False Positive Rate (FPR) metric, which is represented by the following equation:

$$FPR = \frac{FP}{FP + TN}. \quad (4.5)$$

With this, it is possible to compute and plot a graph with the values of TPR and FPR metrics for all values of T between 0 and 1. This graph is known as the Receiver Operational Characteristic (ROC) curve. An example of this curve is represented in Figure 26.

Further, computing the evaluation metric AUC is equivalent to computing the integral of the curve between the interval 0 and 1 in Figure 26 and can be mathematically described by the following equation:

$$\int_0^1 TPR(T) \cdot FPR'(T) dT \quad (4.6)$$

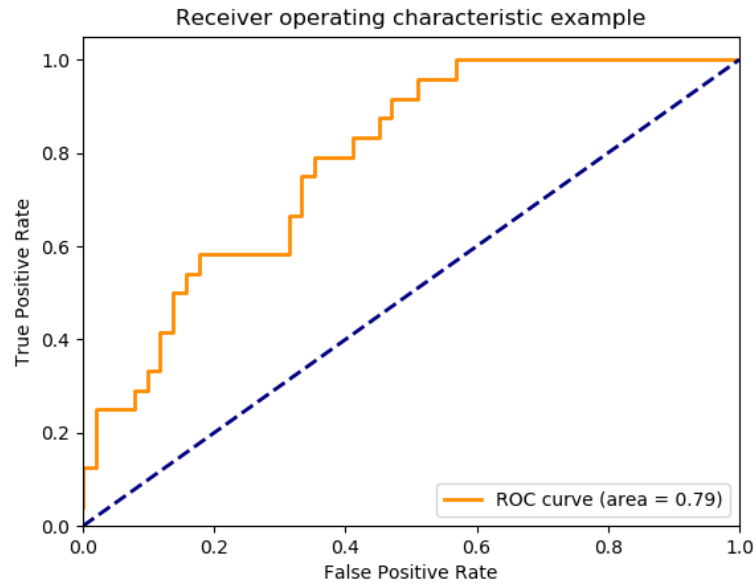


Figure 26 – Illustration of a receiver operating characteristic curve (goo.gl/oLLnBz).

4.3.4 Hardware Settings

It is worth noting that all experiments are conducted under an Ubuntu 17.10 64 bits composed of an Intel Xeon processor with two cores 2.3 GHz and 13 GB RAM. Furthermore, the deep extractors are implemented using the Keras 2.1.4 API with a Tensorflow 1.8 backend and classifiers are implemented using the Scikit-learn 0.19.1 library.

5 Results

The results of this dissertation are presented in two stages. In the first, 66 combinations, between CNN-based feature extractors and classifiers, have their evaluation metrics compared. Then, the best combination obtained in the first step has its results compared to related works found in the academic literature.

5.1 Evaluating Explored Combinations

More specifically, as shown in Table 1, in this first step of the results, 66 combinations have their AUC, ACC, TPR, F-Score and PPV metrics compared. Each combination consists of a feature extractor (based on one of the architectures VGG16, VGG19, MobileNet, Xception, InceptionV3, ResNet50, Inception-ResNet-V2, DenseNet169, DenseNet201, NASNetMobile or NASNetLarge) and a classifier (Bayes, KNN, MLP, RF, Linear SVM or SVB RBF).

In order to establish a criterion to select the best combination explored, the average between the evaluation metrics is computed for all combinations and then ordered incrementally. Subsequently, the 5 combinations with higher averages are illustrated in a bar graph in Figure 27, where each color represents a combination. In this figure, it is observed that the architectures Xception, Resnet50 and Inception-ResnetV2 and the classifiers SVM RBR, MLP and RF stand out among the others.

It can be argued that a common aspect of all these architectures is the concept of "*skipping connections*", introduced by the article that also proposed the ResNet50 architecture. However, this does not seem to be a decisive factor, given that the DenseNet architecture, which takes this concept to the extreme, did not obtain good results.

Further, an alternative interpretation for these results is that the "*skipping connections*" between a few layers enables the training of deeper models, that is, models with greater discriminative power. However, when the "*skipping connections*" is performed between many layers (as in DenseNet), the model becomes very specialized in solving the task for which it was initially trained, thus preventing a transfer of this from learning.

According to Figure 27, despite a similar average between metrics, the approaches that adopt the SVM-RBF classifier have low PPV rates and high TPR rates, while the ones that use the MLP classifier present an inverse behavior, that is, low TPR rates and high PPV rates.

However, since the empirical error is sensitive to the total number of incorrectly classified samples, and the database used in this work has almost 4 times more samples of

Table 1 – Accuracy (ACC), Area Under Curve (AUC), F-Score, Sensitivity (TPR), Positive Predictive Value (PPV) for the evaluated combinations between deep feature extractors and classifiers.

Model	Classifier	ACC(%)	AUC(%)	F-Score(%)	TRP(%)	PPV(%)
Xception	Bayes	83.66	83.87	62.49	54.87	73.66
	MLP	89.91	92.90	79.11	75.13	84.03
	Nearest Neighbors	87.89	89.99	76.36	77.44	75.82
	Random Forest	88.61	90.98	75.48	70.00	82.64
	SVM Linear	86.85	92.55	76.68	84.87	70.44
	SVM RBF	88.28	92.93	78.61	84.87	73.40
ResNet50	Bayes	83.20	82.65	63.08	56.41	72.19
	MLP	89.13	92.25	76.24	69.23	85.37
	Nearest Neighbors	86.85	89.38	73.21	70.77	76.23
	Random Forest	88.80	92.15	77.75	76.92	78.97
	SVM Linear	86.98	92.67	77.08	85.64	70.51
	SVM RBF	88.41	93.19	78.83	85.38	73.48
MobileNet	Bayes	83.21	89.57	69.05	73.59	65.38
	MLP	87.76	91.08	74.81	72.05	78.09
	Nearest Neighbors	87.57	91.04	72.94	66.41	81.20
	Random Forest	87.18	91.64	73.13	68.46	78.68
	SVM Linear	84.25	89.64	70.70	75.13	67.07
	SVM RBF	88.67	91.79	75.73	70.51	82.21
VGG16	Bayes	77.87	85.34	24.81	14.62	90.83
	MLP	83.53	88.55	67.07	66.41	68.07
	Nearest Neighbors	84.05	85.23	64.55	57.18	74.76
	Random Forest	84.83	89.82	60.09	45.90	88.70
	SVM Linear	80.27	87.90	67.61	80.77	58.28
	SVM RBF	86.53	89.40	72.21	69.49	75.86
VGG19	Bayes	76.43	83.60	15.26	8.46	86.67
	MLP	87.69	90.75	74.10	70.00	79.33
	Nearest Neighbors	86.00	88.31	70.25	65.13	77.19
	Random Forest	87.11	89.55	73.27	70.26	76.90
	SVM Linear	82.03	89.61	69.90	82.31	60.84
	SVM RBF	84.90	89.82	72.40	78.21	67.74
InceptionV3	Bayes	80.08	84.50	63.30	67.69	59.59
	MLP	87.83	91.79	74.45	70.77	81.08
	Nearest Neighbors	87.04	89.46	73.61	71.03	76.84
	Random Forest	85.81	89.91	69.21	63.08	77.14
	SVM Linear	87.50	91.41	77.31	83.85	71.88
	SVM RBF	88.87	92.50	77.34	74.62	80.44
InceptionResNetV2	Bayes	83.52	83.91	62.63	54.87	73.46
	MLP	89.00	92.63	77.10	73.33	82.70
	Nearest Neighbors	87.50	88.91	75.51	75.90	75.56
	Random Forest	86.72	90.22	75.05	78.97	71.80
	SVM Linear	87.24	92.53	77.34	85.38	70.90
	SVM RBF	87.04	92.48	76.84	84.62	70.61
NASNetMobile	Bayes	82.81	85.23	54.58	41.28	82.13
	MLP	87.37	89.98	72.24	65.64	81.03
	Nearest Neighbors	85.10	86.98	69.55	67.18	72.63
	Random Forest	84.51	87.66	70.33	73.08	68.09
	SVM Linear	83.73	88.92	72.22	83.33	64.00
	SVM RBF	85.22	89.84	73.67	81.28	67.81
NASNetLarge	Bayes	83.21	85.42	66.44	65.64	67.41
	MLP	88.08	90.79	74.26	69.23	82.41
	Nearest Neighbors	87.63	87.97	74.53	71.54	78.07
	Random Forest	87.56	90.17	72.06	63.33	84.02
	SVM Linear	87.57	91.00	76.75	81.03	73.42
	SVM RBF	88.28	90.09	75.32	70.77	80.98
DenseNet169	Bayes	81.32	82.25	45.64	31.28	86.80
	MLP	86.52	89.67	68.94	61.03	82.26
	Nearest Neighbors	86.46	89.10	72.72	71.54	74.57
	Random Forest	86.53	91.16	74.41	77.44	72.07
	SVM Linear	86.13	91.35	75.15	82.82	68.95
	SVM RBF	86.66	90.24	72.91	70.77	75.70
DenseNet201	Bayes	81.76	86.68	47.09	32.56	87.56
	MLP	87.18	91.83	71.01	62.31	83.84
	Nearest Neighbors	87.11	89.57	73.12	69.49	77.76
	Random Forest	87.50	91.70	71.80	63.08	83.83
	SVM Linear	85.81	92.47	75.26	84.62	68.14
	SVM RBF	87.82	90.66	74.95	72.05	78.78

non-malignant nodules (unbalanced), the behavior of the approaches favoring PPV metrics can be explained as a bias in the classification process.

Furthermore, in medical applications the "cost" associated with a false negative (when the method classifies a malignant nodule as non-malignant) is greater than with a false positive (when the method classifies a non malignant nodule as malignant), since when a malignant nodule is detected the patient is submitted to confirmatory exams (such as blood tests or biopsies). Whereas, when the nodule is classified as non-malignant, the patient is simply released, which can be life-threatening if the diagnosis is a false negative.

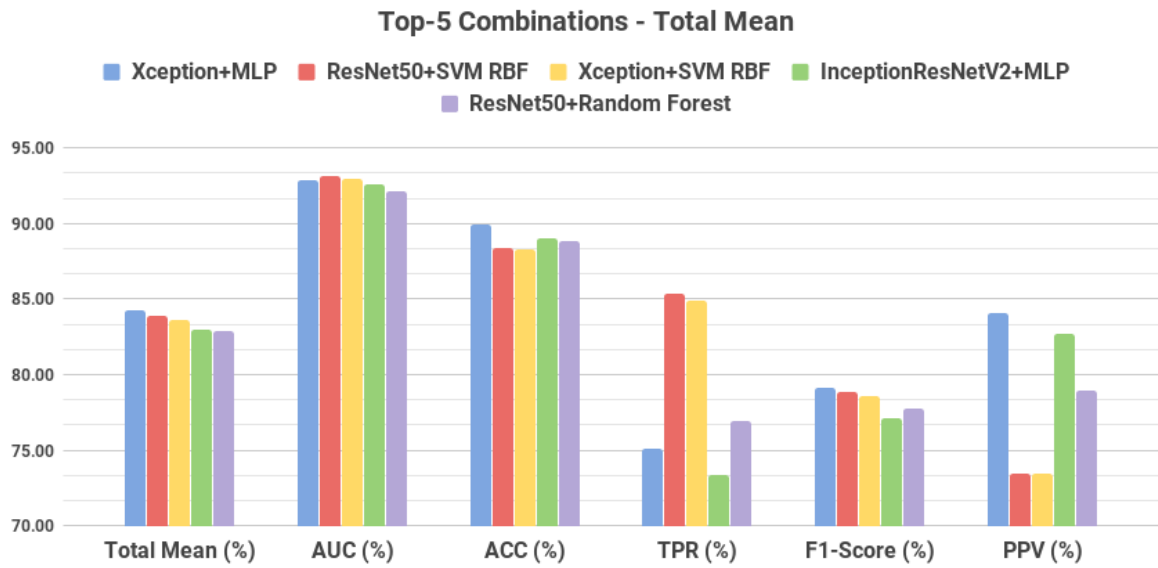


Figure 27 – Comparison of the 5 experiments with highest average between AUC, ACC, TPR, F-Score and PPV evaluation metrics. Each of these experiments is identified by a different color and represents the combination of a CNN-based feature extractor and a machine learning technique.

For this reason, the criterion used to select the best combination was changed. Instead of computing, the average among all the metrics is preferable to compute the average only between the AUC and TPR metrics, since these are the metrics that are most sensitive to the false negative rates. Thus, the 5 combinations with the highest values of this new mean are illustrated in a bar graph in Figure 28, where each group of bars represents an evaluation metric and each color a combination.

In the Figure 28, it is observed that regardless the architecture used as feature extractor, all the selected approaches use the SVM classifier (with a Linear kernel or an RBF kernel).

Another aspect to be highlighted is that, according to the new criterion adopted, the 2 best results use the ResNet50 architecture as a feature extractor. When combined with the SVM-Linear classifier this architecture obtains the highest value of the TPR

metric from all approaches compared in Figure 28, reaching 85.64%. However, this TPR value is only 0.26% greater than the ResNet50 and SVM-RBF combination, which achieves a TPR of 85.38%. Moreover, the combination with the SVM-RBF surpass the values achieved by the combination with the SVM-Linear in all other metrics explored. More specifically, 1.43 % in ACC, 0.52% in AUC, 1.75% in F-Score and 2.97% PPV.

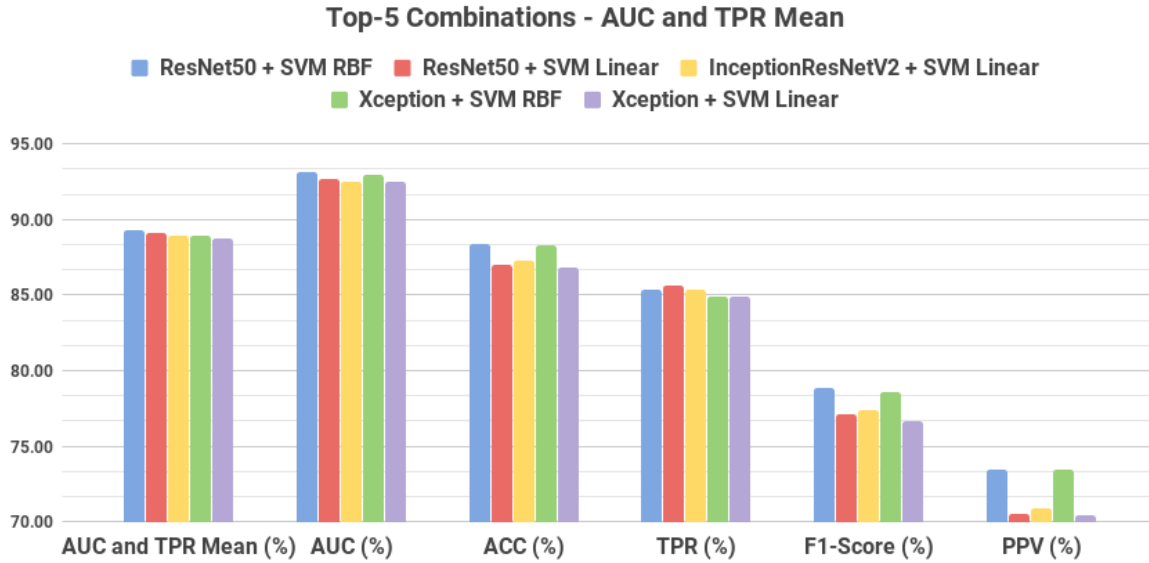


Figure 28 – Comparison between the 5 experiments with the highest average of the AUC and TPR evaluation metrics. Each of these experiments is identified by a different color and represents the combination of a CNN-based feature extractor and a machine learning technique.

Notwithstanding, the feature extraction time does not represent a critical aspect in the lung nodule malignancy classification task, the information may be useful in case the reader wishes to extend the approach herein applied to solve real-time tasks.

For this reason, the execution times of each extractor used are shown in Figure 29. In this bar graph, it is observed that the architecture with lower extraction time is MobileNet with 94ms, 8.8 times faster than DenseNet201 with 828ms. Other models that stand out are VGG16, VGG19 and ResNet50, presenting 154, 180 and 196 milliseconds respectively.

Finally, based on the experiments carried out in this first stage, the combination between the deep feature extractor based on the ResNet50 model and the SVM-RBF classifier is selected as the best approach among the explored ones. This is because, this combination achieves the highest average between the AUC and TRP evaluation metrics, which are essential for medical applications. Further, the results obtained by this approach are shown in Figure 30.

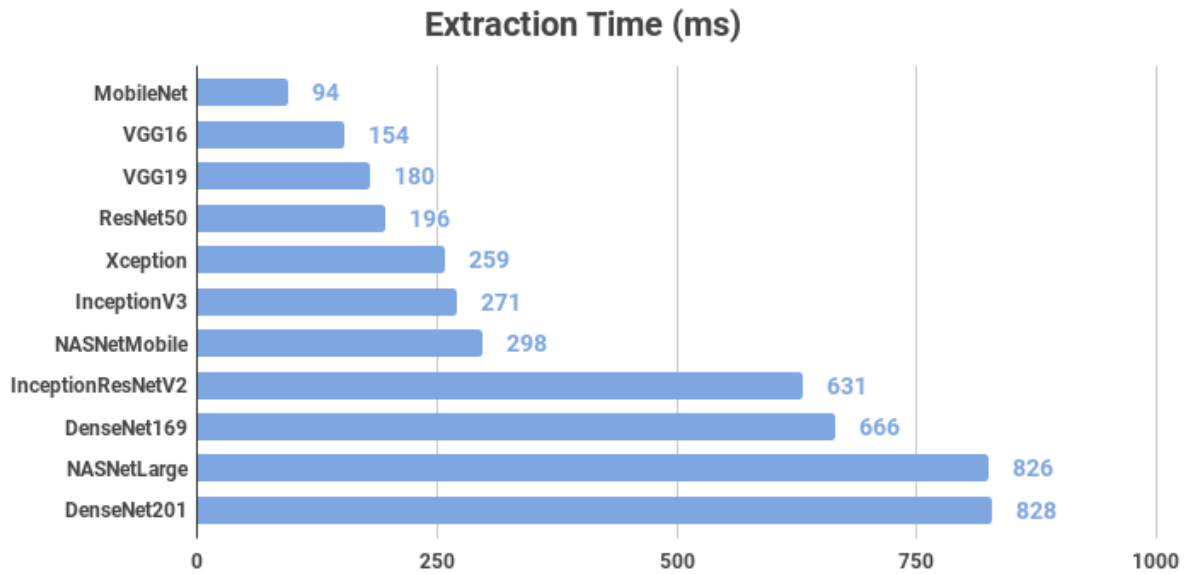


Figure 29 – The average execution time (in milliseconds) of each Deep feature extractor explored in this work.

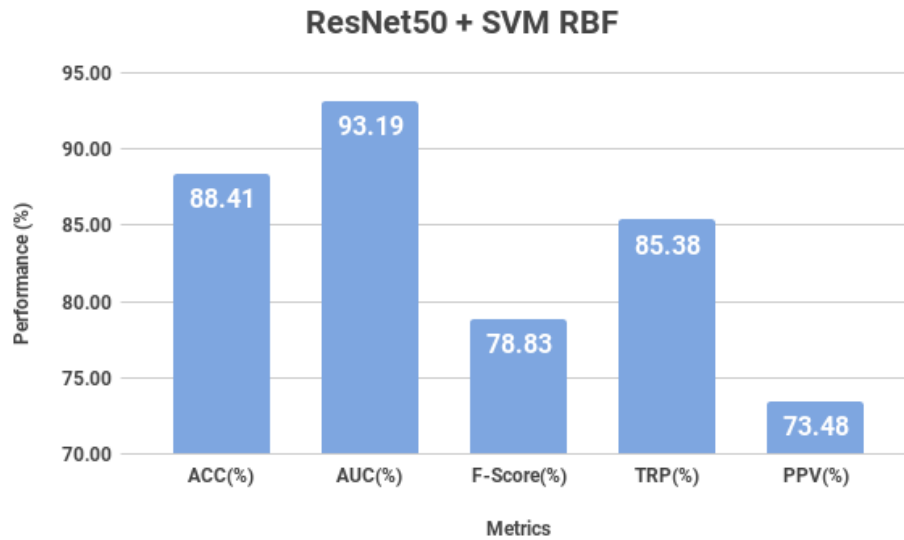


Figure 30 – The values of AUC, ACC, TPR, F-Score, and PPV metrics from the selected combination.

5.2 Comparison of the proposed approach with related works

In this second stage of the results, the performance of the selected approach is compared to related work through its evaluation metrics. As shown in Figure 31, 13 approaches for lung nodule malignancy classification using the LIDC/IDRI database were found in the academic literature.

However, nearly half of these techniques (6 of them) computed their metrics using

only a subset (less than 1000 samples) of the LIDC/IDRI database.

This selection of samples, despite justified by the methodology of the respective works, reduces the reliability of their results, since it makes the database less representative. For this reason, only those approaches that use more than 1000 samples in their studies are considered for the comparison performed in this second stage of the results.

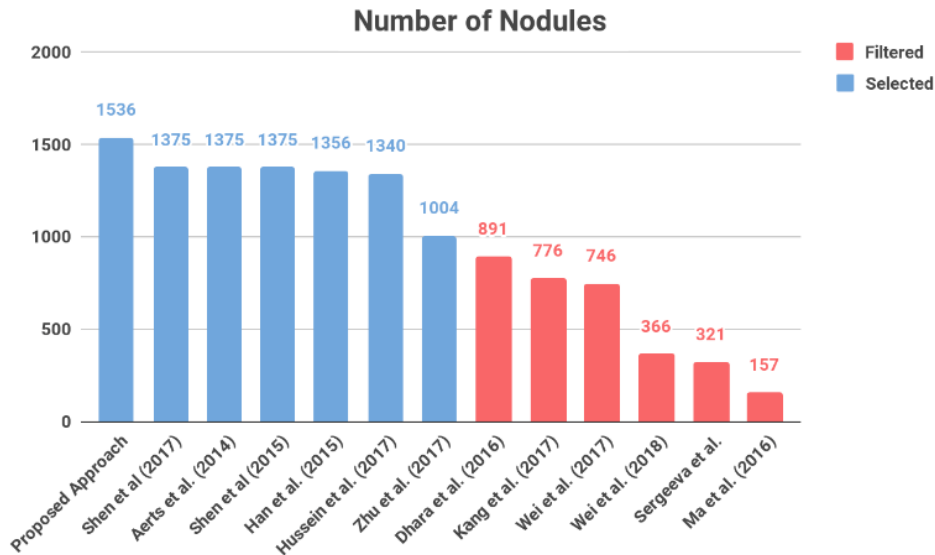


Figure 31 – List of related works sorted by the number of nodules evaluated.

In Figure 32, a comparison between related works is performed. In it, each approach is represented by a set of bars, where each color represents an evaluation metric. Also, it is worth noting that, despite being computed in this work, the PPV and F-score metrics were not shown in Figure 32, since none of the related works provided them.

Among the compared approaches, it is easy to see that the combination proposed by this work obtained the highest value of the AUC evaluation metric (93.13%). Slightly higher than those achieved by Shen et al. (2017) (93%) and Han et al. (2015) (92.7%); and considerably higher than the one achieved by Aerts et al. (2014) (89%).

In addition, the TPR values obtained by the combination of ResNet50 and SVM RBF (85.38 %) are substantially higher than the ones provided by Shen et al. (2017) (77%), but slightly lower than those achieved by Aerts et. al. (2014) (87%).

Despite achieving a higher ACC than most of the comparative approaches (Shen et al. (2015) (86.80%), Shen et al. (2017) (87.14%) and Han et al. (2015) (83.21%)), the approach exploited by this work, which obtained 88.41%, was surpassed by the approaches of Hussein et al. (2017) (91.26%) and Zhu et al. (2017) (90.44%).

However, in unbalanced databases (such as LIDC/IDRI) the ACC metric by itself does not mean much. Mostly because, even if the method under analysis classifies all samples with the label of the most frequent class in the database, the obtained ACC value

is high. That is, despite the approaches presented by the works of Hussein et al. (2017) and Zhu et al. (2017) achieve higher values than those obtained by the combination between ResNet50 and SVM RBF, their results do not show the methodological rigor necessary for an effective comparison, since they only provide the ACC metric of their experiments.

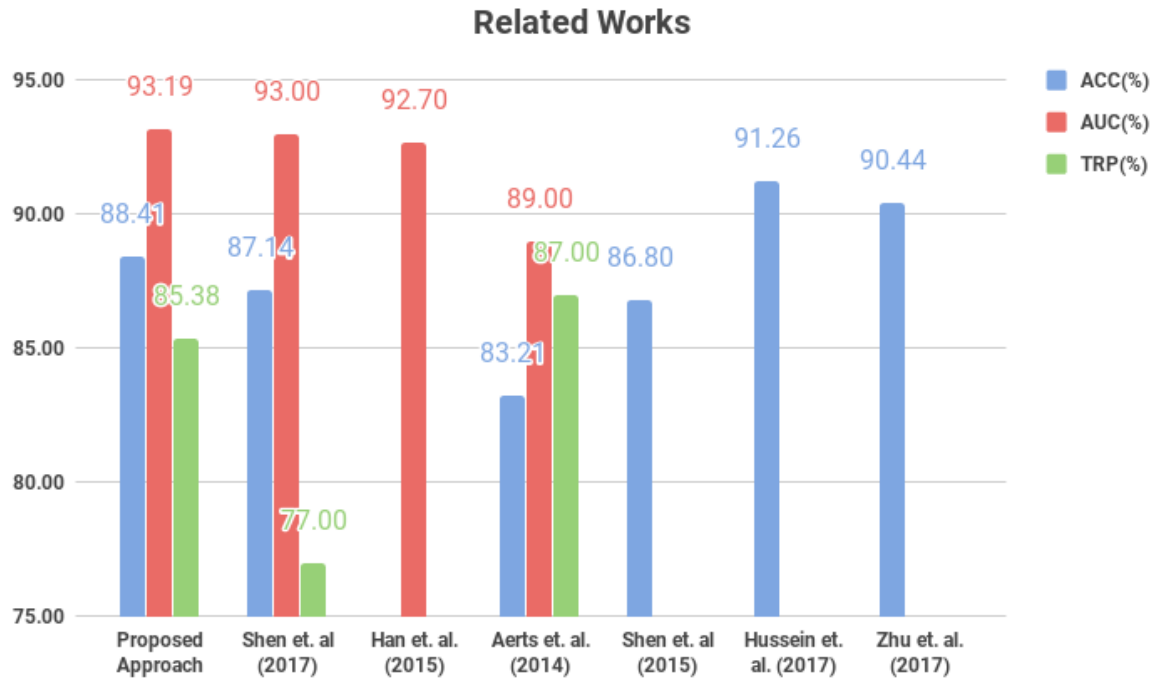


Figure 32 – The results from the proposed approach and related works are compared in a bar chart using the AUC, ACC, TPR, metrics. Each one of these metrics is indicated by a color.

6 Conclusion

A new approach to perform lung nodule malignancy classification based on the concepts of Transfer Learning and Deep Learning is proposed in this dissertation.

First, the VGG16, VGG19, MobileNet, Xception, InceptionV3, ResNet50, Inception-ResNet-V2, DenseNet169, DenseNet201, NASNetMobile and NASNetLarge are built and trained in the ImageNet database, which is composed of more than 1.2 million non-medical images divided into 1000 categories.

Later, each of these networks is converted into a feature extractor and combined with Bayes, KNN, MLP, RF, Linear SVM and SVM RBF machine learning techniques.

Then, each one of these 66 combinations is evaluated in the classification of 1537 lung nodules, being 1146 non-malignant and 390 malignant. More specifically, the set of samples used to represent the nodule malignancy classification problem in this work is obtained from the LIDC / IDRI database, which contains 1018 chest CT scans and their respective diagnoses, performed by 4 experienced radiologists.

Finally, these combinations are compared with each other and with the approaches proposed by Shen et al. (2017), Hussein et al. (2017), Zhu et al. (2017), Han et al. (2015), Shen et al. (2015) and Aerts et al. (2014). This comparison uses the evaluation metrics AUC, ACC, TPR, F-Score and PPV, computed from a Cross-Validation with 10 experiments.

From these tests, it is concluded that among the evaluated combinations the one whom the extractor is based on the ResNet50 architecture and the machine learning technique applied is SVM-RBF, is the best one. This approach has the highest values of AUC and average between TPR and AUC, and also the second highest metrics average. More specifically, this combination achieves 93.13% of AUC, 85.38% of TPR, 78.83% of F-Score, 73.48% of PPV and 88.41% of ACC.

Further, when compared with related works, the selected approach also has the highest value of AUC and average between TPR and AUC metrics, thus obtaining superior results.

Therefore, the main contributions of this dissertation are: a detailed methodology for processing LIDC/IDRI exams and medical annotations; the exploration of several CNN architectures as feature extractors using transfer of learning; an extensive statistical analysis of the experiments results, thus serving as a reference for future works; and the proposition of an effective approach to classify the malignancy of lung nodules in chest CT images.

It is worth mentioning that a preview of this work was published in the 31st

International Symposium on Computer-Based Medical Systems (CBMS 2018), indicating the recognition of such contributions by the academic community.

Overall, the deep extractor ResNet50 combined with SVM RBF classifier have achieved satisfactory results and is eligible to be integrated into a CAD system to give specialist doctors a second opinion about the malignancy of a lung nodule.

6.1 Future Works

The entire development of this dissertation, as well as the results obtained, does not exhaust the topics of Transfer of Learning, CNNs or Deep Learning. In this sense, the following future works can be proposed:

- Integrate the approach developed into a CAD system;
- Propose an approach for automatic segment lung nodules;
- Apply the proposed approach in non-lung nodules;
- Combine the features from more than one extractor before the classification process;
- Compare the achieved results with classic feature extractors, such as, HU, LBP, GLCM etc;
- Perform statistical validation testes, such as, Friedman, Wilcoxon etc;
- After training the target architecture in a baseline dataset, train with a subset of the target database (strategy known as fine-tuning).

Bibliography

AERTS, H. J. et al. Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach. *Nature communications*, Nature Publishing Group, v. 5, 2014. Citado 3 vezes nas páginas 21, 51, and 69.

ARMATO, S. G. et al. The lung image database consortium (LIDC) and image database resource initiative (IDRI): A completed reference database of lung nodules on CT scans. *Medical Physics*, Wiley-Blackwell, v. 38, n. 2, p. 915–931, jan 2011. Citado 2 vezes nas páginas 22 and 79.

BENGIO, Y.; GOODFELLOW, I. J.; COURVILLE, A. Deep learning. *Nature*, Citeseer, v. 521, n. 7553, p. 436–444, 2015. Citado 2 vezes nas páginas 33 and 34.

BENGIO, Y.; SIMARD, P.; FRASCONI, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, Institute of Electrical and Electronics Engineers (IEEE), v. 5, n. 2, p. 157–166, mar 1994. Disponível em: <<https://doi.org/10.1109/72.279181>>. Citado na página 40.

BREIMAN, L. Random forests. *Machine Learning*, v. 45, n. 1, p. 5–32, Oct 2001. Citado 2 vezes nas páginas 29 and 30.

CHOLLET, F. Xception: Deep learning with depthwise separable convolutions. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017. Disponível em: <<https://doi.org/10.1109/cvpr.2017.195>>. Citado 2 vezes nas páginas 22 and 42.

CHRISTODOULIDIS, S. et al. Multisource transfer learning with convolutional neural networks for lung pattern analysis. *IEEE journal of biomedical and health informatics*, IEEE, v. 21, n. 1, p. 76–84, 2017. Citado na página 21.

COVER, T.; HART, P. Nearest neighbor pattern classification. *IEEE transactions on information theory*, IEEE, v. 13, n. 1, p. 21–27, 1967. Citado na página 22.

DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: IEEE. *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. [S.l.], 2009. p. 248–255. Citado 3 vezes nas páginas 22, 35, and 51.

DHARA, A. K. et al. A combination of shape and texture features for classification of pulmonary nodules in lung ct images. *Journal of digital imaging*, Springer, v. 29, n. 4, p. 466–475, 2016. Citado na página 21.

ESTEVA, A. et al. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, Springer Nature, v. 542, n. 7639, p. 115–118, jan 2017. Citado na página 21.

FUKUSHIMA, K.; MIYAKE, S. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In: *Competition and cooperation in neural nets*. [S.l.]: Springer, 1982. p. 267–285. Citado na página 33.

- GLOT, X.; BORDES, A.; BENGIO, Y. Deep sparse rectifier neural networks. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. [S.l.: s.n.], 2011. p. 315–323. Citado na página 35.
- HAN, F. et al. Texture feature analysis for computer-aided diagnosis on pulmonary nodules. *Journal of digital imaging*, Springer, v. 28, n. 1, p. 99–115, 2015. Citado 3 vezes nas páginas 21, 66, and 69.
- HAYKIN, S. *Neural Networks and Learning Machines*. McMaster University, Canada: Prentice Hall, 2008. Citado 3 vezes nas páginas 22, 30, and 33.
- HE, K. et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 770–778. Citado na página 22.
- Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *ArXiv e-prints*, mar. 2015. Citado na página 42.
- HINTON, G. E. et al. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012. Citado 2 vezes nas páginas 35 and 36.
- HO, T. K. Random decision forests. In: IEEE. *Document analysis and recognition, 1995., proceedings of the third international conference on*. [S.l.], 1995. v. 1, p. 278–282. Citado na página 29.
- HOWARD, A. G. et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. Citado 2 vezes nas páginas 22 and 43.
- HSU, C.-W. et al. A practical guide to support vector classification. Taipei, 2003. Citado na página 55.
- HUANG, G. et al. Densely connected convolutional networks. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017. Disponível em: <<https://doi.org/10.1109/cvpr.2017.243>>. Citado 3 vezes nas páginas 11, 22, and 44.
- HUSSEIN, S. et al. Risk stratification of lung nodules using 3d cnn-based multi-task learning. In: SPRINGER. *International Conference on Information Processing in Medical Imaging*. [S.l.], 2017. p. 249–260. Citado 7 vezes nas páginas 20, 21, 22, 51, 66, 67, and 69.
- HUYNH, B. Q.; LI, H.; GIGER, M. L. Digital mammographic tumor classification using transfer learning from deep convolutional neural networks. *Journal of Medical Imaging, SPIE-Intl Soc Optical Eng*, v. 3, n. 3, p. 034501, aug 2016. Citado na página 21.
- HWANG, S. et al. A novel approach for tuberculosis screening based on deep convolutional neural networks. In: TOURASSI, G. D.; ARMATO, S. G. (Ed.). *Medical Imaging 2016: Computer-Aided Diagnosis*. [S.l.]: SPIE, 2016. Citado na página 21.
- Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *ArXiv e-prints*, fev. 2015. Citado 2 vezes nas páginas 11 and 39.
- JIN, J.; DUNDAR, A.; CULURCIELLO, E. Flattened convolutional neural networks for feedforward acceleration. *CoRR*, abs/1412.5474, 2014. Disponível em: <<http://arxiv.org/abs/1412.5474>>. Citado na página 39.

- KARPATHY, A.; LI, F. Convolutional neural networks. *Capturado em: <http://cs231n.github.io/convolutional-networks>*, p. 05–10, 2017. Citado 2 vezes nas páginas 11 and 34.
- KARPATHY, A. et al. Large-scale video classification with convolutional neural networks. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2014. p. 1725–1732. Citado na página 22.
- KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. Citado na página 55.
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2012. p. 1097–1105. Citado 3 vezes nas páginas 22, 34, and 35.
- LÄNGKVIST, M.; KARLSSON, L.; LOUTFI, A. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters*, Elsevier BV, v. 42, p. 11–24, jun 2014. Disponível em: <<https://doi.org/10.1016/j.patrec.2014.01.008>>. Citado 3 vezes nas páginas 11, 41, and 42.
- LECUN, Y. et al. Handwritten digit recognition with a back-propagation network. In: *Advances in neural information processing systems*. [S.l.: s.n.], 1990. p. 396–404. Citado 2 vezes nas páginas 20 and 33.
- LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, Institute of Electrical and Electronics Engineers (IEEE), v. 86, n. 11, p. 2278–2324, 1998. Disponível em: <<https://doi.org/10.1109/5.726791>>. Citado 3 vezes nas páginas 11, 33, and 34.
- LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, IEEE, v. 86, n. 11, p. 2278–2324, 1998. Citado 2 vezes nas páginas 33 and 35.
- LIAW, A.; WIENER, M. et al. Classification and regression by randomforest. *R news*, v. 2, n. 3, p. 18–22, 2002. Citado na página 22.
- LIN, T. et al. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. Citado 2 vezes nas páginas 41 and 51.
- LIN, Y.; JEON, Y. Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association*, Taylor & Francis, v. 101, n. 474, p. 578–590, 2006. Citado na página 30.
- LITJENS, G. et al. A survey on deep learning in medical image analysis. *Medical Image Analysis*, Elsevier BV, v. 42, p. 60–88, dec 2017. Citado na página 21.
- MARCUS, P. M. et al. Did death certificates and a death review process agree on lung cancer cause of death in the national lung screening trial? *Clinical Trials*, SAGE Publications Sage UK: London, England, v. 13, n. 4, p. 434–438, 2016. Citado na página 20.
- NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: FÜRKNRANZ, J.; JOACHIMS, T. (Ed.). *ICML*. Omnipress, 2010. p. 807–814. Disponível em: <<http://dblp.uni-trier.de/db/conf/icml/icml2010.html#NairH10>>. Citado na página 35.

NÄPPI, J. J. et al. Deep transfer learning of virtual endoluminal views for the detection of polyps in CT colonography. In: TOURASSI, G. D.; ARMATO, S. G. (Ed.). *Medical Imaging 2016: Computer-Aided Diagnosis*. [S.l.]: SPIE, 2016. Citado na página 21.

PAN, S. J.; YANG, Q. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, Institute of Electrical and Electronics Engineers (IEEE), v. 22, n. 10, p. 1345–1359, oct 2010. Citado na página 21.

PENG, H.; LONG, F.; DING, C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Institute of Electrical and Electronics Engineers (IEEE), v. 27, n. 8, p. 1226–1238, aug 2005. Disponível em: <<https://doi.org/10.1109/tpami.2005.159>>. Citado na página 21.

RODRIGUES, M. B. et al. Health of things algorithms for malignancy level classification of lung nodules. *IEEE Access*, v. 6, p. 18592–18601, 2018. Citado na página 21.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature*, Springer Nature, v. 323, n. 6088, p. 533–536, oct 1986. Disponível em: <<https://doi.org/10.1038/323533a0>>. Citado na página 33.

RUSSAKOVSKY, O. et al. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, v. 115, n. 3, p. 211–252, 2015. Citado 6 vezes nas páginas 34, 36, 37, 38, 41, and 51.

SERGEeva, M. et al. Classification of pulmonary nodules on computed tomography scans. evaluation of the effectiveness of application of textural features extracted using wavelet transform of image. In: *2016 18th Conference of Open Innovations Association and Seminar on Information Security and Protection of Information Technology (FRUCT-ISPIT)*. [S.l.]: IEEE, 2016. Citado na página 21.

SHEN, W. et al. Multi-scale convolutional neural networks for lung nodule classification. In: SPRINGER. *International Conference on Information Processing in Medical Imaging*. [S.l.], 2015. p. 588–599. Citado 5 vezes nas páginas 20, 21, 51, 66, and 69.

SHEN, W. et al. Multi-crop convolutional neural networks for lung nodule malignancy suspiciousness classification. *Pattern Recognition*, Elsevier, v. 61, p. 663–673, 2017. Citado 6 vezes nas páginas 20, 21, 22, 51, 66, and 69.

SHIE, C.-K. et al. Transfer representation learning for medical image analysis. In: IEEE. *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*. [S.l.], 2015. p. 711–714. Citado na página 21.

SHIN, H.-C. et al. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Transactions on Medical Imaging*, Institute of Electrical and Electronics Engineers (IEEE), v. 35, n. 5, p. 1285–1298, may 2016. Citado na página 21.

SIEGEL, R. L.; MILLER, K. D.; JEMAL, A. Cancer statistics, 2017. *CA: A Cancer Journal for Clinicians*, American Cancer Society, v. 67, n. 1, p. 7–30, jan 2017. Citado na página 20.

- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. Disponível em: <<http://arxiv.org/abs/1409.1556>>. Citado 5 vezes nas páginas 11, 22, 36, 37, and 39.
- SRIVASTAVA, N. et al. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014. Citado 2 vezes nas páginas 11 and 36.
- SZEGEDY, C. et al. Inception-v4, inception-resnet and the impact of residual connections on learning. In: *AAAI*. [S.l.: s.n.], 2017. p. 4278–4284. Citado na página 22.
- SZEGEDY, C. et al. Going deeper with convolutions. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015. Disponível em: <<https://doi.org/10.1109/cvpr.2015.7298594>>. Citado 4 vezes nas páginas 38, 41, 43, and 44.
- SZEGEDY, C. et al. Rethinking the inception architecture for computer vision. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016. Disponível em: <<https://doi.org/10.1109/cvpr.2016.308>>. Citado 3 vezes nas páginas 11, 22, and 40.
- THEODORIDIS, S.; KOUTROUMBAS, K. *Pattern Recognition (Fourth Edition)*. USA: Academic Press, 2008. Citado 3 vezes nas páginas 22, 25, and 26.
- VAPNIK, V. N. *Statistical Learning Theory*. Nova Jersey, EUA: John Wiley & Sons, 1998. Citado na página 22.
- WEI, G. et al. Content-based image retrieval for lung nodule classification using texture features and learned distance metric. *Journal of Medical Systems*, v. 42, n. 1, p. 13, Nov 2017. Citado na página 21.
- WEI, G. et al. Lung nodule classification using local kernel regression models with out-of-sample extension. *Biomedical Signal Processing and Control*, Elsevier, v. 40, p. 1–9, 2018. Citado na página 21.
- WU, S.; ZHONG, S.; LIU, Y. Deep residual learning for image steganalysis. *Multimedia Tools and Applications*, Springer Nature, v. 77, n. 9, p. 10437–10453, feb 2017. Disponível em: <<https://doi.org/10.1007/s11042-017-4440-4>>. Citado 3 vezes nas páginas 11, 40, and 41.
- ZEILER, M. D.; FERGUS, R. Visualizing and understanding convolutional networks. In: *Computer Vision – ECCV 2014*. Springer International Publishing, 2014. p. 818–833. Disponível em: <https://doi.org/10.1007/978-3-319-10590-1_53>. Citado na página 37.
- ZHU, W. et al. Deeplung: 3d deep convolutional nets for automated pulmonary nodule detection and classification. *arXiv preprint arXiv:1709.05538*, 2017. Citado 5 vezes nas páginas 22, 51, 66, 67, and 69.
- ZOPH, B.; LE, Q. V. Neural architecture search with reinforcement learning. *CoRR*, abs/1611.01578, 2016. Citado 2 vezes nas páginas 11 and 45.
- ZOPH, B. et al. Learning transferable architectures for scalable image recognition. *arXiv preprint arXiv:1707.07012*, 2017. Citado na página 22.

Appendix

APPENDIX A – The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI)

Public available databases with a consistent quantitative analysis of several Computer Tomography (CT) scans are a valuable resource, not only to oncologic and radiologic research, but also to development of Computer-Aided Diagnosis (CAD) methods. Partly because a common collection of medical images removes the source unreliability of database composition process, which usually prohibits an appropriate comparison among different approaches. Moreover, such a collection also has the potential to enable artificial intelligence methods to compute statistical correlation between annotations and images. However, to collect and select clinical images and their assessments is an expensive and time consuming task (ARMATO et al., 2011).

To overcome this issue, since 2001, the Lung Image Database Consortium (LIDC) (ARMATO et al., 2011), which is formed by five institutions (University of Michigan, University of California, Weill Cornell Medical College, University of Iowa, University of Chicago and Los Angeles), is developing a public web-accessible database with three main components: a repository of chest CT scans; nodules information based on assessments experienced radiologists; and their respective acquisition technical parameters, for example, reconstruction algorithm and slice thickness, as well as patients information such as: gender, age, and final diagnosis.

Further, in 2004, to extend LIDC efforts the National Institutes of Health (FNIH) created the Image Database Resource Initiative (IDRI) that joined five previous LIDC institutions with other eight companies, such as: Fuji, GE, Philips and Siemens; and another two academic groups at Memorial Sloan-Kettering Cancer Center and MD Anderson Cancer Center. Thus, as a result of this cooperation process, that repeated the same LIDC procedures during the collection and annotation of new IDRI scans, it was possible to combine both chest CT scans database composing what is referred as the LIDC-IDRI database. The annotation process used in the LIDC-IDRI database consisted of four phases: “blinded read”, unblinded read, assessment phase and filtering phase.

In the *“blinded read phase”*, four experienced radiologists were instructed to review the same scan without exchanging any information about it. As a review can be a subjective task, it was also required that they mark lesions and classify each of them into three possible categories: nodule $\geq 3mm$, nodule $< 3mm$ and non-nodule $\geq 3mm$. It is

noteworthy that, in this annotation process the ‘nodule’ label could be interpreted as a noncancerous process, a primary lung cancer, a metastatic, or an indeterminate stage. Moreover, for each “nodule $\geq 3mm$ ” the radiologist used a software to specify the outer boundary of the nodule.

In the unblinded read phase, all radiologists had access to all other colleagues (anonymized) annotations, then, without forcing any consensus, the radiologists independently reviewed their own marks.

In the assessment phase, each radiologist had to independently specify characteristics (e.g., likelihood of malignancy) for all nodules that he/she considered to be bigger or equal to $3mm$.

In the filtering phase, the process to ensure the annotations integrity was based on the detection and elimination of five categories of potential error: annotations outside of pulmonary regions; errant marks inside the lungs; multiple annotations to a lesion by the same radiologist; nodule outer boundaries that are disconnected over adjacent CT slices; nodules $\geq 3mm$ without any outer boundaries mark; evident disparity between the lesion size and the nodule classification (as bigger or smaller than $3mm$).

Thus, after the filtering stage, from all 1010 CT scans in the LIDC/IDRI Database, 7371 lesions were detected and assessed as nodule by at least one of the four radiologists and 2669 of these were classified as a "nodule $\geq 3mm$ " by at least one radiologist. Further, from these 2669 nodules, 777 (29.1%) were classified as a "nodule $\geq 3mm$ " by one radiologist, whereas 928 (34.7%) were classified for all four of them.

Finally, all exams and annotations, that compose LIDC/IDRI Database, are publicly and freely available for download at the National Biomedical Image Archive (NBIA) website (<http://ncia.nci.nih.gov/>).