

UNIVERSITÀ DELLA CALABRIA



DOTTORATO DI RICERCA IN INFORMATION
AND COMMUNICATION TECHNOLOGIES
XXXII CICLO

**Classification of medical images:
instance space optimization models for
Multiple Instance Learning.**

Author:
Eugenio VOCATURO

Supervisors:
Prof. Antonio FUDULI
Prof. Manlio GAUDIOSO

*A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy*

DIMES
Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e
Sistemistica

List of Figures

1.1	Fundamental approaches of Machine Learning	9
1.2	Relations between diverse areas of Data Mining.	11
1.3	Supervised learning: main modules and data flows	14
1.4	Classification and Regression	16
1.5	Unsupervised learning: main modules and data flows	17
1.6	Clustering example	18
1.7	Reinforcement learning: main modules and data flows.	20
1.8	Major Supervised and Unsupervised learning methods.	21
1.9	Steps of an Image Classification System	23
1.10	SVM model for two-dimensional feature vectors	26
1.11	Linearly non-separable cases: a)presence of noise, b)inherent nonlinearity.	26
1.12	Data linearly separable after a transformation into a three-dimensional space.	28
1.13	Mapping between the elements of the biological neuron and artificial neuron	29
1.14	Artificial neuron: inputs, activation function, weighted connections, calculation function	30
1.15	Types of activation function	31
1.16	An example of Neural Networks	31
1.17	Hierarchical clustering	38
1.18	K-means clustering approach	40
2.1	MIL approach for binary classification	54
2.2	Classification of images into beach (top row) and non-beach (bottom row) [94]	58
2.3	Procedure of MIL classification problem	62
3.1	Spherical separation with three negative bags and two positive bags	82
4.1	Incidence rates per sex, top 10 cancers [1]	91
4.2	Incidence and mortality rates, top 10 cancers [1]	92
4.3	Statistics on Melanoma in 2018 [1]	93
4.4	Melanoma age standardized incidents rates per sex [1]	94
4.5	Detailed melanoma incidence per sex and by region [1]	95
4.6	Detailed melanoma mortality rates per sex and by region[1]	95
4.7	Ten Leading Cancer Types for the Estimated New Cancer Cases by Sex, in United States in 2019 [164]	96
4.8	Trends in Incidence Rates for Selected Cancers by Sex, United States, 1975 to 2015 [164]	96
4.9	Melanoma age standardized incidents and mortality rates per sex	96
4.10	Skin Layers	97

4.11	Types of skin Lesion	98
4.12	Main steps in biomedical image processing	99
4.13	A collection of images from PH^2 database: common nevi (1st row), dysplastic nevi (2nd row) and melanomas (3rd row)	102
4.14	Image pre-processing for automated skin detection	103
4.15	The color features of the CIE standard	105
4.16	Visual Noise Effect	107
4.17	An image of nevus: (a) with hair, (b) after pre-processing step	110
4.18	(a) original image (b) segmented image	111
4.19	Original and Weighted 7-Point Check List	116
4.20	Example of CNN architecture	121
4.21	Kernel trick ϕ : from a) Input Space to b) Feature Space	123
5.1	Summary of the classification performances of the methods reviewed from the dermoscopic imaging literature [2]	126
5.2	Images with yellow	133
5.3	Images without yellow	133
5.4	Sequence of extracted blobs	134
5.5	Extracted blobs from subfigure 1.39 of Figure 5.2	134
5.6	Melanomas and common nevi images	136
5.7	Selected images of melanomas and common nevi for pre-processing step	138
5.8	Plain photos of melanoma	141
5.9	Plain photos of common nevi	142
5.10	Left - dysplastic nevus; Right – cutaneous melanoma.	144
5.11	Dermoscopic images of Melanomas (a), Dysplastic nevi (b) and Common nevi (c)	147
5.12	Comparison of obtained performances with the literature results	151

List of Tables

3.1	Characteristics of Medium Size Data sets	87
3.2	Characteristics of Large Size Data sets	88
3.3	Computational results: Average test-correctness on Medium Size Literature Data sets (%)	89
3.4	Computational results: Average test-correctness on Large Size Literature Data sets (%)	89
3.5	Computational results: Average cpu-time (s) and correctness (%) of DC-SMIL training-phase: medium-size problems	89
3.6	Computational results: Average cpu-time (s) and correctness (%) of DC-SMIL training-phase: large-size problems	89
4.1	Advantages of most prominent current imaging techniques in Dermatology .	100
4.2	Limitations of most prominent current imaging techniques in Dermatology .	101
4.3	Mean Filter	108
4.4	Statistic Filter	109
4.5	Adaptive Filter	109
5.1	Results of the first experiment for color image classification	134
5.2	Data set constituted by 40 melanomas and 40 common nevi: average testing values	137
5.3	Data set constituted by 40 melanomas and 80 common nevi: average testing values	137
5.4	Data set constituted by 80 melanomas and 80 common nevi: average testing values	137
5.5	Test on pre-processed Melanoma DB: 10-fold cross-validation	139
5.6	Test on pre-processed Melanoma DB: 5-fold cross-validation	139
5.7	5-fold cross-validation	140
5.8	10-fold cross-validation	143
5.9	Leave-One-Out validation	143
5.10	Data set constituted by 80 melanomas and 80 dysplastic nevi: average testing values	146
5.11	Data set constituted by 80 dysplastic nevi and 80 common nevi: average testing values	146
5.12	Data set constituted by 80 melanomas against 80 dysplastic nevi and 80 common nevi: average testing values	146
5.13	Data set constituted by 80 dysplastic nevi and 80 common nevi: 5-CV average testing values	149

5.14 Data set constituted by 80 dysplastic nevi and 80 common nevi: 10-CV average testing values 149

Contents

List of Figures	iii
List of Tables	v
Contents	vii
Preface	1
Thesis outline	3
I The Basics	5
1 Image Classification Techniques	7
1.1 Machine Learning	7
1.1.1 How Machine Learning Works	8
1.1.2 From Machine Learning to Data Mining: the boundaries between re- search sectors	11
1.1.3 Future evolution of Machine Learning	12
1.2 Classification and Machine Learning	12
1.2.1 Learning	13
1.2.2 Supervised Learning	13
Classification and Regression	15
1.2.3 Unsupervised Learning	17
Clustering	18
1.2.4 Reinforcement Learning	19
1.2.5 Semi-supervised Learning	20
1.3 Image Processing	22
1.4 Pattern Recognition and Image Analysis	23
1.5 Types of classifiers for image analysis	24
1.5.1 Support Vector Machine	24
Dealing with Noise	27
Dealing with Inherent Non-Linearity	27
1.5.2 Neural Networks	29
Training of Artificial Neural Networks	31
Gradient Descent	33
Generalization, Overfitting and Underfitting	33
1.5.3 Nearest Neighbor	34
1.5.4 Clustering	35
Centroid-based clustering	36

	Distribution-based clustering	37
	Density-based clustering	37
	K-means algorithm	38
	Expectation-Maximization	40
1.5.5	Decision Trees	42
1.5.6	Multi-classifiers	45
	Merger at decision level	45
	Confidence level fusion	46
1.6	Applications of classifiers in image analysis	46
1.6.1	Classification at the pixel level	46
1.6.2	Segmentation	47
1.6.3	Object Recognition	48
	Pose Consistency	48
	Template Matching	49
	Relational Matching	49
1.7	Take away	50
2	Multiple Instance Learning Problems: models and algorithms	51
2.1	Introduction	52
2.2	Multiple Instance Learning Assumptions	53
2.3	Characteristics of MIL problems	55
2.3.1	Prediction: instance-level vs. bag-level	55
2.3.2	Bag composition	56
	Witness rate	56
	Relations between instances	57
2.3.3	Data distributions	59
	Multimodal distributions of positive instances	59
	Non-representative negative distribution	60
2.3.4	Label Ambiguity	61
	Label noise	61
	Label spaces	61
2.4	MIL Paradigms	62
2.4.1	Taxonomy of MIL paradigms	62
2.5	Instance Space Models	64
2.5.1	Support Vector Machines for Multiple Instance Learning	64
	The <i>mi-SVM</i> model	64
	The <i>MI-SVM</i> model	67
2.5.2	The Mangasarian and Wild Model	68
2.6	An overview of MIL literature	69
2.7	Take away	74
II	The Advances	75
3	Classification via spherical separation	77
3.1	Spherical models for classification problems	78
3.2	Multiple instance classification via spherical separation	79

3.2.1	Problem statement	80
3.2.2	A DC decomposition of SMIL	82
3.2.3	Solving the DC-SMIL model	84
3.2.4	Data sets	85
3.2.5	Numerical results and final remarks	87
4	Machine Learning and Automated Melanoma Detection	91
4.1	Statistics on Melanoma	91
4.2	Skin layers	93
4.3	Computer Aided Diagnosis Systems	98
4.3.1	Image Acquisition methods	99
4.3.2	Image Pre-processing	102
	Image Enhancement	103
	Image Restoration	106
	Hair Removal	110
4.3.3	Image Segmentation	111
4.3.4	Features Selection	112
	ABCD Rule	112
	Seven Point CheckList	116
	Texture Analysis	116
4.3.5	Classification	118
	K-Nearest Neighbour Algorithm.	118
	Decision Trees	119
	Logistic Regression	119
	Artificial Neural Network	120
	Support Vector Machines	122
	Multiple Instance Learning techniques	123
5	MIL Models application in Automated Melanoma Detection	125
5.1	Classification performances of ML methods on dermoscopic images	125
5.2	An overview of numerical experiments	127
5.3	The MIL-RL algorithm	128
5.3.1	The model	128
5.3.2	The algorithm	129
5.3.3	Numerical experiments on image classification task	131
5.3.4	Classification tasks involving dysplastic nevi	143
	Melanomas vs Dysplastic Nevi	146
	Dysplastic Nevi vs Common Nevi	148
	Melanomas vs (Dysplastic Nevi and Common Nevi)	148
5.4	DC-SMIL for automated Melanoma Detection	149
5.5	Results obtained with MIL approach	150
5.6	Discussion	150
5.7	Future Work	153
5.7.1	The market and competition	154
5.7.2	Market value	154
	Appendix	155

Preface

We are living in an age characterized by the extraordinary proliferation of digital data.

This global phenomenon is due to the massive diffusion of devices, also wearable, producing data in the form of text, audio, images, videos, or metadata coming from their composition. As more and more data are generated, the human ability to understand and process information is clearly inadequate due to their volume and complexity. The abundance of data brings the need of providing mathematical programming-optimization type, according to the celebrated Leonhard Euler's statement:

"Nothing happens in the universe that cannot be traced back to a problem of maximum or minimum."

Recently, there has been considerable research efforts in various disciplines, ranging from psychology to artificial intelligence, aiming at developing general models for knowledge representation. Software engineering and optimization of mathematical models gave rise to new artificial intelligence approaches that are applied in many fields; this effort produced machine learning paradigms enabling the realization of interesting techniques for data analysis.

In particular, in health care research areas the spread of digital imaging techniques and technologies leads us into a world in which the observation of phenomena and the related knowledge extraction are paradigms that must be constantly updated.

One of the most interesting issues in the medical field concerns the classification of biomedical data and images, with the aim of supporting diagnostic processes. There are many areas of scientific research, such as statistics and databases theory, machine learning, pattern recognition, artificial intelligence and computer vision which deal with classification problems, proposing for them various solving methodologies.

Healthcare is one of the most important contexts that is taking advantage from machine learning approaches. Human longevity is intimately connected to the development of new effective diagnosis techniques and treatment for specific diseases. The possibility of exploiting machine learning approaches together with the availability of digital data opens up the opportunity to support both diagnostics and follow-up of aggressive diseases.

Various kind of classifiers, such as support vector machines, decision trees and neural networks, have been used in the development of decisions support systems providing a second opinion for the diagnosis of certain pathologies. These solutions turn out to be even more important for those diseases for which a rapid intervention at the onset of the disease proves to be decisive.

The objective of the thesis is to apply a peculiar classification technique known as Multiple Instance Learning (MIL) for the automatic classification of medical images. In particular, we focus on the domain of skin cancers, an area where to the best of our knowledge, no MIL

approaches have been used for the classification of melanomas against dysplastic nevi and common ones. In particular we introduce a variant of the MIL approach where classification is based on the use of spherical separation surfaces.

Referring to the statistics of the World Health Organization [1], it emerges that melanoma is a skin lesion in the process of spreading and registering more than 60.000 deaths a year with over 280.000 new cases diagnosed per year. Considering the open debate on Atypical Mole Syndrome (AMS), according to which the simultaneous presence of a high number of common nevi with a certain number of dysplastic nevi implies a greater risk of the onset of cutaneous melanoma, new classification challenges become important. This is not only for the classification of melanomas against dysplastic nevi, a task to which, up to now, little interest has been reserved in the literature [2], but also for the new task related to the classification of dysplastic nevi against common ones for a correct evaluation of AMS.

These last two challenges are very difficult due to the similarity of the lesions that we intend to discriminate. The possibility of being able to favor both the diagnosis of specialists and the self-diagnosis of the person passes through the definition of efficient classification algorithms. In total agreement with Descartes, mathematics has become a tool to support the diagnosis of melanoma.

"I am convinced that mathematics is the most important instrument of knowledge among those left to us by human action, being the source of all things."

A handwritten signature in blue ink, reading "Eugenio Volpato". The signature is written in a cursive style with a large, stylized initial 'E'.

Thesis outline

In the thesis work we focused on the study of Multiple Instance Learning (MIL) techniques applied to the binary classification of medical images. Unlike the standard classification, which consists in discriminating some points by assigning each of them to a class, a MIL problem consists in classifying different sets of points: these sets are called *bags* and the points inside them are named *instances*. In particular, differently from the classical supervised classification, in the learning phase of a MIL problem only the labels of the bags are known, whereas the labels of the instances inside them remain unknown. Problems of this type fit very well for image classification, where the images are represented by the bags and the sub-regions inside them correspond to the instances. In the medical field, for example, the image of a CT scan identifies a pathology on the basis not of the entire image, but on the basis of some portions of it. In the binary case, where the aim is to discriminate between positive and negative bags, the Multiple Instance Learning problems are based on the following standard assumption (used when also the instances can belong to only two different classes): a bag is positive if at least one of its instances is positive and, vice-versa, it is negative if all its instances are negative. In particular, for this kind of problems, we focused our attention on the instance-space methodologies, where the classification process takes place by classifying the individual instances, from which it is possible to successively compute the class label of the corresponding bags. This thesis is organized in two parts.

Part I, *The Basics*, provides an introduction to machine learning methods for image classification and Multiple Instance Learning, an emerging approach that fits very well with the task of image and video classification.

Part II, *The Advances*, presents the MIL-RL algorithm and our proposal of a MIL algorithm (DC-SMIL) that uses spherical surface for image classification tasks. In particular, MIL-RL and DC-SMIL are applied to automated melanoma classification.

The two parts are organized in the following chapters.

In the first chapter we review machine learning techniques for image classification; after we have described different machine learning approaches, i.e. supervised, unsupervised, reinforcement and semi-supervised learning, we have revised the most common types of classifier useful for image analysis.

In the second chapter we deepen the treatment of Multiple Instance Learning, by describing the characteristics of related problems, the taxonomy, the models and the algorithms. In particular, we focus on instance-space models, reporting the formulations of the *mi-SVM*, *MI-SVM* models and of the Mangasarian and Wild model. Finally, a road map of the works that use the MIL approach is reported by highlighting those that are useful for image classification.

In the third chapter we focus on the classification approaches that use spherical separation surfaces. This is motivated by the need for new tools that can perform well when the classes of data to be separated show strong similarities. After introducing some reference

models, we report our original contribution, presenting *DC-SMIL*, a spherical classification algorithm for MIL which has proved to be effective for image classification tasks. A numerical result section reports the classification performances on reference data sets used in the literature.

In the fourth chapter we introduce automatic melanoma classification. After recalling statistics of this particular form of skin cancer, we introduce Computer Aided Diagnosis (CAD) Systems, that are interesting tools for automatic diagnosis of skin lesions. CAD systems foresees fundamental steps such as image acquisition, image pre-processing, segmentation, features extraction and selection and finally classification.

For each of these phases we recall the main fundamental aspects, under the perspective of defining the desirable characteristics that can be useful for the implementation of a specific application. In particular, at the end of the chapter, we focus on automatic diagnosis of melanoma.

In the last chapter we report the numerical results of the application of two selected MIL models in classification of skin lesion images. We focus on *MIL-RL*, the model we have chosen to demonstrate that MIL approaches perform well for the particular chosen tasks; comparisons with other Machine Learning techniques are appropriately reported. We pay attention to some emerging aspects, starting from the current debate concerning the role of dysplastic moles as a factor that increases the risk of incoming melanoma. This aspect raises new challenges related to the classification of melanomas against dysplastic nevi and of the classification of dysplastics nevi against common ones. While the first challenge has been little addressed, the second, to the best of our knowledge, has not been taken into consideration [2]. The reported results for the latter classification task highlight how, although *MIL-RL* performs better than the other considered methods, it provides inadequate performance. Classification performance on literature data sets shows that *DC-SMIL*, even without an exhaustive parameter setting, provides interesting results for image classification. The results on skin lesion classification tasks, are reported in a dedicated section.

In general, the numerical section is obtained by considering two different types of data sets:

- a dermatoscopic data set named *PH²* containing 200 melanocytic lesions images [3]
- a data set of photographs (jpeg) publicly available from online databases [4].

The application on the second data set is justified by the fact that self-diagnosis systems are being studied and this raises the challenges related to algorithms that have to work well on non-dermatoscopic images. The relative experimental section shows how the proposed approaches are better in general than those found in the literature, but also that the phase of pre-processing of images as well as a suitable choice of features are necessary when the images considered are not dermatoscopic and generally low quality.

Finally, a section with a reference to possible future developments of the research themes and a brief discussion is presented.

PART I

THE BASICS

Chapter 1

Image Classification Techniques

"All models are wrong but some are useful"

—George Box, Journal of American Statistical Association (1976)

Discussions on computers, machine learning and artificial intelligence seem, nowadays, entirely smooth. However the road that has taken us here has been very complex and difficult due to the skepticism surrounding such field of research.

The first experiments on machine learning date back to the early fifties of the past century, when some mathematicians started thinking of adopting probabilistic theory to teach a machine how to predict a certain event. The first name linked to machine learning is certainly that of Alan Turing, who devised the possibility of producing machines capable of learning. In the same years, the studies on artificial intelligence, expert systems and neural networks experienced periods of impressive growth as well as periods of abandonment, because of the difficulties inherent in the realization of intelligent systems, together with the absence of economic subsidies in a field really surrounded by a skeptical atmosphere. Starting in the 1980s, a number of experiments led to the revival of this field of research.

The renaissance was made possible by new investments in the sector. In the late nineties machine learning found a second life due to a series of innovative techniques that allowed *machine learning* to become a branch of research in high demand.

1.1 Machine Learning

The term "Machine Learning (ML)" refers to techniques and approaches used for automatic detection of relevant patterns from data collections. The growing availability of digital data makes the ML approaches widely used for information extraction. We are surrounded by machine learning based technology: search engines learn how to deliver results in the most efficient way, anti-spam software learns to filter our email messages, and credit card transactions are secured by software solutions that learn how to detect frauds. Smartphones are now equipped with advanced digital cameras through which they are able to detect faces interacting with voice commands. In sectors such as bioinformatics and medicine, ML approaches are increasingly adopted to address specific challenges. The use of computers becomes crucial due to the complexity of the models that must be detected: it is not always possible for humans to provide an explicit specification of the tasks that must be performed, thus machines have to learn by themselves.

The growing digitalization of our world and the following proliferation of data allow the proposal of algorithms for large-scale machine learning (Big Data), giving rise to a wide spectrum of different learning techniques. Machine Learning aims at teaching computers and robots to perform actions and activities in a natural way like humans: learning from experience.

Summing up, machine learning algorithms exploit mathematical - computational methods to obtain learning information directly from data. Machine Learning algorithms may improve their performance in an "adaptive" way, as the examples with which they work increase without having been explicitly programmed. Arthur Lee Samuel, a pioneer scientist in the field of Artificial Intelligence, in 1959 was the first to coin the term "Machine Learning", although the most accredited definition by the scientific community is that provided by Tom Michael Mitchell, director of the Machine Learning department of Carnegie Mellon University:

«... it is said that a program learns from experience E with reference to some classes of tasks T and with measurement of performance P , if its performance in task T , as measured by P , improves with experience E ». [5].

Machine Learning allows computers to learn from experience; there exists "learning" whenever the performance of the program improves after the performance of a task or the completion of a possibly wrong action. Instead of writing the programming code through which, step by step, the machine is "told" what to do, the computer is only provided with data sets inserted in a generic algorithm that develops its own logic to perform the function, the activity, the task required. The evolution of the concept of "intelligence" in "artificial intelligence" follows:

Intelligence: Complex of psychic and mental faculties that allow humans to think, understand or explain facts or actions, elaborate abstract models of reality, understand and to be understood by others, judge, and to render possible adaptation to new situations and to change the situation itself when it presents obstacles to adaptation [6].

Artificial intelligence: Partial reproduction of the intellectual activity proper to man (with particular regard to the processes of learning, recognition, choice) realized through the elaboration of ideal models, or with the development of machines that mostly use electronic computers for this purpose [7].

1.1.1 How Machine Learning Works

In principle, machine learning works on the basis of two distinct approaches, which were originally identified by Arthur Samuel at the end of the 1950s. These approaches make possible to differentiate machine learning in two general sub-categories depending on whether the computer is given examples on how to perform the required task (supervised learning) or let the software work without any "help" (unsupervised learning).

Indeed, a more rich taxonomy is available which allows us to make a further and even more detailed classification of the Machine Learning techniques based on its *modus operandi*. All these techniques are used to classify data.

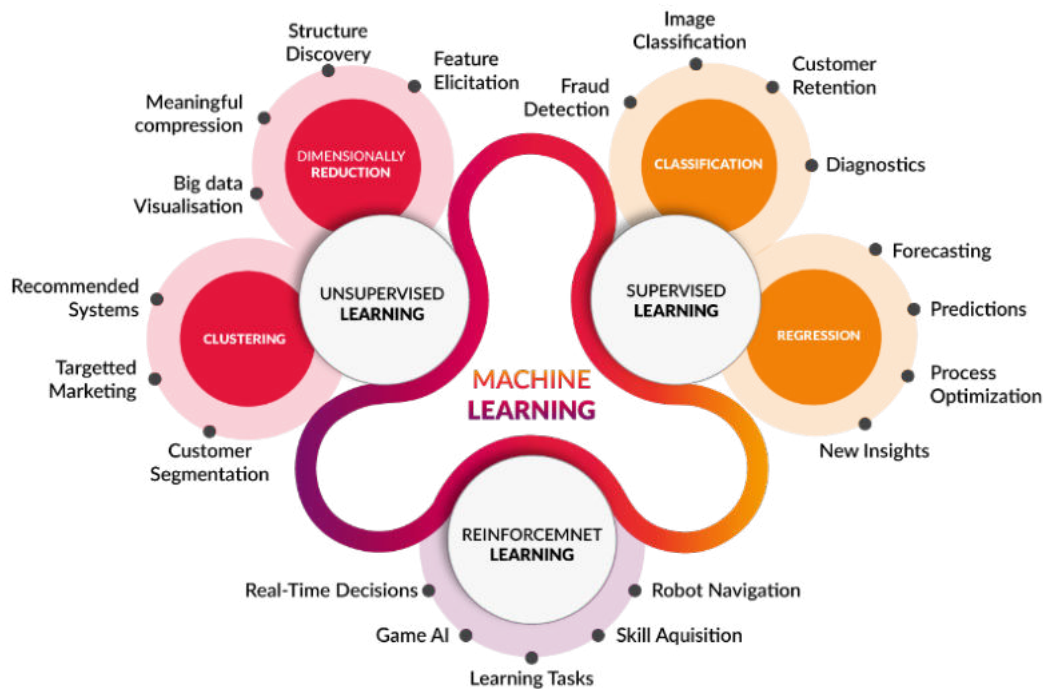


FIGURE 1.1: Fundamental approaches of Machine Learning

A- *Machine Learning with supervised learning.*

In this area of machine learning, both the data sets used as inputs and the information on the desired results “feed” the system with the aim of identifying general rules that link incoming data with outgoing data. The so identified rules can then be reused for other similar tasks.

“In supervised learning, resolution work is left at the computer. Once you understand the mathematical function that led to solving a specific set of problems, it will be possible to reuse the function to respond to any other similar problem” writes Adam Geitgey in his article “Machine Learning is Fun!” [8].

Interesting examples of Machine Learning with supervised learning come from scientific research in medical field where algorithms learn to make increasingly accurate predictions to prevent outbreaks of epidemics or to accurately and promptly diagnose tumors or rare diseases. And again, in the context of supervised learning, there are interesting Machine Learning applications on the level of speech recognition or identification of manual writing. Typical applications of Supervised Learning include Classification and Regression task.

B- *Machine Learning with Unsupervised Learning.*

In this second Machine Learning area, only data sets are supplied to the system without any indication of the desired result. The purpose of this second learning method is to “go back” to hidden patterns and models, i.e. to identify the logical structures which

have not been previously labeled. Two of the main methods used in unsupervised learning are "principal component" and "cluster analysis".

Principal components analysis (PCA), also known as Karhunen-Loève transform, is a technique for simplifying the data used in the context of multivariate statistics [9]. This method was first proposed in 1901 by Karl Pearson and later developed by Harold Hotelling in 1933, and is part of the factorial analysis. The purpose of the technique is to reduce the more or less large number of variables that describe a set of data to a smaller number of latent variables, limiting the loss of information as much as possible.

Cluster analysis is used in unsupervised learning approaches to aggregate classes of data with common features highlighting the relationships between them. Cluster analysis is a branch of machine learning that groups the data that have not been labelled, classified or categorized. Instead of responding to feedback, cluster analysis identifies commonalities in the data and reacts according to the presence or absence of such commonalities in each new piece of data. This approach is useful for anomalies detection of data that do not belong to any group. Therefore, clusters are determined using density estimation based on similarity.

C- *Machine Learning with reinforcement learning.*

In this case, the system must interact with a dynamic environment and achieve a goal also by learning from errors. The behavior of the system is determined by a learning routine based on reward and punishment. With such a model, the computer learns, for example, to beat an opponent in a game by concentrating its efforts on performing a certain task, aiming to reach the maximum value of the reward; in other words, the system learns by playing and by the mistakes made improving performance precisely in relation to the results previously achieved.

Systems based on reinforcement learning are the foundation for the development of self-driving cars which, through Machine Learning, learn to recognize the surrounding environment (with data collected by sensors, GPS, etc.) and to adapt their "behavior" to specific situations they encounter.

D- *Machine Learning with semi-supervised learning.*

In this area, the computer is supplied, through "hybrid" model, with a set of incomplete training / learning data ; some of these inputs are "endowed" with examples of output (as in supervised learning), others lack them (as in unsupervised learning). The basic objective is always the same: to identify rules and functions for solving problems, as well as models and data structures useful for achieving certain objectives.

E- *Other practical approaches to Machine Learning: from probabilistic models to Deep Learning.*

There are other ways for classifying Machine Learning approaches which suggest the adoption of sub-categories functional to a "practical" classification of Machine Learning algorithms.

We speak for example of the so-called *graph-based decision trees*. In machine learning a decision tree is a predictive model, where each internal node represents a variable, an arc towards a child node represents a possible value for a certain property and a leaf

node represents the predicted value for the target variable. A decision tree is a graph representing possible decisions and their implications, used to create actions aimed at a specific purpose.

Another concrete example comes from "clustering" or from "mathematical models" that allow the grouping together of "similar" data. This practical approach of machine learning can be effectively implemented through different learning models ranging from identification of structures, to objects recognition that must be part of one group rather than another.

Then there is the sub-category of "probabilistic models" where the system's learning process is based on the calculation of probabilities; the best known is the "Bayes network", a probabilistic model that represents the set of random variables in a graph and its conditional dependencies.

Finally, we mention the artificial neural networks (ANN) that use algorithms inspired by the structure, functioning and connections of biological neural networks (i.e.those of the human being) for learning, [10]. More advanced models, such as the so-called multilayer neural networks, lead to Deep Learning.

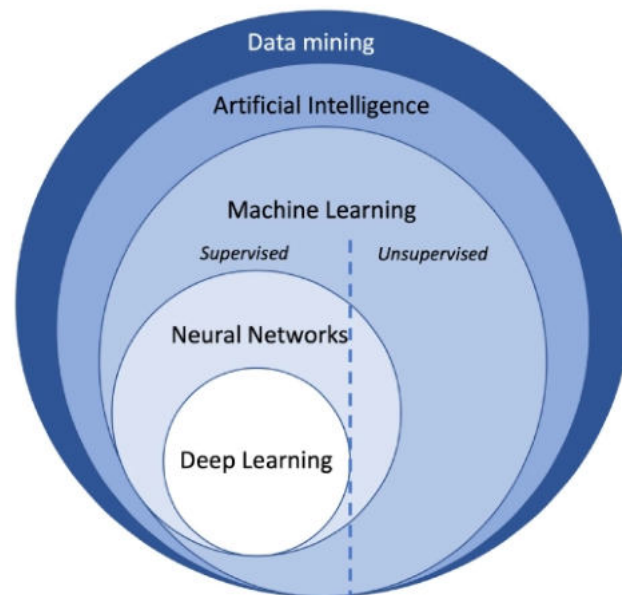


FIGURE 1.2: Relations between diverse areas of Data Mining

1.1.2 From Machine Learning to Data Mining: the boundaries between research sectors

One of the main characteristics of machine learning is its close correlation with other branches of computer science, statistics, optimization and many other areas of modern science. Incursions in different fields and sectors, in fact, are very common and of fundamental importance in order to create structures that allow a machine to learn using different approaches.

What is Data Mining?

Data Mining is the process of extracting implicit, previously unknown and potentially useful information from data [11]. When considering large data sets, the main method used for computational detection of patterns is data mining. The boundaries of data mining are not clearly delineated and intersect those of machine learning, artificial intelligence, statistics and database systems. There are many directions in which Data Mining is applied: from data pre-processing to the management of recurrent patterns. On the other hand, machine learning is a part of computer science and it is very similar to Data Mining. Machine Learning is also used to search models and to explore the construction and study of algorithms. Machine Learning is an artificial intelligence paradigm aiming to develop software solutions that allow machines to adapt themselves to new situations [12]. It also has strong links with mathematical optimization. Machine Learning sometimes conflicts with Data Mining because both are like two faces on a dice. Machine Learning has a more proactive approach respect to Data Mining: while Data Mining is used to extract rules from available data, ML is used to teach the computer the way to learn the rules of interest [12].

More generally, even if the use of techniques can be similar, what differentiates the branches related to machine learning, artificial intelligence, Data Mining and other intelligent systems, is the purpose for which these systems were created.

Similarly, overlapping of methodologies and results occurs in other branches of machine learning research. Among these, for example, there is optimization, that is the improvement of the efficiency of the system that allows to obtain results in a more rapid and less dispersed way. Also in this case, the boundary between the two sectors are often weak, and are above all defined in terms of the specific objectives which are pursued.

1.1.3 Future evolution of Machine Learning

While in recent years research has made great strides on forms of intelligent learning, still much needs to be done to optimize approaches, algorithms and techniques. The possibilities of future development of this branch are many, all linked to different fields of application. Although home automation has already made use of some of the simplest machine learning systems, it must be acknowledged that many other sectors can take advantage of the use of machines capable of making intelligent choices. Probably, the only factor limiting the full use of tools that can learn by themselves is man's fear that machines can become so intelligent, to take over a part of his freedom.

This is a fear that, as stated by Professor Pedro Domingos of the University of Washington, an expert in machine learning and Data Mining, is not well founded: *"people are afraid that computers will become too intelligent and dominate the world, but the real problem is that being still too stupid they have already conquered it"* [13].

1.2 Classification and Machine Learning

Before considering pattern recognition and image analysis themes, it is appropriate to recall some basic concepts of machine learning. The definition of learning for computational models will be functional for the three fundamental approaches of machine learning: supervised,

unsupervised and reinforcement. In the following sections the concepts of classification, regression and clustering will be introduced also referring to the most used Machine Learning algorithms.

1.2.1 Learning

Machine Learning tasks are often described in terms of how the system deals with a collection of features evaluated on the referred data. Typically, the input is represented by a vector $x \in \mathbb{R}^n$ where each x_i represents a descriptive characteristic (feature) of the input data. To evaluate the abilities of a machine learning algorithm, a quantitative measure P must be estimated indicating its performance. Often the measure of P is specific to a certain task T which the system must perform. For tasks such as *classification*, P is evaluated by measuring the accuracy of the model, evaluating percentage of examples for which the model provides a correct output. Another reference parameter may be the *error rate*, defined instead as the proportion of examples for which the system provides a wrong output.

It is important to check how the algorithms are able to evaluate unseen data, enhancing performance indices on *testing set*, and on *training sets*, in order to evaluate the performance on independent assessments. In Appendix A we will recall some indexes universally adopted to evaluate classification performances.

Machine Learning approaches are typically classified depending on the nature of the "signal" used for learning or the "feedback" available to the learning system. As mentioned earlier, in literature, we find three general categories of machine learning: supervised, unsupervised and reinforced learning. Halfway between supervised and unsupervised learning is the *semi-supervised learning* in which the teacher provides an incomplete training data set, that is, a set of training data among which there are data without the respective desired output [14]. Another categorization of machine learning approaches considers the output of the machine learning system [15]:

- In *classification*, the outputs are divided into two or more classes and the learning system must produce a model that assigns the inputs not yet seen to one or more of the classes. This is usually dealt with in a supervised manner. Anti-spam filtering is an example of classification, where the inputs are emails and the classes are "spam" and "not spam".
- In *regression*, which is a problem usually solved using supervised approach, the output and model used are continuous. An example is the prediction of the value of the exchange rate of a currency in the future, given its values in recent times.
- In *clustering* an input set is divided into groups. Unlike the case of classification, groups are not known before, typically making it an unsupervised task.

1.2.2 Supervised Learning

In supervised learning, data are provided with relative input features and output value. As previously mentioned, in the event that the output value is discrete, such as belonging or not belonging to a specific class, the problem is set up as a classification problem. If, on the other hand, the output is a continuous real value in a given range then we will have a Regression Problem. In both cases we want to find the function, called *hypothesis* h , which

given an unknown input, estimates the value of the output. The goal of supervised learning problems is to make a prediction based on known properties learned from the input data.

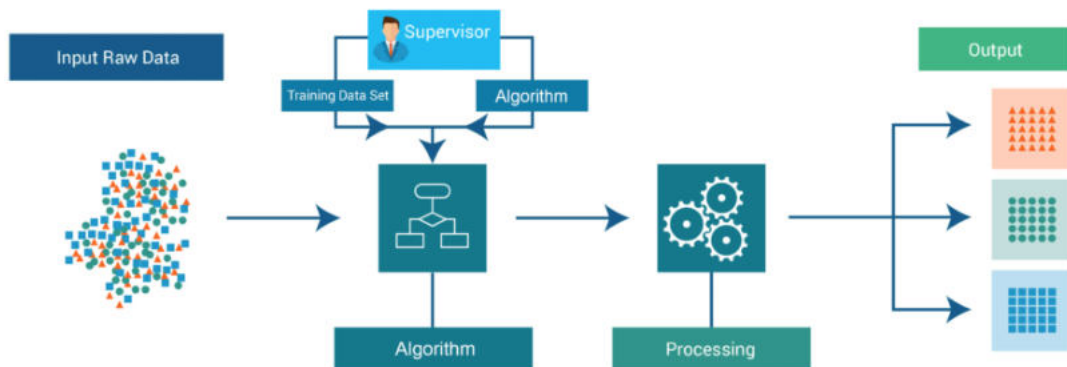


FIGURE 1.3: Supervised learning: main modules and data flows

The input data are called training sets and are composed of a number n of examples for each of which some features are evaluated. The output value is called target value. To identify each example and its output, we will use the notation $x^{(i)}$ to indicate the i -th example and $y^{(i)}$ to indicate the i -th output value. If there are multiple features we will use the notation $x_j^{(i)}$ to indicate the j -th feature of the i -th example. The label $y^{(i)}$ can be either an element belonging to a finite set of classes $\{1, 2, \dots, C\}$ or a real number, or a more complex structure, like a vector, a matrix, a tree, or a graph.

The goal of a supervised learning algorithm is to use the data set to produce a model that takes a feature vector x as input and outputs information that permits deduction of the label for this feature vector. For instance, the model created using the data set of people could take as input a feature vector describing a person and output a decision about the fact that the person is affected or not by a given pathology. The most widely used learning algorithms are:

- Support Vector Machines
- linear regression
- logistic regression
- naive Bayes
- linear discriminant analysis
- decision trees
- k-nearest neighbor algorithm
- Neural Networks (Multilayer perceptron)
- Similarity learning

One of the crucial aspects in the prototyping phase of a software solution to a real problem concerns the choice of the mathematical model and the algorithms to be used.

The choice of a learning algorithm for a specific application involves the evaluation of many factors.

Heterogeneity of the data

A first aspect to consider is the nature of features. In the presence of discrete, rather than ordered, or continuous values, the use of some algorithms is more appropriate than others. For examples Support Vector Machines, linear regression, logistic regression, neural networks, and nearest neighbor methods, require that the input features be numerical and scaled to similar ranges. Decision trees easily handle heterogeneous data and are preferred to nearest neighbors' methods and support vector machines with Gaussian kernels. Many of the methods that adopt a distance function are sensitive to data heterogeneity.

Redundancy in the data

Redundant information in input features, cause numerical instability in some learning algorithms, such as linear regression, logistic regression and distance-based methods. These problems can be overcome by adopting some form of regularization. Another way is to adopt data pre-processing steps in order to select a number of relevant features which are not related to each other. These steps are clearly dependent on the nature of the problem and the type of available data.

Presence of interactions and non-linearities.

If every feature contributes to the output in an independent way, the algorithms based on linear functions, among which we remember the linear regression, the logistic regression and support vector machines with Gaussian kernels, normally return satisfactory performances. Decision trees and neural networks are specifically indicated in presence of data whose features have strong interactions. One possible way, when we are dealing with a specific problem, is to compare the behavior of multiple learning algorithms aiming to identify which works better. Next phase involves the algorithm performance optimization: this phase can take a long time and it is often preferable to devote time to collect new data and to identify new useful features with respect to the setting of the chosen algorithm.

Classification and Regression

Classification is one of the main purposes of Machine Learning, and is related to the problem of identifying the class (or label) defined *a priori*, of a new object on the basis of knowledge extracted from a training set of data. In a classification problem, a label is a member of a finite set of classes. If the size of the set of classes is two ("spam"/"not-spam"), we talk about binary classification. In machine learning, the classification problem is solved by a classification learning algorithm, called *classifier*, that takes a collection of labeled examples as inputs and produces a model that can take an unlabeled example as input and either directly outputs a label or outputs a number that can be used by the data analyst to deduce the label easily.

The construction of the classifier takes place through a training phase on a training set, allowing to learn the distribution of features as a function of a known class. Thus, a learning algorithm L , is used to determine the model that best identifies the relationship between the attributes of data and the various classes. After creating the model the classifier quality is assessed in a *validation* phase. Usually, the validation implies the use of a new data set, called *testing set*, which has been kept apart during the construction of the classifier.

The validation phase provides a comparison between the classes of records predicted by the model with the real ones; therefore, it is appropriate that these be known *a priori*. It is possible to build countless classifiers using various techniques and with different performances, depending on the type of problem to be solved. Through classification, the predicted output may assume only a finite number of possible values such as $\{Yes, No\}$ or $\{High, Medium, Low\}$. There are learning models whose output does not belong to a class but has a numerical value.

In the case of *Regression*, the predicted output is quantitative, which means it can assume continuous values. Therefore, the output variables can assume an unlimited number of values. A possible application regards for example, the use of a regression model to predict the Y profit in euros that a specific X customer will bring over a given period of time.

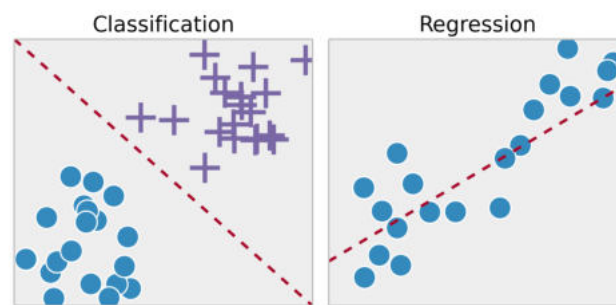


FIGURE 1.4: Classification and Regression

In Figure 1.4, on the left is shown a line that separates the data belonging to different classes (classification), instead on the right (linear regression) a line is shown that approximates as much as possible the data points, estimating a function linking the dependent variable and the independent variable.

Regression has the task of predicting a real-valued label (often called a target) given an unlabeled data. The regression problem is solved by a regression learning algorithm that takes a collection of labeled examples as inputs and produces a model that can take an unlabeled example as input and outputs a target.

Most supervised learning algorithms are model-based. Model-based learning algorithms use the training data to create a model. After the model is learned, the training data can be discarded. Instance-based learning algorithms use the whole data set as the model. One instance-based algorithm frequently used in practice is *k-Nearest Neighbors* (*k-NN*). In classification, to predict a label for an input example the *k-NN* algorithm looks at the close neighborhood of the input example in the space of feature vectors and outputs the label that it saw more often in this close neighborhood. Above, on the left is shown a line that separates the data belonging to different classes (classification), on the right (linear regression) a line is shown that approximates as much as possible the data points, and shows the link between the dependent variable and the independent variable.

A shallow learning algorithm learns the parameters of the model directly from the features of the training examples. Most supervised learning algorithms are shallow. The notorious exceptions are neural network learning algorithms, specifically those that build neural networks with more than one layer between input and output. Such neural networks are called deep neural networks. In deep neural network learning, commonly referred to as *deep*

learning, contrary to shallow learning, most model parameters aren't learned directly from the features of the training examples, but from the outputs of the preceding layers.

1.2.3 Unsupervised Learning

In unsupervised learning, the data set is a collection of unlabeled examples x_i , $i = 1 \dots N$, referred to as a feature vector. The goal of an unsupervised learning algorithm is to create a model that takes a feature vector x as input and either transforms it into another vector or into a value that can be used to solve a practical problem. For example, in *clustering*, the model returns the identifier of the cluster for each feature vector in the data-set. In unsupervised learning [16] examples are provided with related input features, but no output value is provided, so unlabeled data are available.

The goal of unsupervised learning algorithms dedicated to *Clustering Problem* is to find structures within these unlabeled data, in order to identify groupings of similar elements. Instead if the aim is to identify different sources that contributed to the creation of the data, the problem is referred to as *Blind Source Separation*.

Unsupervised learning also includes *dimensionality reduction*, in which the output of the model is a feature vector with fewer features than the input x . In this context, clustering algorithms are adopted to identify groups of data that have common characteristics, while dimensional reduction algorithms are applied to obtain more compact data representations.

Hawkins defines outliers as "*Observation which deviates so much from other observations as to arouse suspicion it was generated by a different mechanism*" [17]. In outlier detection, the output values, referring to a generic example x , measure how much x is different from an example in the data set. There are many application contexts characterized by the availability of data without labels for which unsupervised learning algorithms are useful to discover new properties in the input data. Some of the most important unsupervised learning algorithms are:

- k-means
- Principal Component Analysis (PCA).

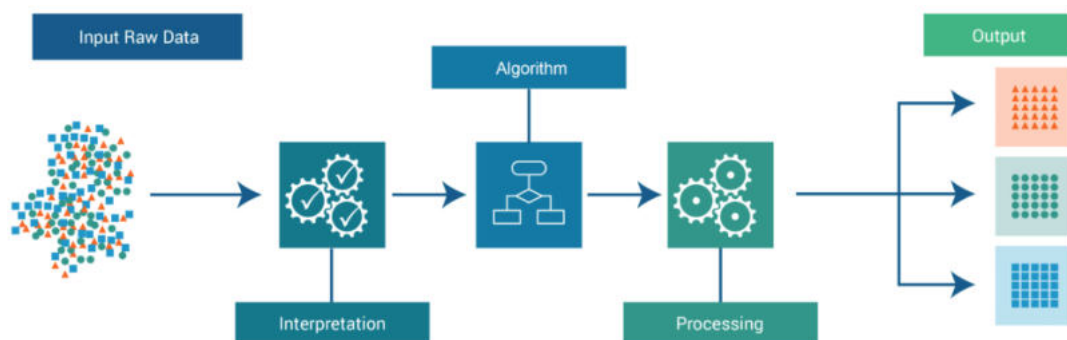


FIGURE 1.5: Unsupervised learning: main modules and data flows

The boundary between supervised learning and unsupervised learning is not always well defined. We will see next an approach which is somehow intermediate, known as semi-supervised learning.

Clustering

Clustering allows the grouping of set of objects of a data set by combining the objects that are more similar to each other. It is an important issue of exploratory Data Mining and a technique widely adopted for the analysis of statistical data in machine learning, image analysis, bioinformatics, and computer graphics.

More formally, clustering means the partitioning of a heterogeneous group into homogeneous subgroups (see Figure 1.6) called *clusters* [18]. The impossibility of giving a univocal definition of "cluster" is found in the number of clustering algorithms presented by researchers [19].

Cluster analysis can be performed by various algorithms that differ significantly in their understanding of what constitutes a cluster and how to efficiently find it. Clustering can refer to either clusters whose members are at small distances from one another, or to dense areas of the data space, or at intervals characterized by particular statistical distributions. It follows that clustering can be formulated in terms of a multi-objective optimization problem. The specific characteristics of the data and the use of the expected results, condition the choice of the appropriate clustering algorithm and of the settings of the distance function, the density threshold or the number of expected clusters.

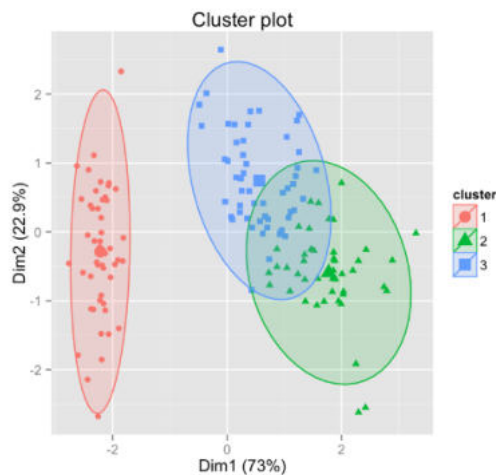


FIGURE 1.6: Clustering example

What distinguishes clustering from classification is that in this case no predefined classes are used. Application examples include the detection of a set of symptoms that indicate a specific pathology, grouping video and audio into homogeneous classes. There is a common denominator: a group of data objects.

However, different cluster models are adopted and different algorithms are available for each one. There are several ways in which algorithms determine clusters. Understanding the various "cluster models" is useful for understanding the differences between the algorithms themselves.

Clustering essentially identifies a set of clusters, which enclose all the objects of the data set. Different clustering approaches can be differentiated according to the hierarchical relationships existing among the clusters of which they are composed:

- *Hard clustering*: it occurs when each object belongs exclusively to a single cluster.

- *Soft clustering*: it occurs when an object belongs to each cluster according to a certain probability.

In general, more detailed distinctions are possible such as:

- *Strict partitioning clustering*: each object belongs exactly to a cluster.
- *Strict partitioning clustering with outliers*: it is possible that some objects do not belong to any cluster and are considered outliers.
- *Overlapping clustering (multi-view clustering)*: objects can belong to more than one cluster.
- *Hierarchical cluster*: there is a hierarchical property by which the objects belonging to a child cluster also appear to belong to the parent cluster.
- *Subspace cluster*: while an overlapping clustering, within a uniquely defined sub-space, clusters are not expected to overlap.

Based on the specific cluster formation approaches different clustering algorithms have been published. Some of them are described in Section 1.5.4.

1.2.4 Reinforcement Learning

Reinforcement learning (RL) is a field of machine learning where the machine “lives” in an environment and is capable of perceiving the state of that environment as a vector of features. RL differs from supervised learning since it is no longer necessary to have only pairs of labeled data and also because non-optimal actions do not have to be explicitly corrected.

The right strategy is to find a balance between exploration and exploitation of knowledge [20]. Often, formulations inspired by Markov’s decision-making processes (MDP) are adopted using dynamic programming techniques [21]. Reinforcement learning algorithms do not require knowledge of an exact mathematical model of the MDP and can be effectively used for large MDPs where exact methods become unachievable [22].

The machine can execute actions in every state. Different actions bring different rewards and could also move the machine to another state of the environment. The goal of a reinforcement learning algorithm is to learn a “policy”. A policy is a function f (similar to the model in supervised learning) that takes the feature vector of a state as input and outputs an optimal action to be executed in that state. The action is optimal if it maximizes the expected average reward. The central part shows the key mechanism of Reinforcement Learning: trial and error, that is the interaction between the environment and the agent, in which the latter tries to maximize the reward by choosing the best actions.

With unsupervised learning, the algorithm discovers from which actions the major rewards are generated, passing through experiments and errors. In fact, the search mechanism for the best actions is based on the trial-and-error heuristic search.

This type of learning includes:

- the agent (who learns or makes decisions),
- the environment (everything the agent interacts with)
- the actions (what the agent can do).

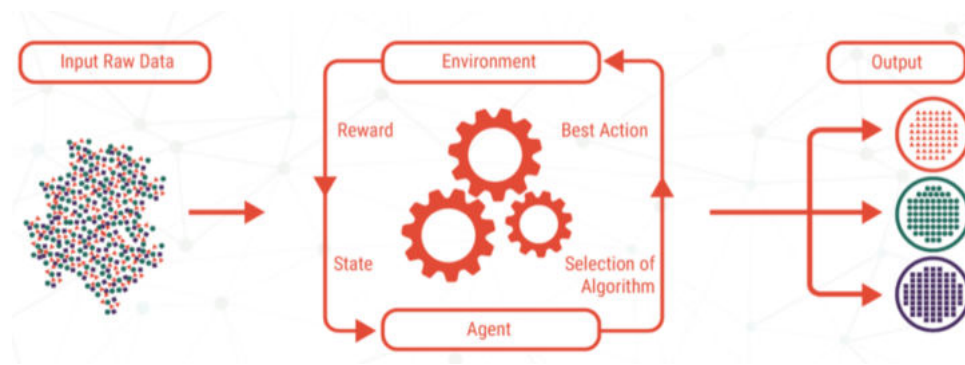


FIGURE 1.7: Reinforcement learning: main modules and data flows.

The agent's goal is to choose those actions that maximize the reward expected in a given time frame. By choosing the right actions, the agent will reach the goal faster. Reinforcement learning is aimed at identifying the best actions to be implemented [23]. From the comparison of the agent's performance with that of an agent that acts in an optimal way, the notion of regret arises.

The agent operates considering long-term results, although short-term ones may be negative. Reinforced learning has been successfully applied to robotics, in telecommunications and in video games.

The possibility of using samples to optimize performance and the wise use of functions approximation for large contexts management make the reinforcement learning powerful. It follows that reinforcement learning is useful in large environments when:

- An environment model is known, but an analytical solution is not available;
- Only a simulation model of the environment is provided [24];
- Interaction is the only way to gather information on the environment.

Referring to these problems just introduced, the first two can be considered planning problems, as some form of model is available, while the last problem can be considered as a learning problem. Both planning problems can be managed as machine learning problems through the adoption of reinforcement learning.

1.2.5 Semi-supervised Learning

The first approach that refers to semi-supervised learning is the heuristic approach to self-labeling [25]. Although the first applications of semi-supervised learning date back to the 1960s [26], the variety of problems for which large amounts of unlabeled data are available has brought it back into fashion. Potential applications include those of texts contained in websites, protein sequences and image analysis [14].

Semi-supervised learning refers to those machine learning techniques that use few data labeled with a large amount of unlabeled data. It follows that semi-supervised learning exploits both supervised learning techniques and those of unsupervised learning ones.

An improvement in terms of learning accuracy is possible through the use of unlabelled data with a small amount of labeled data. Given a learning problem, the acquisition of labeled data is time consuming and the related cost can make difficult to dispose of a completely labeled training set; on the other hand unlabeled data acquisition is less expensive.

These reasons justify the current popularity of semi-supervised learning which has a great practical value. Semi-supervised learning is also of theoretical interest in machine learning being able to simulate human learning.

Semi-supervised learning attempts to use the combined information of supervised and unsupervised examples to overcome the classification performance that could be obtained by discarding unlabeled data using supervised learning or discarding labeled data using unsupervised learning. Semi-supervised learning can refer to both transductive learning and inductive learning. The goal of transductive learning is to infer the correct labels for unlabeled data, while the goal of inductive learning is to predict new unseen data [14]. In order to use unlabeled data, we have to assume a structure for the underlying data distribution. Typically, learning algorithms characterized by a semi-supervised approach adopt at least one of the following hypotheses [25]:

Continuity assumption

Points close to each other are more likely to share a label. The boundaries of decisions in low density regions are such that there are fewer points close together but in different classes.

Cluster assumption

Points in the same cluster are more likely to share a label. This assumption derives from the continuity assumption and gives rise to features learning through the clustering algorithms.

Manifold assumption

If the data is on a manifold characterized by a smaller size than the input space, it is possible to learn the manifold using both labeled and unlabeled data, through distance and density metrics.

In cases characterized by few degrees of freedom, in which the data have many dimensions and come from processes that are difficult to be directly model, the manifold assumption is convenient. Consider the case of the human voice which is controlled by few vocal cords: the adoption of distances and smoothness in the natural space of the generating problem is indicated, rather than in the space of all possible acoustic waves.

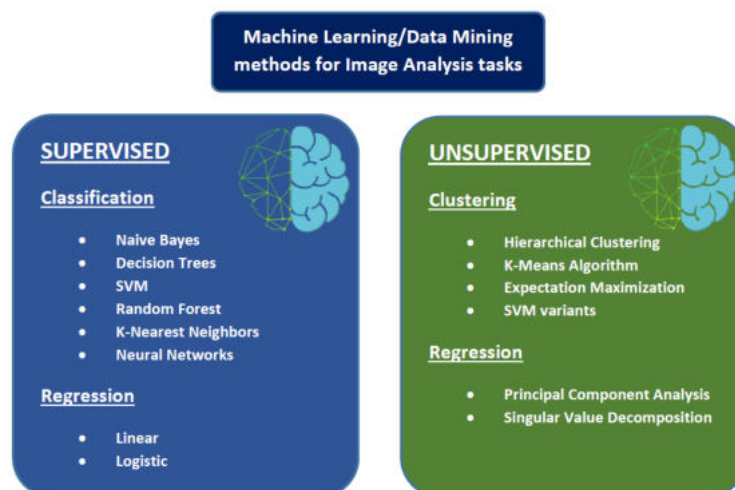


FIGURE 1.8: Major Supervised and Unsupervised learning methods.

Semi-supervised learning takes light from the combination of supervised and unsupervised learning methods (see Figure 1.8). In semi-supervised learning, the algorithms learn from data sets that include both labeled and unlabeled data. Semi-supervised learning algorithms include variants of algorithms of supervised and unsupervised versions, such as the variants proposed for SVM [27] or algorithms for which generative or graph-based approaches are proposed.

1.3 Image Processing

The image processing task, aimed at interpreting and classifying the contents of the images, has attracted the attention of researchers since the early days of computers. With the advancement of computing system technology, image categorization has found increasingly broader applications, covering new generation disciplines such as *image and scene analysis*, *image understanding*, *object recognition* and *Computer Vision*, with applications quite general both in scientific and humanistic fields. On a more general level, the problem is part of Pattern Recognition, the discipline that deals with the automatic recognition and classification of the entity of interest of a phenomenon under observation. Classification tasks also include those related to the categorization of images, such as the construction of a recognition system, the representation of patterns, the selection and extraction of features and the definition of automatic recognition methods. The automatic recognition, description and classification of the structures contained in the images are of fundamental importance in a vast set of scientific and engineering fields that require the acquisition, processing and transmission of information in visual form. Digital imaging techniques are constantly evolving not only in terms of technological refinement, but also from a conceptual view point. The images obtained, thanks to the intervention of electronic processing, lose much of their purely "iconographic" character, in order to acquire an ever greater functional meaning, with an information content to be correctly interpreted. Examples of applications in this sense are:

- Text recognition;
- Medical Imaging;
- Industrial automation;
- Robotics;
- Cartography;
- Remote sensing;
- Environmental modeling;
- Simulation and mobility control in transport;
- Biometrics;
- Conservation of cultural assets;
- Radar Images Recognition in the military sphere.

1.4 Pattern Recognition and Image Analysis

Pattern recognition (PR) has its origins in engineering and it is popular in the context of computer vision [28]–[31].

Through the Machine Learning approach, interest is focused on maximizing the recognition rates, while in pattern recognition there is a greater interest in detecting significant patterns. A useful definition of pattern recognition is:

«The field of pattern recognition is concerned with the automatic discovery of regularities in data through the use of computer algorithms and with the use of these regularities to take actions such as classifying the data into different categories» [12].

The construction of a PR system essentially requires the development of three aspects:

- data acquisition and pre-processing;
- patterns representation;
- the definition and improvement of a decision function for patterns recognition.

When the data are images, the process of developing a PR system is generally instantiated with specific operations concerning the acquisition and processing of the images, the extraction of the structures to be recognized or classified and their representation, operations that fall within the framework of Image Processing and Analysis. In particular, this process can be schematized as shown in Figure 1.9, which also shows the operations performed for each phase and the corresponding results.

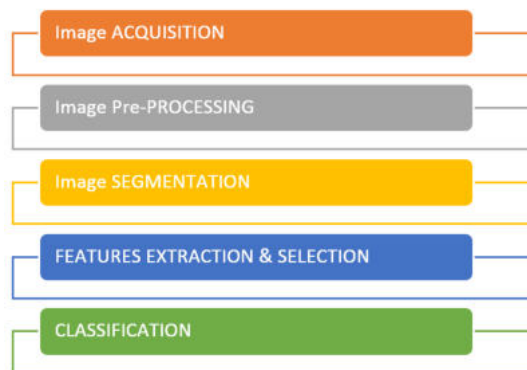


FIGURE 1.9: Steps of an Image Classification System

The first two phases are related to image processing, or rather to the digitization of analog data and optimization of the obtained images; the two subsequent phases belong to the image analysis characterized by the extraction of the features of interest, through the segmentation process: such features are used for the representation of the patterns.

Finally, the last phase consists in defining a classification method translating the quantitative analysis into qualitative information. Furthermore, it is possible to design post-processing phases to improve the entire classification process [32].

1.5 Types of classifiers for image analysis

The classification methods are quite numerous, also due to the fact that several classifiers can be combined together. In general we can identify two main types of approaches: the statistical and the syntactic ones.

Statistical approach generally seeks to maximize the *posterior probability* (i.e. the probability that a sample belongs to a given class), starting from the estimates (obtained from the training set) of the *a priori* probabilities of the classes.

Syntactic approach focuses on the analysis of distinctive features of the objects to be classified: the classification involves the comparison between the structural features of the examples to be tested and those ones of the training set, by using supervised, unsupervised or semi-supervised algorithms.

Apart from the bayesian classifiers [33], which minimize the probability of misclassifying based on the Bayes theorem, in the following we list the classifiers used in *image analysis*, highlighting the advantages and disadvantages of their adoption.

1.5.1 Support Vector Machine

Given two classes of data, the Support Vector Machine (SVM) is a well-known supervised machine learning technique [34], aiming at separating the two classes, with the maximum margin, by means of a hyperplane. When dealing with linearly separable classes, no particular problems arise; if, on the other hand, we are dealing with classes that are not linearly separable we need to use a sort of trick, modifying the SVM algorithm so that it looks for the hyperplane in spaces of higher dimensionality. In fact, if two classes are not linearly separable in a certain multidimensional space, it is probable that they will be if we increase space dimensionality. Although, the SVM operates generally with two classes, there are also some variants that allow us to obtain good results even in case of n classes. Interesting surveys of SVM can be found in [35], [36].

The aim of the algorithm is to determine a *hyperplane* that separates examples with positive labels from examples with negative labels. Support Vector Machine requires the transformation of labels into numbers, so the positive label, for example with the numeric value of $+1$, may represent the class of “sick patients” instead the negative label, with the numeric value of -1 , may represent the class of “healthy patients”. The equation of the hyperplane is given by two parameters: a real-valued vector w with the same dimensionality as input feature vector x , and a real number b . The equation that defines the hyperplane is:

$$w^T x + b = 0 \quad (1.1)$$

where the expression $w^T x$ is the *inner product* between vectors $w, x \in \mathbf{R}^d$. The predicted label for an input feature vector x is given by:

$$y = \text{sign}(w^T x + b)$$

where $sign$ is a mathematical operator that takes any value as input and returns $+1$ if the input is a positive number or -1 if the input is a negative number.

The goal of SVM learning algorithm is to find the optimal values w^* and b^* for parameters w and b . Once the learning algorithm identifies these optimal values, the model $f(x)$ is then defined as:

$$f(x) = sign(w^{*T}x + b^*) \quad (1.2)$$

For the determination of w^* and b^* , it is necessary to solve an optimization problem under constraints. As a first step, the model has to correctly predict the labels of the considered examples (*training phase*): each example i is given by a pair (x_i, y_i) , where x_i is the feature vector representing the example i and y_i is its label equal to -1 or $+1$. So the constraints are:

$$\begin{cases} w^T x_i + b \geq +1 & \text{if } y_i = +1 \\ w^T x_i + b \leq -1 & \text{if } y_i = -1 \end{cases} \quad (1.3)$$

The objective is to separate the two classes by maximizing the margin, defined as the distance between the so called *supporting hyperplanes*:

$$\begin{cases} w^T x + b = +1 \\ w^T x + b = -1 \end{cases} \quad (1.4)$$

A large margin favors the generalization necessary for the correct functioning of the model in classifying new examples. It is possible to show that the margin is computable as $\frac{2}{\|w\|}$; then, in order to maximize it, the optimization problem to be solved is:

$$P \begin{cases} \min \frac{1}{2} \|w\|^2 \\ y_i(w^T x_i + b) \geq -1 \text{ for } i = 1, \dots, N \end{cases} \quad (1.5)$$

where the expression $y_i(w^T x_i + b) \geq -1$ is just a compact way to write the two above constraints (1.3), and $\|w\|$ indicates Euclidean norm of vector w .

The solution of the problem P , obtained through w^* and b^* , is referred as *statistical model*, while the term *training* indicates the process of building the model. Assuming that feature vectors are two-dimensional, the problem and its solution can be visualized as in Figure 1.10.

The blue and orange circles represent, respectively, positive and negative examples, and the line given by $w^T x + b = 0$ is the separating hyperplane.

Any classification learning algorithm that builds a model implicitly or explicitly creates a decision boundary. The decision boundary can be straight, or curved, or it can have a complex form, or it can be a superposition of some geometrical figures.

The form of the decision boundary determines the accuracy of the model, namely the ratio of examples whose labels are predicted correctly. The form of the decision boundary and the way it is computed differentiate a learning algorithm from any another type. Another essential factor to be considered is the speed in building the model and in performing a prediction. In many practical cases, it is preferable to have a learning algorithm that builds a less accurate fast model.

Classification performances can be invalidated both by noise in data, because of the presence of outlier, or by the intrinsic nature of the data that cannot be linearly separated using a hyperplane. In fact, even though the maximum margin allows the SVM to select among

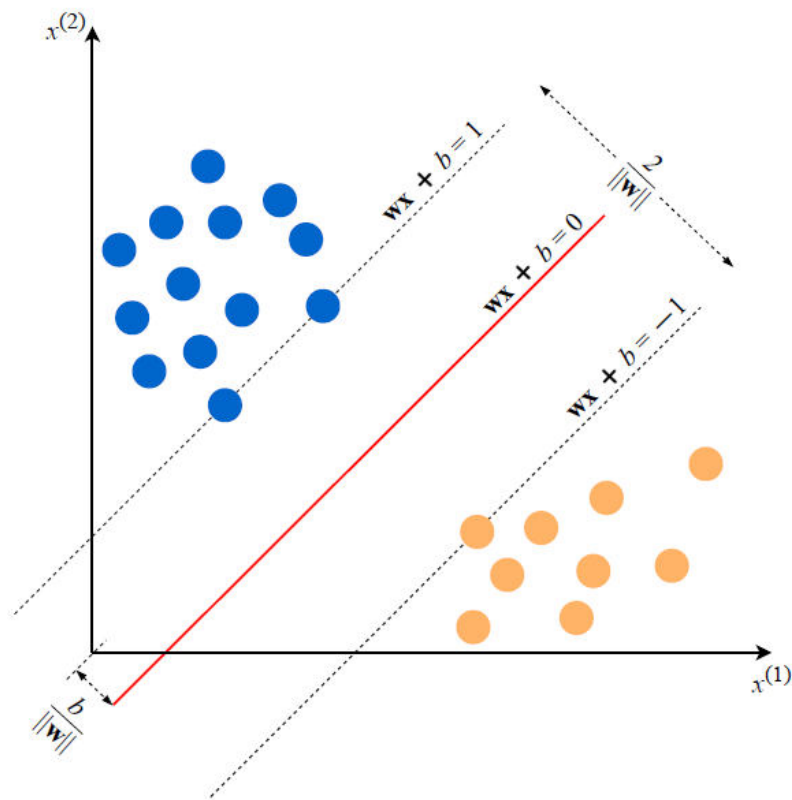


FIGURE 1.10: SVM model for two-dimensional feature vectors

multiple candidate hyperplanes, for many data sets, the SVM may not be able to find any separating hyperplane at all because the data contains misclassified instances.

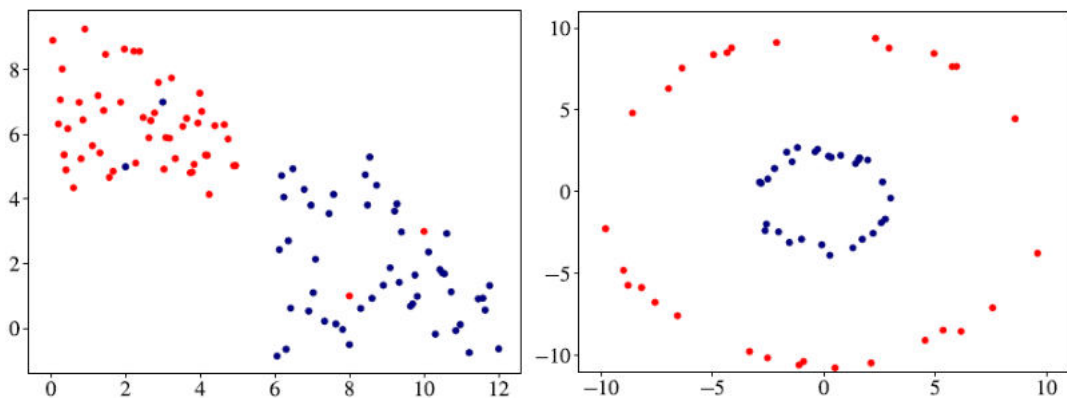


FIGURE 1.11: Linearly non-separable cases: a) presence of noise, b) inherent nonlinearity.

These situations can be effectively managed. In Figure 1.11 a), the data could be separated by a straight line except for the outliers with wrong labels. In the Figure 1.11 b), the decision boundary is a circle and not a straight line. The question of solving the optimization problem arises.

Dealing with Noise

To extend the SVM to cases in which the data is not linearly separable, the so called *hinge loss function*, defined as:

$$\max\{0, 1 - y_i(w^T x_i + b)\}$$

has been introduced.

The hinge loss function is zero if the constraints of P are satisfied, that is x_i lies on the correct side of the hyperplane. For data on the wrong side of the decision boundary, the function value is proportional to the distance from the decision boundary. The aim is to minimize the following cost function:

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \max\{0, 1 - y_i(w^T x_i + b)\} \quad (1.6)$$

where the parameter C determines the tradeoff between the maximization of the margin and the minimization of the hinge loss function.

SVMs that optimize (1.6) are called soft-margin SVMs, while the original formulation P is referred to as a hard-margin SVM.

To effectively manage the non-smoothness in formulation (1.6) due to the presence of the maximum in the objective function, we introduce the auxiliary variables ζ by obtaining:

$$P'' \begin{cases} \min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \zeta_i \\ \zeta_i \geq 0 \\ \zeta_i \geq 1 - y_i(w^T x_i + b) \end{cases} \quad (1.7)$$

The value of C is experimentally tuned: for small values of C , the second term in the cost function becomes negligible, so the SVM algorithm will try to find the highest margin by completely ignoring misclassification. On the other hand, increasing the value of C makes classification errors more costly, so the SVM algorithm will try to focus on minimizing the classification error, by sacrificing the margin.

In other words, since a larger margin is better for generalization, C tunes the tradeoff between correctly classifying the training data (minimizing empirical risk) and correctly classifying future examples (generalization).

Dealing with Inherent Non-Linearity

The use of SVM is possible even when data sets are not separable in their original space via a hyperplane. The transformation of the original space into a higher dimensionality space could imply that the examples will become linearly separable in this transformed space. Using the kernel functions allows us to operate in a high-dimensional feature space, reducing the computational burden. In fact, it is sufficient to calculate the inner products between the images of all the pairs of data without calculating the coordinates of the data in the high-dimensional feature space. This approach is called *kernel trick* [37], and is often cheaper than the computational burden associated with explicit computation of coordinates. The kernel functions have been adopted to deal with problems related to sequence data, graphics, text, images and vectors.

The effect of applying the kernel trick is illustrated in Figure 1.12, where a two-dimensional non-linearly-separable data are transformed into a linearly-separable three-dimensional data using a specific mapping $\Phi : x \mapsto \Phi(x)$, with $\Phi(x)$ being the vector x higher in dimensionality.

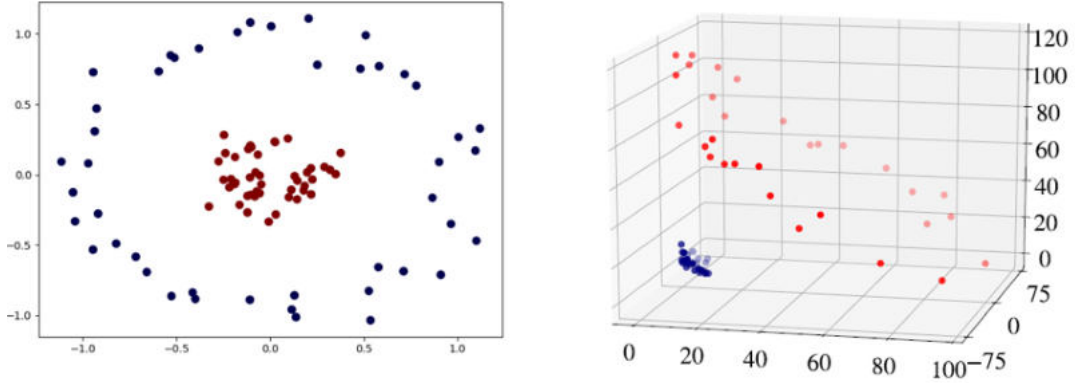


FIGURE 1.12: Data linearly separable after a transformation into a three-dimensional space.

We don't know *a priori* which mapping Φ would work for a specific data-set. To understand how kernels work, it is necessary to understand how the optimization algorithm for SVM finds the optimal values for w and b . Problem (1.7), is generally solved by means of its Wolfe Dual:

$$D \left\{ \begin{array}{l} \min_{\lambda_1, \dots, \lambda_N} \frac{1}{2} \sum_{i=1}^N \sum_{k=1}^N y_i \lambda_i (\mathbf{x}_i^T \mathbf{x}_k) y_k \lambda_k - \sum_{i=1}^N \lambda_i \\ \sum_{i=1}^N \lambda_i y_i = 0, \\ 0 \leq \lambda_i \leq C \quad i = 1, \dots, N \end{array} \right. \quad (1.8)$$

where λ_i are the Lagrange multipliers. In this formulation, D is a convex quadratic optimization problem, efficiently solvable by quadratic programming algorithms. In the above formulation, the term $\mathbf{x}_i^T \mathbf{x}_k$ is the only place where the feature vectors are used. In order to transform a certain vector space into an higher dimensional space, it is necessary to transform x_i into $\Phi(\mathbf{x}_i)$ and x_j into $\Phi(\mathbf{x}_j)$ and then multiply $\Phi(\mathbf{x}_i)$ and $\Phi(\mathbf{x}_j)$.

Through kernel trick, the inner product $\mathbf{x}_i^T \mathbf{x}_k$ can be substituted by a kernel function such as the Radial Basic Function (RBF):

$$\mathbf{k}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$$

where $\|\cdot\|^2$ is the squared Euclidean norm.

By varying the hyper-parameter σ , the data analyst can choose between getting a smooth or curvy decision boundary in the original space.

To summarize, kernels are a special class of functions that allow inner products to be calculated directly in the feature space, without performing the mapping described above [38]. Once a hyperplane has been created, the kernel function is used to map new points into

the feature space for classification. The selection of an appropriate kernel function is important, since the kernel function defines the transformed feature space in which the training set instances will be classified. In [39], Genton described different classes of kernels, without however giving indications as to which class is most suitable for a given problem.

It is a common practice to estimate a range of potential settings and use cross-validation over the training set to find the best one. For this reason a limitation of SVMs is the low speed of the training.

The complexity of an SVM model is not affected by the number of features in training data. It follows that SVMs are suitable for dealing with learning tasks with large number of features compared to the number of instances used for model's training.

1.5.2 Neural Networks

In many real-world problems, the classification is non-linear and the number of features is very high. Artificial Neural Networks (ANNs), provide simpler and more efficient non-linear classifiers. The concept of "natural network" derives from electronic models in order to mimic the neural structure of human brain.

A neural network is a set of artificial neurons interconnected in order to obtain a complex global behavior, which is precisely determined by weighted connections and specific parameters of the neurons. In practical terms, ANNs are non-linear structures of statistical data organized as modeling tools. ANNs can be used to simulate complex relationships between inputs and outputs that other analytic functions fail to represent.

An artificial neuron is the basic unit of ANNs, and has a structure completely inspired by the biological neuron (see Figure 1.13), whose structure is emulated within a computer [40]:

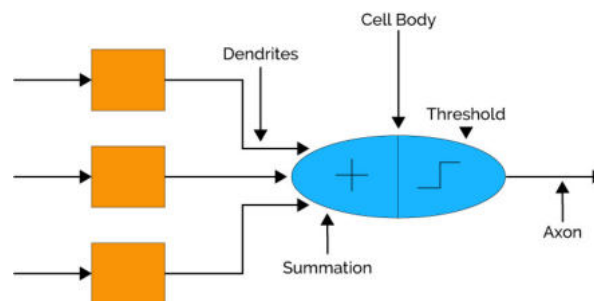


FIGURE 1.13: Mapping between the elements of the biological neuron and artificial neuron

Due to its physiological and chemical properties, the biological neuron is able to integrate, receive and transmit nerve impulses, from/to other artificial neurons.

The dendrites in the biological neural network are analogous to the weighted inputs on their synaptic interconnection in the ANN. The cell body is comparable to the unity of artificial neurons, which also includes units of sum and threshold. Axon is analogous to the output unit of the artificial neural network. Thus, ANN is modeled using the functioning of basic biological neurons. Figure 1.14 reports more in detail the structure of the artificial neuron. We identify the following quantities:

- *the inputs x_i* : they represent the input signals of the neuron, which can come from other neurons or from the environment;

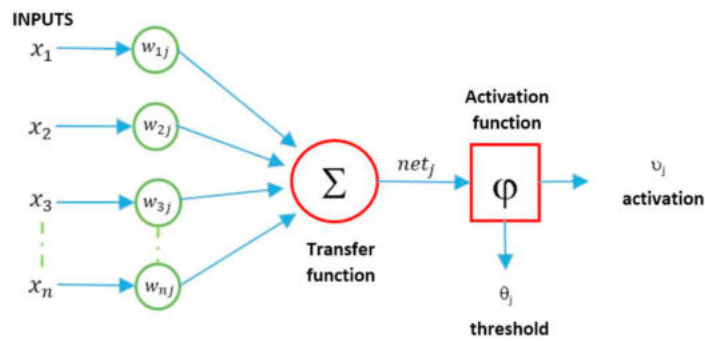


FIGURE 1.14: Artificial neuron: inputs, activation function, weighted connections, calculation function

- *the weights w_i* : they are multipliers associated to the input values and used to weight them;
- *the activation function φ* : it is the mathematical function that determines the output value of the neuron on the basis of the inputs; there are several types of φ , each for specific cases;
- *the transfer function sum*: it generates the value to be submitted to the activation function φ and obtained by adding the input values previously multiplied by the corresponding weights w_i ;
- *the output*: it represents the output signal of the neuron, which can be directed towards other neurons or in the environment.

Thus, an artificial neuron receives input values x_i that are multiplied by a factor w_i , all the resulting values are added together; such sum constitutes the input for the activation function φ that determines the final output value of the neuron. The activation function is useful because it adds possibly non-linearity to the neural networks. In such sense, the ANNs are considered *universal function approximators*. There are different types of activation functions in the literature, each of which has its own advantages and application contexts [41]. Some of them are plotted in Figure 1.15.

All the activation functions, are characterized by the fact that they can be treated, in practice, by minimizing the nonlinear error function of the back-propagation algorithm, useful for learning complex behaviors.

Summing up, in neural networks the basic unit is the neuron that work like simple processor. Each neuron receives the weighted sum of the input nodes and through the activation function generates the output of the next neuron. In neural networks, the neurons are grouped in the so called "layers".

In fact, more complex network topologies include a dedicated layer for input neurons and output neurons and one or more hidden layers (see Figure 1.16).

To design a neural network, the structure of the network (topology), the transfer function and the learning algorithm must be defined. The distinction of different topologies of neural networks depends on the directions of interconnection in the layer; among the most popular topologies, it is worth mentioning *Feed Forward Topology (FFT)* and *Recurrent Topology (RNT)*.

In particular, in the FFT network, the nodes are "hierarchically arranged " in levels. The various levels follow one another starting from the input to the output, where the hidden

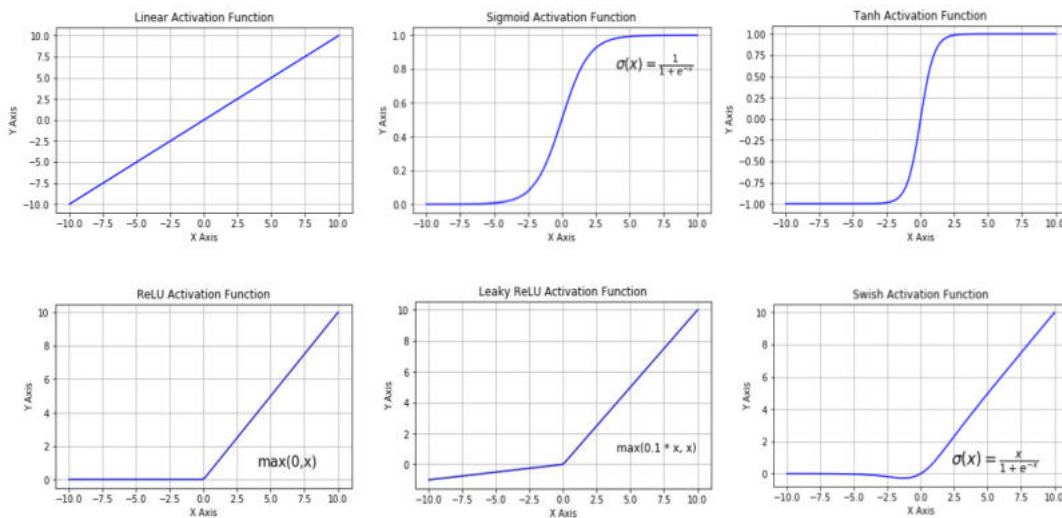


FIGURE 1.15: Types of activation function

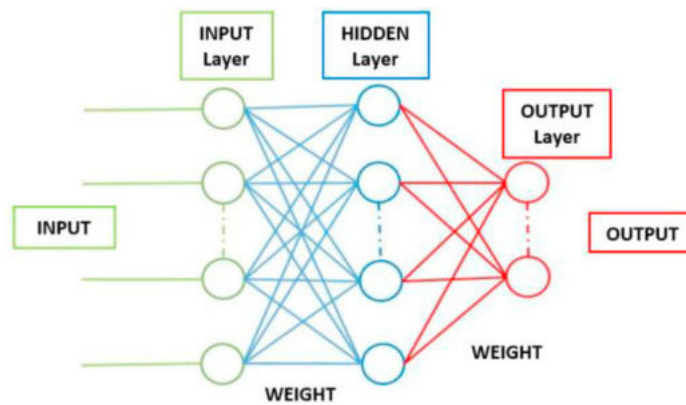


FIGURE 1.16: An example of Neural Networks

levels provide most of the computational power of the network. *Multilayer Perception Network (MPN)* and *Radial Basic Function Network (RBFN)* [42], represent typical applications in which the nodes of each level are connected to the nodes of the next level through unidirectional paths.

The output of a given layer feeds the nodes of the following layer in a forward direction and does not allow feedback flow of information. Unlike the FFT, in the RNT the flow of information between connected nodes is bidirectional. Typical applications of RNT, for example, are *Hopfield Network* [43] and *Time Delay Neural Network (TDNN)* [44].

Training of Artificial Neural Networks

In the previous subsection we described the morphology of neural networks starting from its basic unit, the neuron, introducing the various network architectures. Once all the characteristics of the neural network have been established, such as topology, number and type of neurons and connections, it is necessary to determine the weights of the connections so as to be operatively able to construct a classifier.

These operations, referred to as *training*, will be described together with the most commonly used basic algorithm, followed by the various techniques that use them.

Back propagation is a gradient-type algorithm for an efficient design of a neural network. Basically it is used to modify the weights that connect the various neurons to make them give the expected results. More complex and efficient algorithms are available in literature, such as [45] and [46].

In [45], the problem of neural network training is formulated as a unconstrained minimization of a sum of differentiable error - terms on the output space with respect to the expected values. Solution algorithms of the back propagation-type are considered, in which, taking advantage of the special structure of the objective function, the evaluation of the gradient is divided into several steps.

In [46], a novel approach is presented based on the use of a conjugate gradient method incorporating an appropriate line search algorithm. The algorithm updates the input weights of each individual neuron in a parallel way, using an approach similar to back-propagation algorithm, but with even better performance.

Feed-forward networks often use supervised learning approaches: in order to correctly learn the relationship between input and output, the network is powered for many cycles with different pairs of input and output values.

The *loss function* records the values of one or more variables on a real number, thus providing a "cost" associated with these values. The *error function* evaluates the effect of the propagation of the input through the network in terms of the difference between the actual network output and the expected one.

The algorithm repeats a two-stage cycle, propagation and weight update. When a vector feeds the network, it propagates layer by layer, until it reaches the output level. The loss function is used to compare the obtained network output with the value of expected output.

The resulting error value is calculated for each of the neurons in the output level. Subsequently, they are propagated from the output through the network, until each neuron has an associated error value that reflects its contribution to the original output. Back propagation uses error values to determine the gradient of the loss function. The optimization method used the gradient value to update the weights aiming at minimization of the loss function. Summarizing, the learning algorithm can be divided in the following two phases:

- *propagation*: the propagation phase is characterized by the following steps:
 - forward propagation through the network to generate the output starting from the input;
 - error calculation;
 - propagation of exit activation backwards through the network using the training input target, to evaluate the variations of all the output and hidden neurons.
- *weight update*: for each weight, the following steps must be followed:
 - the difference between the weight exit and the activation of the input are multiplied to find the gradient of the weight itself;
 - a percentage of the weight gradient is subtracted from the weight. This ratio is called *learning rate*, and coincides with the step-size used in the gradient method.

The gradient descent method was the first iterative optimization algorithm used to determine the minimum of a function [47]. It works creating a sequence of points obtained by

moving along the opposite of the gradient direction on the basis of an appropriately calculated step-size.

Gradient Descent

The back-propagation algorithm usually uses the gradient descent method to find the set of weights for minimizing the error rate. In optimization and numerical analysis the gradient descent method is a technique that allows us to determine a stationary point of a function of several variables. The gradient descent technique is based on the fact that, for a given function $f(x)$, the direction P_k of maximum descent at an assigned point x corresponds to the opposite of its gradient at that point $P_k \triangleq -\nabla f(x)$.

The gradient method therefore involves starting from an initial solution x_0 chosen arbitrarily and proceeding iteratively updating it according to the following formula: $x_{(k+1)} = x_k + \alpha_k P_k$ where $\alpha_k \in \mathbb{R}^+$ corresponds to the length of the descent step, whose choice becomes crucial in determining the speed with which the algorithm will converge to the required solution.

Stationary gradient method refers to the case in which, through the choice of a step $\alpha_k = \alpha$ constant for each k , it is possible to define a dynamic gradient method. In the latter case a convenient choice, but computationally more expensive than a stationary method, consists in optimizing, once determined the direction of descent P_k , the function of a variable $f_k(\alpha_k) = f(x_k + \alpha_k P_k)$ in analytical way or in an approximate way. Of great importance is the fact that, depending on the choice of the descent step, the algorithm may converge to any of the minima of the function f , whether local or global.

Generalization, Overfitting and Underfitting

Algorithms, such as back propagation, starting from the error at the exit of the network, calculates how much the weights exiting the last layer of neurons must be corrected to approach the ideal result, using a maximum likelihood estimation. In this way, by minimizing the error, the maximization of the "probability of data" is pursued. This correction can be measured as the *error* from the previous layer, which then repeats the procedure and tries to adjust its own weights to get closer to the requested output: in essence the error "propagates" for the whole network. The procedure is repeated iteratively, until a final error considered tolerable is reached.

In this regard it is important to note that obtaining an excessively small error is not a positive result, because in this case it means that the network has "memorized" the values of the training set, and if presented with new data it will be difficult to work successfully because the model loses the capacity of generalizing during the training.

A recurrent network structure has a sort of memory, which helps storing information in output nodes through dynamic states. The factors that determine how efficient a learning algorithm concerning the ability of:

- minimizing training error;
- minimizing the gap between training error and test error.

These two factors correspond to the main challenges in machine learning of underfitting and overfitting. *Underfitting* occurs when the created model is too simple and fails to well classify the training set, while *overfitting* occurs when the model adapts extremely well to

the training set, and the gap between the error on the training set and the error on the test set is too large, i.e. the algorithm well classifies only the training set of data.

Three causes related to overfitting can be identified:

- presence of noise in the training set records;
- limited training set
- multiple hypothesis testing: since greedy-type algorithms are used, as the space of the solutions increases, the possibility of running into an excellent local rather than a global one will also increase. This happens when the number of attributes is high.

This problem of *overfitting*, can be reduced using methods such as early stopping and weight decay.

Early stopping requires that part of the training set is not passed to the training network, but is kept "hidden" to be used as a validation set; after each back-propagation step the error is measured both on the data used in the training and on the "hidden" data: when over-fitting comes into play it will be noticed that the error on the training data, after being dropped in the early stages of training, will tend to 0, while the one on the comparison data will tend to increase again. As soon as an increase is detected in the error on the validation set, learning has to be interrupted, because starting from that point the network would lose in generality and not learn, but would simply "memorize".

Weight decay, on the other hand, provides for a penalty period, during back-propagation, for weights that assume large absolute values. Very large weights tend to reduce the generality of the network, because they imply an excessive variance, so at the least deviating from the values that the network has learned, the output tends to become unpredictable and unreliable. Therefore penalizing large absolute values can reduce the effects of over-fitting. It is clear that neural networks can have a wide variety of applications: from trend analysis, approximation of functions, filtering and data compression. One of their possible uses is classification: in particular, given a labeled training set, supervised learning can be carried out as explained above, setting the inputs (feature values) and the desired outputs (the labels).

However, it is possible to use an unsupervised approach to train networks, i.e. without providing desired outputs: in these cases the network will have to adapt based on the results obtained in the various training steps; in this way it is possible to carry out clustering operations.

1.5.3 Nearest Neighbor

Among the simplest machine learning algorithms, *Nearest Neighbor* should be mentioned. For our purposes, we recall Nearest Neighbor methods for classification and regression problems, focusing on their performance for binary classification and on the efficiency of implementing these methods. The idea is to memorize the training set and then to predict the label of any new instance on the basis of the labels of its closest neighbors in the training set.

A sample will be assigned to the class of the "nearest neighbor", usually using metric such as Euclidean distance. It is clear that the algorithm will be more expensive depending on the size of training set and on the number of features to be considered: it is appropriate to find

a fair compromise, because a larger training set tends to be more representative, and a high number of features allows to better discriminate between the possible classes. If, on the one hand we have these advantages, on the other we have an increase in the calculations. Some variations of the algorithm have therefore been elaborated, in order to reduce the number of distances to be calculated, for example partitioning the feature space and measuring the distance only with respect to some of the volumes thus obtained.

k-Nearest Neighbor (*k*NN) is a variant that determines the *k* closest elements using a distance metric: each of these elements "votes" for the class to which it belongs, and a new unseen sample will be assigned to the most voted class. *k*NN is a non-parametric learning algorithm that, differently from other learning algorithms that allow discarding the training data after the model is built, keeps all training examples in memory. Once a new, previously unseen example *x* comes in, the *k*NN algorithm finds *k* training examples closest to *x* and returns the majority label for classification problem or the average label for regression problem. A strategy often adopted, both for classification and regression problems, suggests the assignment of weights to the contributions of the neighbors: in this way the contribution provided by the nearest neighbors is greater than the average of the more distant ones. More precisely, a weighting scheme consists in giving each neighbor a weight of $1/d$, where *d* is the distance to the neighbor.

Apart the Euclidean distance, other popular distance metrics include *Chebychev*, *Mahalanobis*, and *Hamming distance* [48]. The choice of these hyper-parameters, i.e. the distance metric, and the value of *k*, is to be made before running the algorithm.

1.5.4 Clustering

Clustering is one of the most widely used techniques for exploratory data analysis. An approach adopted in various research areas involves a study of the reference data, trying to deduce the presence of some significant groups: this is an attempt to deduce intuitively possible cluster formations. Clustering is the task of grouping a set of objects such that similar objects end up in the same group and dissimilar objects are separated into different groups. There may be several very different conceivable clustering solutions for a given data set. As a result, there is a wide variety of clustering algorithms that, on some input data, will output very different clustering. Clustering algorithms can be categorized based on their cluster model.

There is no "correct" clustering algorithm in an absolute sense: "*clustering is in the eye of the beholder*" [19]. The most appropriate clustering algorithm for a particular problem often has to be chosen in an experimental way, except for cases in which it is a mathematical model suggesting the adoption of a solution instead of others. In general, an algorithm designed for a particular context may not perform well when applied to a different data set [19]. As reported in [49], clustering methods can be classified as follows:

- *Connectivity-based clustering*
- *Centroid-based clustering*
- *Distribution-based clustering*
- *Density-based clustering*

Connectivity-based clustering

Connectivity-based clustering, more widely referred to as *hierarchical clustering*, considers that objects are more similar to nearest objects rather than those more distant. These algorithms connect "objects" to form "clusters" based on their distance.

Popular choices are:

- *single linkage clustering*, where distance between two clusters A and B , is defined as the minimum distance between members of the two clusters, namely,

$$D(A, B) = \min\{d(x, y) : x \in A, y \in B\};$$

- *complete linkage clustering*, where the distance between two clusters A and B is defined as the maximum distance between their elements, namely,

$$D(A, B) = \max\{d(x, y) : x \in A, y \in B\};$$

- *average linkage clustering*, where the distance between two clusters is defined to be the average distance between a point in one of the clusters and a point in the other, namely,

$$D(A, B) = \frac{1}{|A||B|} \sum_{x \in A, y \in B} d(x, y).$$

Some algorithms using average Linkage clustering approach are UPGMA and WPGMA algorithms ("Unweighted and Weighted Pair Group Method with Arithmetic Mean"), while SLINK [50] and CLINK[51] are single linkage and complete linkage clustering approaches, respectively.

Finally, another class of clustering approaches is constituted by *hierarchical clustering*, based on constructing a hierarchy of clusters.

A method in this class can be defined as *agglomerative* if it creates clusters by aggregating individual elements, or *divisive* if it starts with the complete data set dividing it into partitions. These methods do not produce a single partitioning, but a hierarchy among which the appropriate clusters must be chosen. These approaches do not effectively manage outliers, which typically create additional clusters or cause the merging of existing clusters. In the general case, both agglomerative clustering and divisive clustering, are too slow for large data sets.

Centroid-based clustering

In centroid-based clustering, a central vector represents clusters. The central vector representing the cluster may not even belong to the data set. Once k , the number of clusters has been fixed, *k-mean clustering* can be formulated as an optimization problem where k cluster centers must be found and objects are assigned to the nearest one in order to minimize the

squared distances from the cluster. The optimization problem is NP-hard, so only approximate solutions are sought, among which Lloyd's algorithm, known as *k-means* algorithm, is one of the best known methods [51]. In order to allow the choice of the best of multiple runs, the variations of *k-means* include such optimizations, but also to limit the centroids to the members of the data set (*k-medoids*), choosing medians (grouping of *k-medians*), choosing less the initial centers randomly (*k-mean ++*) or allowing a fuzzy cluster assignment (*fuzzy c-means*). Generally, *K-means* has a number of interesting theoretical properties:

- data is divided into a structure known as the *Voronoi* diagram;
- it is similar to the nearest neighbor classification;
- it can be interpreted as a variant of model-based clustering and Lloyd's algorithm as a variant of the Expectation-Maximization algorithm.

Most of the algorithms of *k-means* type require the number of clusters k to be specified in advance, which is considered a troublesome drawback of these algorithms.

The algorithms aim at choosing clusters of similar size, so that they can easily assign an object to the nearest centroid. The undesired effect of this assumption is manifested in an incorrect cutting of clusters borders. In any case, only a local optimum is determined by multiple runs performed with different random initializations.

Distribution-based clustering

The clustering model most related to statistics is based on the assumption that clusters can be defined as objects belonging to the same distribution. Clusters can then easily be defined as objects belonging most likely to the same distribution. A convenient property of this approach is that this closely resembles the way artificial data sets are generated: by sampling random objects from a distribution. These methods often suffer from *overfitting*, which can be avoided by placing constraints on the complexity of the model.

One prominent method in this class is based on the *Gaussian mixture models* and use the expectation-maximization algorithm. With this model the data set is modeled considering a fixed number of randomly initialized Gaussian distributions whose parameters are iteratively optimized aiming at adapting to the data but also to contain overfitting. The model allows the identification of a local optimum, so it is not excluded that multiple runs can produce different results. To obtain a hard clustering, objects are assigned to the Gaussian distribution to which they probably belong. Distribution-based clustering is an approach that enables the generation of complex cluster models useful to intercept the correlation between attributes. On the other hand assuming Gaussian distributions is a rather strong assumption for many real data sets, due to the fact that it is not always possible to define an appropriate mathematical model.

Density-based clustering

In density-based clustering [52], clusters are identified by considering the areas of greatest density within the data set. Objects that belong to areas with low density are usually interpreted as noise. DBSCAN [53] is the most popular density-based clustering method; the adopted cluster model is based on connection points within certain distance thresholds, and it is referred as "density reachability".

It only connects the points for which a density criterion applies, typically defined as a minimum number of other objects falling within this radius. It follows that, regardless of shape, a cluster consists of all objects connected to density, in addition to the objects that result within these objects' range. DBSCAN has a low complexity and allows to obtain substantially the same results with each execution: multiple executions are no longer necessary.

OPTICS [54] is a generalized version of DBSCAN with which it is not necessary to set an appropriate value for the interval parameter. OPTICS produces a hierarchical result related to that of linkage clustering.

Mean-shift is a clustering approach in which the objects are allocated in the proximal area with the highest density, evaluating the kernel density estimation. With mean-shift approach, the objects converge towards local maxima of density. The iterative procedure as well as the evaluation of the density estimation are expensive, making Mean-shift slower when compared to the DBSCAN or *k*-means methods. Moreover, if multimedia data are considered, the application of Mean-shift algorithm is hindered by the irregular behavior of kernel density estimate, which produces fragmentation of the cluster tails [55].

K-means algorithm

To effectively use the *k*-means algorithm, we need to define the distance between the clusters and to determine when the merging process between various clusters has to be stopped. The result of such an algorithm can be represented using the clustering *dendrogram* i.e. a tree structure of domain subsets (see Figure 1.17).

In clustering dendrogram, singleton sets are represented in its leaves and the entire domain as its root. This view provides a visual summary of the data. More generally, the pursued approaches can be agglomerative or divisive, according to respectively bottom-up or top-down approach.

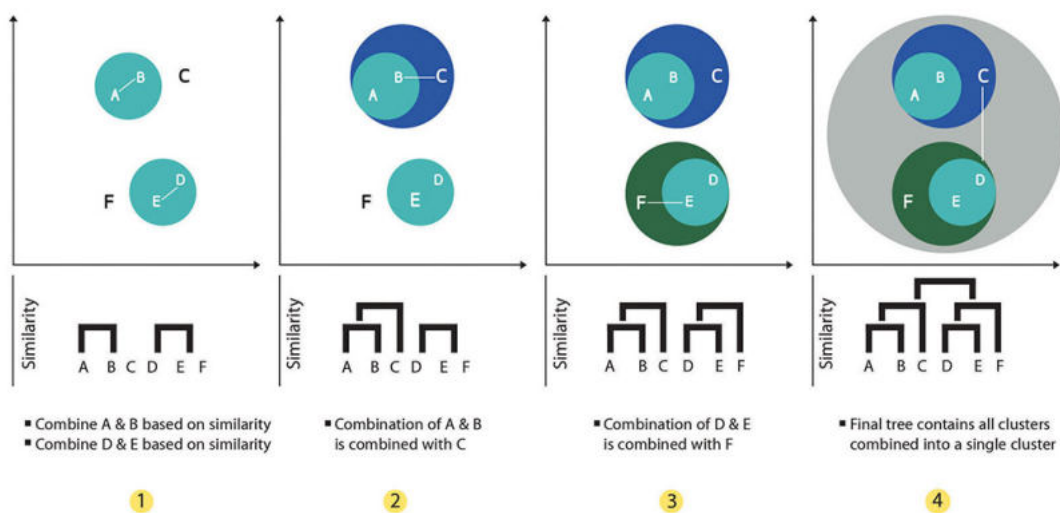


FIGURE 1.17: Hierarchical clustering

In the first case the algorithm starts by trying to aggregate single elements; at each step elements or sub-clusters that are more similar to each other in a cluster are merged into a cluster. In the second case, more complex and therefore less used, the algorithm start with a single cluster and at each level the most different elements are subdivided into sub-clusters. In both cases the result can be represented through a tree.

Common stopping criteria include:

- Fixed the number of clusters k , and stop merging clusters as soon as the number of clusters is k .
- Appropriately defined a limit distance $r \in \mathbb{R}^+$ the process of generating clusters is interrupted as soon as all the distances between the clusters are greater than r . If $r = \max d(x, y) : x, y \in X$ for some $\alpha < 1$, the stopping criterion is referred as “scaled distance upper bound.”

Another approach to clustering starts by defining a *cost function* over a parameterized set of possible clusterings aiming to find a partitioning (clustering) of minimal cost. In this paradigm, the clustering task is turned into an optimization problem, and the solutions implies the use of some appropriate search algorithm. Many common objective functions require the number of clusters k as a parameter. Practically, it is often up to the user of the clustering algorithm to choose the parameter k that is most suitable for the given clustering problem.

The k-means objective function is quite popular in practical applications of clustering. However, it turns out that finding the optimal k-means solution is often computationally unfeasible (the problem is NP-hard, and even NP-hard approximated within some constant). As an alternative, the following simple iterative algorithm is often used. Considering the Euclidean distance function $d(x, y) = \|x - y\|$, the algorithm can be represented by the following pseudo code:

Algorithm 1 : k-Means Algorithm

Input $X \subseteq \mathbb{R}^n$; k = number of cluster;

- 1: **initialize**: Random choose of centroids μ_1, \dots, μ_k ;
 - 2: **repeat until convergence**:
 - 3:
 - 4: $\forall i \in [k]$ set $C_i = \{x \in X | i = \arg \min_j \|x - \mu_j\|\}$;
 - 5:
 - 6: *break ties in some arbitrary manner*;
 - 7: $\forall i \in [k]$ update $\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$
-

This algorithm tends to converge rather quickly (it is rare that more than 10 steps are needed) and to give rather good results, starting from a reasonable initial solution (see Figure 1.18).

K-means has some disadvantages: first the number of K classes must be known *a priori*; furthermore the optimization is iterative and local, therefore it is possible to have convergence on a local maximum of the solution. The *fuzzy* variant of k-means allows a pattern to belong with a certain degree of probability to different classes; this variant sometimes provides a more robust convergence towards the final solution, but suffers in essence from the same problems as the standard k-means.

Several variants have been proposed to solve these problems: for example, to minimize the risk of convergence towards local minima, the algorithm can be performed many times starting from different initial solutions, random or perhaps produced by an evolutionary method (genetic algorithm).

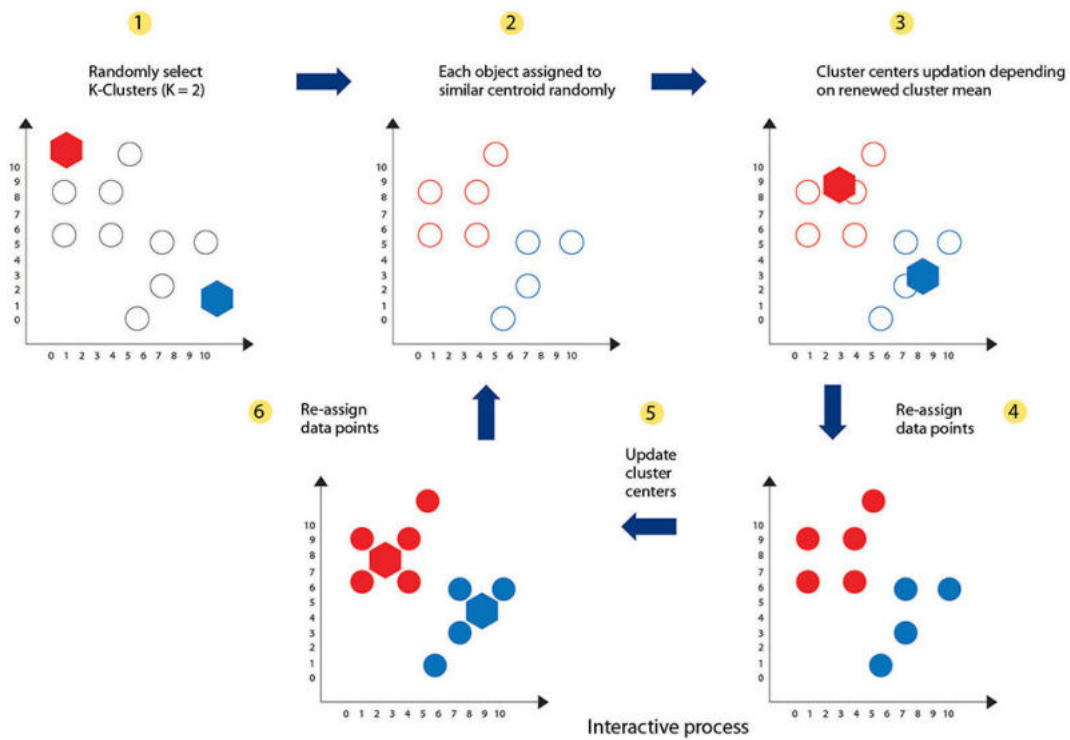


FIGURE 1.18: K-means clustering approach

There are recent approaches that solve the problem faced by k-means. For example, in [56] the authors propose an approach based on the explicit expression of the clustering problem as a non linear, non smooth, non convex optimization problem. Such problem is treated by its reformulation as a DC (Difference of Convex) optimization problem, which in turn is numerically treated by a sequence of partial linearizations.

In [57], a novel DCA algorithms for edge detection was proposed. This proposal adopted clustering on the pixels representing any given digital image into two sets i.e. the "edge pixels" and the "non-edge" ones. The process is based on associating to each pixel an appropriate vector representing the differences in brightness of surrounding pixels. Clustering is driven by the norms of such vectors, thus it takes place in \mathbb{R} , which allows to use a (simple) DC (Difference of Convex) optimization algorithm to get the clusters.

Clustering validation techniques (the determination of the number of classes without this information being known), on the other hand, tend to evaluate *a posteriori* the goodness of the solutions produced for different K values, and to choose one on the basis of a validation criterion that takes account both of the goodness of the solution and of its complexity.

Expectation-Maximization

The Expectation-Maximization (EM) algorithm was presented in 1977 by Arthur Dempster, Nan Laird and Donald Rubin [58], although the method was introduced in other circumstances by previous authors. Rolf Sundberg in [59], proposed an in-depth study of the EM method related to exponential families.

The expectation-maximization (EM) algorithm is an iterative method used to find the maximum *a posteriori* estimation (MAP) of the parameters in the statistical models with the so-called *latent variables*, i.e. deduced from other directly measured variables.

A large number of observable variables can be aggregated to represent an underlying concept. The use of latent variables allows the reduction of dimensionality of data, thus facilitating their understanding.

The *likelihood function* measures the probability that particular values of statistical parameters are associated with a series of observations. Given a certain probability distribution and given a set of observations, the probability of a set of parameters is equal to the joint probability distribution of this random sample. Considering the fixed observations, the probability function is exclusively a function of parameters that index the family of those probability distributions [60]. Mapping from the parameter space to the real line, the likelihood function traces a *hypersurface* whose peak, if it exists, represents the combination of the values of the model parameters that maximize the probability of drawing the sample actually obtained [61].

The estimate of the *maximum likelihood*, evaluated through the natural logarithm of the likelihood (*log-likelihood* function), is the procedure that allows us to determine the arguments on the maximum of the likelihood function. The characteristics relating to the shape and curvature of the probability surface are related to the stability of the estimates; the probability function is therefore usually plotted as part of a statistical analysis [62].

The EM algorithm is used to determine the maximum likelihood parameters of a statistical model if it is not possible to resolve them directly. This happens for example, when not all the data values are present or when it is possible to formulate the model assuming that for each observed data point there exists one that is not observed, specifying the mixture component to which each data point belongs.

The EM iteration alternates between the execution of an expectation step (E), which creates a function for the likelihood expectation evaluated using the current estimate for the parameters, and a maximization step (M), which calculates the parameters that maximize the expected log-likelihood found at step E. These parameter-estimates are then used to determine the distribution of latent variables in the next phase E.

In order to find a maximum likelihood solution, the equations resulting from the derivatives of the likelihood function with respect to all unknown latent values, parameters and variables must be solved.

In statistical models with latent variables, this is generally impossible. The EM algorithm proceeds from the observation that there is a way to solve these two sets of equations numerically. More precisely, once arbitrary values have been chosen for one of the two sets of unknowns, the estimates for the second set are made and iteratively used to find a better estimate of the first set, until the resulting values converge at fixed points [63]. In general, more maximums can occur, with no guarantee that the global maximum will be found.

Aiming to give a more formal appearance to what has been introduced, assume that X is a set of observed data generated by a given statistical model, that Z is a set of unobserved latent data or missing values and that θ indicates a vector of unknown parameters and finally, let $L(\theta; X; Z)$ be the likelihood function associated with these quantities. The estimate of the maximum likelihood (MLE) of unknown parameters can be determined by maximizing the marginal likelihood of the observed data L

$$L(\theta; X) = p(X|\theta) = \int p(X, Z|\theta)dZ. \quad (1.9)$$

This quantity is not always determinable and this is why the EM algorithm tries to find the MLE of the marginal probability by iteratively applying the following two steps:

1. *Expectation step (E step)*. Define $Q(\theta|\theta^{(t)})$ as the expected value of the log-likelihood function of θ , with respect to the current conditional distribution of Z given X and the current estimates of the parameters $\theta^{(t)}$:

$$Q(\theta|\theta^{(t)}) = E_{Z|X,\theta^{(t)}} [\log L(\theta; X; Z)] \quad (1.10)$$

2. *Maximization step (M step)*. Find the parameters that maximize this quantity:

$$\theta^{(t+1)} = \arg \max_{\theta} Q(\theta|\theta^{(t)}) \quad (1.11)$$

Typical models to which EM is applied use Z as a latent variable indicating membership in one of a set of groups. In general:

- The observed data points X may be discrete or continuous. Associated with each data point may be a vector of observations.
- Latent variables Z are discrete, taken from a fixed number of values and with a latent variable for the observed unit.
- The parameters are continuous and are associated with all data points or with a specific value of a latent variable.

It is possible to apply EM to other types of models, considering that if the value of the parameters θ is known, the value of the latent variables Z can typically be found by maximizing the likelihood on the possible values of Z , simply iterating over Z or through algorithms such as Baum-Welch's for hidden Markov models [64].

Therefore knowing the value of the latent variables Z , grouping the data points based on the value of the associated latent variable, it is possible to estimate the parameters θ by averaging the values of the points in each group. Thus, an iterative algorithm, in case both θ and Z are unknown, may be structured as follows:

Algorithm 2 : Expectation-Maximization algorithm

- 1: Initialize the parameters θ on some random values;
 - 2: Given θ , compute the probability of each possible value of Z ;
 - 3: Use the newly calculated values of Z , to compute a better estimate for the parameters θ ;
 - 4: Iterate steps 2 and 3 until convergence.
-

This algorithm approaches monotonically to a local minimum of the cost function. The EM algorithm is widely used in the reconstruction of medical images, in particular in positron emission tomography and in single photon emission computed tomography.

1.5.5 Decision Trees

"Decision tree" refers to a structure similar to a graph where each internal node represents a query on a specific attribute, each branch represents the result of the query and finally each leaf node represents a class label or the resulting decision. In fact, a classification tree is a classifier defined as a set of *if-then*.

One of the main advantages of decision trees is the simplicity of the model. The classification rules are represented by the paths from the root to the leaves. Decision trees are used as a tool to calculate the expected values of several concurrent alternatives. A decision tree consists of three types of nodes that are differentiated in the graphic display [65]:

- 1- Decision nodes - represented by squares
- 2- Probability nodes - represented by circles
- 3- End nodes - represented by triangles

To build the classifier, it is necessary to define the topology of its associated root tree, as well as the subdivision associated with each node. A decision tree can be understood as a predictor $h : X \rightarrow Y$, which determines the label associated with a certain instance $x \in X$, moving through the decision tree from a root node to a leaf one. In the most immediate case of the binary classification, the label will assume dichotomous values, i.e. $Y = \{0, 1\}$; decision trees can also be applied to other forecasting problems. On each node of the *root-leaf path*, the successor child is chosen based on a subdivision of the input space. The tree is then constructed following a greedy procedure in which the new nodes are recursively created and connected to the previously defined ones until a stop criterion is defined.

A widespread rule of division in the internal nodes of the tree is realized considering the threshold of the value of a single function; starting from the root node, the algorithm chooses the right or the left child of the node based on $[x_i < \theta]$, where i is the index of the relevant feature and $\theta \in \mathbb{R}$ is the threshold. The resulting decision tree corresponds to a cell division of the instance space, where each leaf of the tree corresponds to a specific cell. Considering decision trees of arbitrary dimensions, it is possible to obtain a class of hypotheses of infinite size; the risk inherent in this approach is the *overfitting* of the model on the considered data.

To remedy overfitting, the principle of minimum length of the description (MDL) can be used, aiming at learning a decision tree that adapts well to the data but is not too extensive [66]. Unfortunately, solving this problem is computationally difficult. It follows that the decision-making tree learning algorithms are based on heuristics with a greedy approach, in which the tree is built gradually and locally optimal decisions are taken in the construction of each node. Although these algorithms do not guarantee the return of the optimal global decision tree, they work well in practice.

With reference to a generic tree, at the root of which a label is assigned based on the majority vote among all the labels on the training set. At each iteration, the effect of the division of a single leaf is evaluated by defining some *gain* measures able to quantify the improvement due to this division. Concerning all the possible divisions, the configuration that maximizes the gain is chosen, or one chooses not to divide the leaf at all. The features that best divide training data would be the root node of the tree.

There are numerous methods for determining the characteristic that best divide training data such as *information acquisition* [67] and index data [68], but most studies have concluded that there is no single best method [69]. Considering a given data set, comparing individual methods can be important in deciding which metric to use. The process of generation of trees and sub-trees continues until all training data has been divided into subsets of the same classes.

We report below the pseudo-code of a general decision tree algorithm as described in [70]:

Algorithm 3 : Decision Trees algorithm

-
- 1: Check for the above base cases;
 - 2: For each attribute a , find the normalized information gain ratio from splitting on a ;
 - 3: Let a^* be the attribute with the highest normalized information gain;
 - 4: Create a decision *node* that splits on a^* ;
 - 5: Recur on the sublists obtained by splitting on a^* and add those nodes as children node.
-

A decision tree, or any learned hypothesis h , is said to overfit training data if another hypothesis h'' exists that has a larger error than h when tested on the training data, but a smaller error than h when tested on the entire data set. There are two common approaches that can be used to avoid overfitting on training data:

- Interrupt the training algorithm before it reaches a point where it adapts perfectly to the training data;
- Prune the induced decision tree.

Many studies have been presented on techniques to manage overfitting [71]: in general, with the same test and prediction accuracy, the tree with less leaves is preferred. The easiest way to deal with overfitting is to pre-prune the decision tree by not allowing it to grow to its maximum size. Adopting termination criteria as a threshold test for the function quality metric is effective. Decision tree classifiers usually use post pruning techniques that evaluate the performance of decision trees, since they are pruned using a validation set. Any node can be removed and assigned to the most common class of training instances that are ordered to it [72], [73]. To reduce the disadvantages and solve the problems of decision trees related to overfitting, many decision trees can be combined together.

The *random forest* is an ensemble of learning algorithm [74]. An ensemble is a collection of different classifiers put together to create a more powerful model. In particular, the random forest is based on the bagging technique, which is a statistical method to create different training sets starting from just one. The random forest builds a decision tree on each training set, using only a subset of random features for each classifier. Since decision trees are very unstable, building them on different training sets with random characteristics would lead to very different classifiers. This reduces the correlation between the models, thus increasing overall performance. Indeed, to carry out the classification the random forest combines all decision trees with a voting system. Each tree "votes" a class for the record and therefore the random forest chooses the "most voted" as the final result [75]. The voting system averages the forecasts of the individual decision trees, which are affected by a high variance, therefore the results show greater accuracy, even with a large amount of data. This also means that the random forest is less prone to overfitting than a single decision tree, because it averages forecasts, leading to more robust results. Furthermore, it has only a few parameters to set, which is always desirable. The main disadvantage of this approach, however, is the calculation time, which increases proportionally to the number of trees.

Among the best known algorithms for decision trees construction, we should remember ID3 [76] and its extension C4.5 [77]. Subsequently a more efficient version of the algorithm was implemented, called *EC4.5*, which is able to calculate the same decision trees as C4.5 with an increase in performance up to five times [78].

One of the assumptions of C4.5 is that the training data adapt to the memory: this provided the starting point for the creation of frameworks such as *emph Rainforest*, oriented to

the development of fast and scalable algorithms to build decision trees that fit to the amount of main memory available [79]. The various proposals for the parallelization of the C.45 algorithm focused on proposing approaches based on features, nodes and data. Since a decision tree constitutes a hierarchy of tests, an unknown feature value during classification is usually treated by passing the example on all branches of the node where the unknown feature value was detected and each branch generates a distribution of class.

The output is a combination of several class distributions that add up to 1. In decision trees it is assumed that instances belonging to different classes have different values in at least one features. Decision trees work best when discrete and categorical features are used. Using decision trees allows to easily understand why an instance is classified as belonging to a specific class.

1.5.6 Multi-classifiers

A multi-classifier is a system in which different classifiers are used (normally in parallel, but sometimes also in cascade or in a hierarchical manner) to perform pattern classification; the decisions of the individual classifiers are therefore merged to some level of the classification chain. Recently it has been shown that the use of combinations of classifiers (multiclassifier) can improve, sometimes very markedly, the performance of a single classifier. Therefore it may be appropriate, instead of focusing on small improvement of the accuracy of a classifier, to add to it other classifiers based on different features and algorithms. The combination is in any case effective only if the individual classifiers are somehow independent of each other, that is, they do not all make the same type of errors.

Independence (or diversity) is normally achieved by trying to:

- Use different features to identify patterns
- Use different algorithms for feature extraction
- Use different classification algorithms
- Train the same classification algorithm on different training sets (bagging)
- Insist on the training of some classifiers with the most frequently erroneously classified patterns (boosting).

The combination of classifiers is a solution that is gaining ground in the implementation of Computer Aided Diagnosis (CAD) systems that help doctors to take decisions swiftly. These solutions allow to classify particular images in their reference domain. Analysis of imaging in medical field is a very crucial task that allows to diagnose specific diseases at the earliest avoiding costly and invasive investigations [80]. The combination of classifiers can be performed at the decision or at the confidence level.

Merger at decision level

Each individual classifier outputs his own decision, which consists of the class to which he assigned the pattern and optionally of the reliability level of the classification performed (that is, of how much the classifier feels sure of the decision made). Decisions can be combined with each other in different ways like voting schemes and ranking-based schemes.

One of the most well-known and simple methods of fusion is the so-called *majority vote rule*: each classifier votes for a class, and the pattern is assigned to the most voted class; the reliability of the multi-classifier can be calculated by averaging the single confidences. On the other hand, another way is that each classifier produces a class ranking based on the probability that the model to be classified belongs to each of them. The rankings are converted into scores that are added together, and the class with the highest final score is the one chosen by the multi-classifier.

Confidence level fusion

Each individual classifier outputs the confidence in the classification of the pattern with respect to each of the classes, or a dimensionality vector s in which the i -th element indicates the probability of the pattern belonging to the i -th class. Different casting methods are possible, including sum, average, product, max, min.

The *sum method* is a well known method used for its robustness: it expects to perform the vectorial sum of the different confidence vectors, and to classify the pattern on the basis of the major element. A very effective variant is using the weighted sum, where the sum of the confidence vectors is performed by weighting the different classifiers according to their degree of skill: the degrees of skill can be defined basing on the individual performances of the classifiers, for example inversely proportional to the classification error.

1.6 Applications of classifiers in image analysis

The classifiers can be used in different fields of image analysis, starting from simple pixel clustering to advanced operations such as face recognition. However, the topic is still a field of research, and there are no standard approaches or well-established theories: for example, the choice of the type of classifier to be used is more than anything left to intuition or experience. In the same way, choosing a statistical approach rather than a syntactical one has in every case pros and cons: the statistical approach can be more precise and allows us to have a probabilistic indication about belonging to one rather than to another class, but it is rare that the amount of data involved in the analysis of images can be treated within a reasonable time by such an approach; vice-versa, a syntactic approach is faster, but less precise.

In general it is therefore possible to solve the same problem with different types of classifiers, and a certain classifier can be applied to several operations; in the following paragraphs we will see some of the most common applications of classifiers, bearing in mind that they are not necessarily the only ones or the best.

1.6.1 Classification at the pixel level

It can be useful to apply low-level classification algorithms, working directly on features, like the color and intensity of the pixels that make up an image; in this situation the dimensions are not high, and it is therefore possible to think of using a *Bayesian classifier*.

A possible application is the recognition of the pixels representing the skin. Using histograms and a sufficiently large training set, it is indeed possible to derive conditional probabilities: the probability that if we have observed a pixel of skin, this is a generic pixel x which

can be calculated as the percentage of pixels of skin present in a certain range. Through histograms it is possible to divide the adopted space of colors (RGB, grayscale) into a series of value ranges; for example, for the grayscale intensity a possible box division can be $[0 - 64]$, $[65 - 128]$, $[129 - 192]$, $[193 - 256]$. The *a priori* probabilities of the two classes (skin/ non-skin) can be easily calculated as a percentage of the total of the training set, so as previously seen, it is possible to obtain the posterior probability for each pixel. A similar application may be needed in the analysis of a satellite image, in order to associate a pixel with a type of terrain (wooded, cultivated, urban, etc.).

As mentioned above, there is no single method for a given problem: in such case it is possible to use histograms and a Bayesian classifier, but we could prefer unsupervised learning by training a neural network able to detect non-linear relationships, or directly use for partition clustering, the K-means algorithm.

1.6.2 Segmentation

Taking single pixels into consideration is often not enough, as many of the pixels in the image will be potentially useless (for example the background pixels), and a complete analysis leads to a strong performance degradation. It could be desirable to group the related pixels into a compact representation, so as to be able to work on a high-level image, taking into consideration only certain components. The desired features for these representations tend always to be the same, regardless of the specific field of application: the number of components in an image should not be high and the components should be representative of the objects present in the scene. The process of obtaining these components is called *segmentation*.

The task of segmentation occurs in different contexts, and can therefore be tackled in different ways. Speaking of segmentation means to "summarize" the content of a video, dividing it into sequences of similar frames and choosing a representative frame for each sequence: the video is thus summarized by these frames. In this case, *Shot Boundary Detection* algorithms are used to detect the change of scene; such algorithms are for example implemented using MPEG-4 codec (DivX, XviD) to detect the best positions in which to insert a key frame [81].

Another possible application of segmentation is the so-called "Background Subtraction", whose aim is to identify only the relevant part of the image, subtracting the pixels that represent the background, which does not contain useful information. The "Subtraction" is realized through a series of successive frames of the same scene, in which the background can be partially covered by moving objects. Again thanks to segmentation it is possible to detect the presence of a person considering it as a union of several segments (arms, legs, torso): in the image these segments will correspond to uniform regions (if the clothes do not have particular textures), so it will be advisable to segment the image looking for zones of the same color.

To identify buildings in satellite images, the image can be segmented into polygonal regions. The same can be said for the tracking of cars in videos for road surveillance and traffic controls and for the detection of certain mechanical parts.

The most natural approach to the problem of segmentation is *clustering*, since our aim is to represent an image in terms of clusters of pixels that have a relationship between them, not only in terms of intrinsic characteristics such as color and intensity but also spatial (distances,

provisions in space). In particular, both *K-means* and *EM* are often used, since hierarchical methods struggle to manage the large number of pixels present without making use of any technique to "summarize" the most significant data.

K-means can be applied by choosing a certain distance measure, for example one that considers both the color and the position of a pixel, so as to divide the image into its main components; also *EM* can be applied in this case, to obtain segmentation of colors or textures, posing the problem in terms of missing data. It can be assumed that each pixel is generated by a particular probability distribution chosen between N possible (where N is the number of segments that make up the image); each of these distributions may be considered a Gaussian with certain unknown parameters that have to be derived, in order to be able to trace back to which segment generated which pixel, or obtain a segmentation of the image.

Another way to make a segmentation is through *model fitting*, that is to say that a group of pixels must be considered a segment because it belongs to a certain model, for example a line or a circumference. In this case the model is explicit, and the level of abstraction rises: for example it is not possible to search for groups of points that form a line by looking only at the relationships between pairs of points, as is the case in pure clustering. However it is possible to apply a modified version of *K-means* for the fitting of lines: instead of starting from K centers, it is possible to assume straight K in the image, associating at each step a point to the nearest line, then recalculating the line as an interpolation of the points.

Similar to segmentation, *EM* can also be adapted to the fitting of lines. *EM* can however also be applied in the case of "Segmentation of the movement", to determine the ascertainable pixel movement by comparing two successive images: the points of the background will move slightly, those of the objects in the foreground will undergo more significant displacements. Taking into account appropriate spatial relationships it will be possible to identify different movement's zones, making it possible to segment the image.

1.6.3 Object Recognition

Another field of interest is the Object Recognition. This task is an open challenge for computer vision systems and there are many proposals presented and implemented over the last few years. Humans are able to recognize many objects in a single image, even if objects are framed from different points of view or appear in different scales or be partially covered. Many approaches to the task have been implemented over many decades.

This is a very complicated area, which includes a large number of sub-problems and different approaches in which there are still no truly effective standards or general solutions. Classifiers often act in this context, sometimes in a decisive manner, other times acting as a support to optimize other techniques.

Pose Consistency

This approach tries to recognize an object in the image by assuming a mapping between the geometric features of the object and those displayed in the image, thus estimating the pose of the object in the environment. In particular, considering a sufficiently number of features, frame groups are defined. A simple algorithm is the following: for each frame group in the object and for each frame group in the image the possible matches are used to estimate a pose of the object and reconstruct it, reprojecting it on the image; if this rear projection is consistent, we find the object in the hypothesized pose. Clearly the pose will be chosen so

that the rear projection's adherence to the image is maximum. It is clear that the number of possible matches to be calculated can be quite high. A system that can be used to reduce it is Pose Clustering. Generally an object will have many frame groups, and therefore there will be many correspondences between object and image that will indicate the same pose. Conversely, incorrect correspondences, due to noise, will indicate completely different and isolated poses. Instead of checking all possible poses, it is therefore possible to perform a sort of clustering of the poses, analyzing only the most frequent ones during the analysis. In reality this system is not a real classifier, as it operates as a simple voting system to reduce the search tree.

Template Matching

The search for objects with a given form characterizes the problems of Object Recognition, and it can be done by considering the parts that make up the object itself. For example, in face recognition, it is possible to separately search facial elements such as nose, mouth and eyes: the assumption of the template matching is to verify if an object is present in the windows of a certain size within an image.

If size and orientation of the object are unknown, rotation or resizing phases of the search windows are provided. For the template matching task vary methods are adopted from the simplest ones that provide a correlation at pixel level between a sample image and the search window to more complex cases in which it is appropriate to use properly trained classifiers.

The choice of the possible classifiers to be used is wide and depends on the specific problem and on the expected performances both in terms of classification and response time. For example, face detectors have been implemented that use neural networks: these solutions perform well if the faces are frontally framed and the images have uniform lighting conditions.

Identifying a generic object, in normal lighting conditions and from any point of view, requires classifiers trained on an adequate number of examples. To avoid high computation times, working in feature space turns out to be a promising road. Another possibility is the adoption of multi-classifiers: solutions characterized by the use of cascading classifiers working on appropriate descriptive features are very effective both for faces and car recognition.

A very interesting approach involves the use of a "visual vocabulary" for the *real-time recognition* of a specific class of objects. The learning is based on the availability of a set of images appropriately segmented and labeled, while the recognition of the object comes from the clustering in the space of the features [3].

Another interesting method is the *Principal Component Analysis (PCA)* which is often used for the dimensional reduction of feature space. Through PCA new features are obtained, such as linear functions of the originals ones, which occupy a reduced dimensional space. The basic idea of the PCA approach is based on the fact that man is able to recognize many faces, using only a few parameters. Analogously PCA allows to obtain from an image of Eigen pictures (Eigenvector) that summarize the salient information of the image.

Relational Matching

The Relational Matching try to describe an object by exploiting the relationships between characteristic templates that make up the object. Instead of considering a single template,

which would be often too complex to be directly recognized, smaller templates linked together are taken into consideration: for example, instead of trying to directly recognize a whole face, it is possible to separately identify eyes, nose and mouth.

The possible detected templates are put into relations with probabilistic methods, increasing the overall “assembled” time and again: for example, we start from what we believe to be an eye, after which we add the mouth (if it respects the constraints imposed by the model), and so on, until the complete object is obtained. In this situation a classifier trained on a large number of examples can be used to reduce the number of searches to be carried out, eliminating *a priori* the hypotheses that will never lead to a sensible reconstruction.

The problem with the Relational Matching approach is that it may not be able to handle complex objects. For now these systems work quite well with simple objects (such as a face), and it seems difficult to build matchers for more complicated models.

1.7 Take away

The classifiers are widely used in various fields of computer vision, and many times they are used in conjunction with other geometric and probabilistic techniques of image analysis.

So far there are no standards or guidelines regarding the choice of the classifier type, nor for the configuration of a specific classifier.

Generally the best approach is to evaluate the type of problem and try to use the most suitable classifier type, eventually even adjusting it and optimizing it with some trial and error steps. Many classifiers and related applications require deal of computations, and often good performance in terms of CPU times are not easy to achieve. On the other hand good results seem to be obtainable with multiclassifiers and cascading classifiers like in real-time object detection.

The main objective of the present work concerns the task of medical image classification. In the next chapter we will focus on a recent machine learning paradigm, known as Multiple Instance Learning (MIL), which proves to be particularly suitable for medical images and videos analysis.

MIL algorithms by exploiting the class labels assigned globally to images or videos, allow the detection of significant patterns at the local level useful to derive the global classification. Manual segmentation is not necessary to train MIL algorithms, unlike traditional SIL (Single-Instance Learning) ones. As a result, the troublesome manual image labeling phase is no longer necessary at a local level, since it is sufficient to know only the global label of the data. These solutions are attracting interest from the Medical Image and Video Analysis (MIVA) community.

In addition , MIL algorithms guarantee good classification performance, allowing specialists not to lose sensitivity on data. The MIL approaches outline ideal solutions for image and video analysis activities in a medical context.

Chapter 2

Multiple Instance Learning Problems: models and algorithms

"Ignoranti quem portum petat nullus suus ventus est."

[There is no favorable wind for the sailor who doesn't know where to go.]

– Seneca, Letters to Lucilius, letter 71; 1975, pp. 458-459

Alan Turing in 1951 introduced the term of “artificial intelligence” referring to the attempt to make programmable machines capable of analyzing and making decisions like humans. Over time, “artificial intelligence” has developed research tracks such as expert systems and machine learning, which have enormously grown in recent years thanks to the availability of powerful computers and large and ever increasing amounts of digital data.

Multiple instance learning (MIL) is a variant to a variation of supervised learning initially proposed in the 1990s to solve the prediction problem of pharmacological activity [82]. MIL is one of the most recent and promising solutions, and is a weakly supervised form of learning in which training instances are organized in sets, called bags, and a label is provided for the entire bag. This formulation is gaining interest because it naturally fits various problems and it makes possible even to exploit weakly labeled data. As a result, it has been used in various fields of application such as computer vision and document classification. MIL provides a much more natural representation than that used in classical machine learning, where a single feature vector is used per object. Multiple instance classification (MIC) is one of the most popular sub-paradigms of MIL, but not the only one. In recent years, articles on multi-instance regression (multi-instance learning with continuous output) and multi-instance clustering have appeared too.

One of the areas where MIL approach is gaining success is image classification task in the medical field, as it allows to automatically detect target models locally in images or videos and to propose automatic diagnoses for each image or video as a whole. However, learning from bags creates important challenges from a mathematical point of view. We will analyze MIL problems according to four main aspects: the composition of the bags, the types of data distribution, the ambiguity of the instance labels and the activity to be performed. Referring to recent works in the literature, it emerges that the methods that extract global information at the bag level generally show a superior performance. The analysis of the introduced aspects will allow us to establish why some types of methods have more feedback than others as well as to outline useful guidelines for the design of new MIL methods.

2.1 Introduction

Multiple instance learning (MIL) has attracted a lot of attention from the research community, especially in recent years [83], [84]: the nascent interest is linked to the different nature of the problems that scientific and industrial communities are called to face. The availability of useful data to deal with major problems has increased exponentially. This is especially in classification tasks where data labeling has always been a long-standing problem, weakly supervised methods, such as MIL, appear to be useful because weak supervision is generally used more efficiently.

Multiple instance learning deals with training data organized in sets, called bags. Supervision is provided only for whole sets and the individual labels of the instances inside the bags are unknown. This assumption implies that images collected directly from Web can be used to train object detectors, using tags associated with images as labels, taking advantage of weak supervision rather than using locally annotated data sets [85], [86]. Medical images are often characterized by only global labels related to patient diagnosis. By using weak supervision, however, it will be possible to use them to train computer-aided diagnosis algorithms. Computer vision systems are of particular interest as these integrated solutions are able to provide an increasingly reliable second opinion to the specialists, and from them it is possible to derive mobile applications that directly support the patient in self-diagnosis and in the follow up of some diseases. In this sense the capacity of cameras and microphones already equipped with smartphones as well as biometric signals that can be extracted from wearable sensors feed the system with a continuous flow of data.

Alongside the possibilities described above it should not be overlooked that there are different types of problems that lend themselves to being formulated as MIL problems. The first applications of this technique concern the problem of the prediction of pharmacological activity [82], in which it is expected that a molecule can induce a certain effect. Since the single molecule can assume many conformations, each of which can alert or not, the observation of the effect of the single conformations is impossible. The solution to the problem involves the observation of molecules as a group of formations: these premises have led the scientific community to provide the MIL formulation. Recently, MIL has been increasingly used in many other fields of application, including medical images and videos classification [87],[88], text classification [89] and sound event detection [90].

While, on the one hand, the interest in MIL methods is growing in view of possible applications in various contexts, on the other hand, the knowledge of the main characteristics of MIL problems has not been yet totally examined. The experimental results are not always easy to interpret, and it happens that the proposed algorithms are applied to sets of inappropriate reference data. Another pitfall that must be taken into consideration concerns the accuracy of the used models which, despite of good performance on synthetic data, do not always generalize properly when applied to real world data. It is therefore important for our aims to analyze the characteristics of MIL problems focusing on the ambiguity of instance labels and on the grouping mode of data in bags. Recent literature [84], proposes to group problem characteristics into four categories:

- Prediction level
- Bag composition

- Label ambiguity
- Data distribution

Each of the listed characteristic offers interesting challenges to be addressed. A fundamental concept concerns the level to which the predictions can be executed once the instances have been grouped according to some criterion in the bag. A bag-level prediction instead of an instance-level prediction [91] has different misclassification costs, and requires the adoption of specific algorithms [92], [93].

The proportions of the instances of the various classes and their relationships, also due to the potential presence of noise on the labels, affect the performance of the MIL methods. Problem characterization cannot be ignored when new MIL methods are proposed and comparative experiments are conducted. There are various aspects that must be investigated in all application contexts. In computer vision, for example, instances can be spatially related, but this relationship is not exploited in most bioinformatics applications.

The rest of the chapter is organized as follows. In Section 2.2 we will focus on MIL assumptions looking at different learning activities that can be performed using the MIL framework.

In Sections 2.3 and 2.4 we discuss respectively characteristics of MIL problems and the possible paradigms to use for solving them.

The more important instance space-model are recalled in Section 2.5. Finally, in Section 2.6, a general further overview of the literature is provided.

2.2 Multiple Instance Learning Assumptions

When talking about MIL, a first aspect to understand concerns the two fundamental assumptions for MIL problem: the standard assumption and the collective assumption [94].

The standard MIL assumption states that negative bags must contain only negative instances and consequently that positive bags contain at least one positive instance. In the literature, these positive instances are often referred to as *witnesses*. Binary classification works fine in application contexts such as the medical one, when to make a diagnosis it is necessary to decide between benign or malignant lesion.

To give a more formal definition, let us suppose that X is a bag defined as a set of feature vectors $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ and each x_i belonging to the feature space \mathcal{X} . Each instance is associated with a specific class using a function $f : \mathcal{X} \rightarrow \{0, 1\}$, where the negative and positive classes correspond to 0 and 1 respectively, i.e. to benign versus malignant diagnosis. The bag classifier $g(X)$ is defined by:

$$g(X) = \begin{cases} 1, & \text{if } \exists x \in X : f(x) = 1; \\ 0 & \text{otherwise} \end{cases} \quad (2.1)$$

This assumption is adopted in many of the early methods [82], [95], and it is also effectively used in recent methods [96]–[98].

To correctly classify the bags according to the standard assumption, it is sufficient to identify at least one positive instance in each positive bag and no positive instances in each negative bag. See for example Figure 2.1, where a separating hyperplane correctly classify

the represented bags. In particular, the red bags represent the positive bags while the blue bags represent the negative ones.

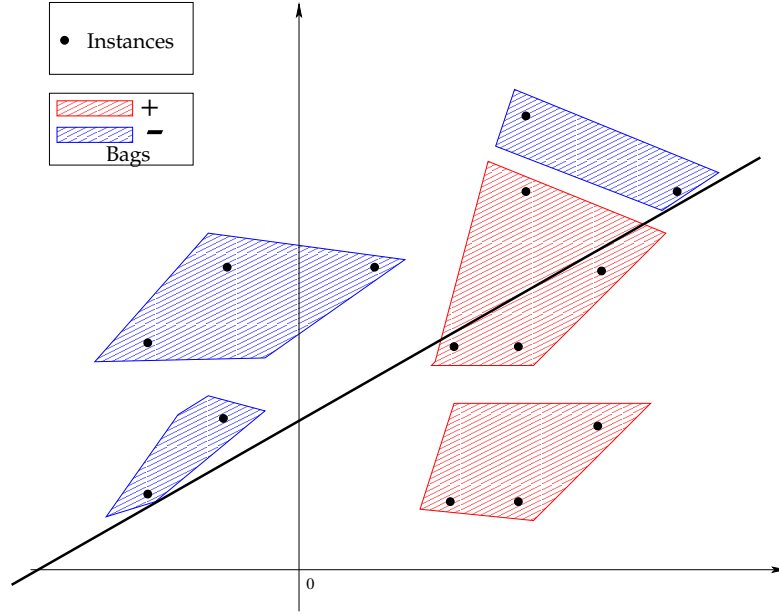


FIGURE 2.1: MIL approach for binary classification

In many real problems it is necessary that several instances must be positive to correctly assign a positive label to the whole bag. For example, when surveying a forest, a single tree is a positive example. However, an image containing only one tree cannot be defined as a forest for which the image cannot be considered positive. In this case a bag classifier can be provided by:

$$\begin{cases} 1, & \text{if } \theta \leq \sum_{x \in X} f(x) \\ 0 & \text{otherwise} \end{cases} \quad (2.2)$$

where θ indicates the minimum threshold of positive instances needed to define positive the whole image. By *collective assumption* we refer to the case in which the instances involve more than one concept in order to correctly belong to a class. The example described in [94], widely reported in the literature, refers to the concept of “beach” which must imply both the presence of sand and of the sea. In cases such as those described, for a correct images classification, the model have to verify the simultaneous presence of both types of witnesses. This circumstance invalidates the operation of many standard MIL approaches and it is overcome when a positive bag requires the contextual presence of instances belonging to different positive classes. In this case the bag classifier $g(X)$ can be defined as:

$$\begin{cases} 1, & \text{if } \forall c \in C^+ : \theta_c \leq \sum_{x \in X} f_C(x) \\ 0 & \text{otherwise} \end{cases} \quad (2.3)$$

where C^+ is the set of the positive classes, and $f_C(x)$ is a process that generates 1 if x belongs to the class C and θ_c indicates the number of instances belonging to C required to observe a positive bag. To handle an assumption with multiple classes, such as the case just described, various proposals have been made. In particular, in [99] the bag space is defined as the set of all distributions of probability on the instance space. The author proposed that bags should

be treated as latent distributions from which samples are observed. G. Doran showed that it is possible to learn accurate instance- and bag-labeling functions in this setting as well as functions that correctly rank bags or instances under weak assumptions.

2.3 Characteristics of MIL problems

In MIL literature four categories of key characteristics associated with MIL problems are identified:

1. Prediction level
2. Bag composition
3. Data distribution
4. Label ambiguity

These categories of characteristics influence the behavior and performance of MIL algorithms. Knowledge of the characteristic facilitates the proposal of various algorithms useful for dealing with specific contexts. Moreover, each characteristic raises particular challenges that must be faced.

2.3.1 Prediction: instance-level vs. bag-level

If we consider the object localization in images, the goal is reached if the individual instances are classified. Solving this problem implies that in the classification task, the task is to learn $f(x)$ instead of $g(x)$. In this particular case, a perfect classifier of instance $f^*(x)$ would result in a perfect classifier of bags according to the standard MIL assumption:

$$g^*(X) = \begin{cases} 1, & \text{if } \exists x \in X : f^*(x) = 1; \\ 0 & \text{otherwise} \end{cases} \quad (2.4)$$

Viceversa, since an instance can be viewed as a singleton bag, a perfect bag classifier $g^*(X)$ allows perfect instance classification. Given a dataset, the relationship between optimal classifiers is no longer reciprocal, in the sense that a perfect instance classifier still leads to an optimal bag classifier, but the inverse is not true. In MIL, instance labels are not available and therefore training an instance classifier is not trivial. In fact, many methods use the accuracy of bag classification to train an instance classifier in the hope that bag-level accuracy is representative of instance-level accuracy.

The bag level accuracy of a method does not reflect its accuracy at the instance level. This differences in the cost function of the two activities highlight this aspect. In [92], it has been shown that the relationship between accuracy at the two levels depends on many factors:

- number of instances in bags,
- class imbalance
- accuracy of the instance classifier.

It follows that the algorithms designed for bag classification are not necessarily effective for the instance classification.

According to the MIL standard assumption, a bag is labeled positive when a witness is identified by leaving out the labels of all other instances.

In this case, false positives (FP) and false negatives (FN) do not influence the accuracy of the classification of the bags, but are still counted as classification errors at the instance level.

In MIL literature, a lot of methods deal with bag classification problem [83]. Many proposed methods use measure bag classification accuracy to train an instance classifier. For these purposes, predictions of the instances of a bag are aggregated, i.e. using the max function or a differentiable approximation, and the loss is calculated with respect to the bag label. This idea has been adopted to train models such as logistic regression [100], boosting classifier [101] and deep neural networks [86]. Conceptually, these methods, although proposed for the instance classification, are not different from the bag classification methods working in the instance space. These methods individually classify instances before predicting bag labels; it is therefore clear that they can be directly used for instance level classification.

Since the use of bag classification accuracy as a surrogate optimization target is sub-optimal, it has been proposed to consider negative and positive bags separately in the classifier loss function [102]. The accuracy on positive bags is assessed at bag level; conversely, all instances are treated individually for negative bags.

This criterion has been proposed to improve the decision threshold of bag classifiers for instance classification and to improve their accuracy [103].

In [104], within a SVM approach, a different weight is assigned to FP and FN. Virtually any bag-level classifier can classify instances if they are viewed as singleton bags. Some methods operate in discovering the true label of the instances and then subsequently training an instance classifier.

One of the best known methods reported in the literature is *mi-SVM* [95]. This method involves two iterative steps. First, the instance labels are initialized, and then an SVM classifier is trained and used to update the label assignment. The method continues until the label assignment remains unchanged. The SVM classifier thus determined is used to predict the label of test instances.

In MILD [105], the authors propose a novel disambiguation method to identify the true positive instances in the positive bags. The probability that an instance is positive depends on the bag labels in its vicinity defined by a Gaussian kernel. The detected positive instances are used to train an SVM classifier.

To transform the MIL problem into a classic single-instance learning (SIL) problem, two feature representation schemes are proposed, respectively for instance-level and bag-level classification.

2.3.2 Bag composition

Witness rate

The witness rate (WR) indicates the relationship between positive instances in positive bags. High WR values indicate that the positive bags mainly contain positive instances and it is possible to abstract that the label of the instances is the same as the label of the bag to which they belong. The problem can be addressed as a supervised problem with unilateral noise [106]. In some applications, WR despite being low can hinder the performance of many

algorithms. For example, in methods such as Diverse Density (DD) [107] and APR [82] instances are considered to have the same label as their bag.

When the WR is low, the simplification according to which it is possible to associate to instances the label of the bag to which they belong implies lower performances. For example, in methods that analyze the instance distribution in bags [108] a low WR creates conflict because the positive and negative bag distributions become similar. Finally, in instance classification problems, lower WRs arise from serious class imbalance problems that lead to poor performance for applied algorithms.

Several authors have put forward proposals for methods dedicated to contexts characterized by low WR. In [109] a sparse transductive MIL (stMIL) is proposed which is a SVM formulation that, in order to better manage the low WR bags, requires the modification of the SVM optimization constraints such as to be satisfied in correspondence with the identification of at least one witness in positive bags. To solve this situation, Sparse balanced MIL (sbMIL) [109], adopts a WR estimation as a parameter in the optimization function.

Relations between instances

Most existing MIL methods assume that positive and negative instances are sampled independently from a positive and a negative distribution. However, in the real world data there is a correlation between instances and bags [110]. In literature we distinguish three types of possible relationships: similarities within the bag, co-occurrences of instances and structure.

- *Similarities within the bag.* In some problems, the instances belonging to the same bag share similarities with the applications of other bags. Consider the case of medical images of an organ which, although extremely similar, are indicative of a healthy state rather than a pathology. In this case it is likely that all the segments share some relative similarities such as for example a lighting factor or the presence of noise due to the image acquisition technique. Even the segmentation algorithms used could cause the background of an image to be divided into very similar segments. The similarity of the instances is enhanced by their proximity in the metric space used by the classifier. Depending on the type of data, the similarity or dissimilarity can be measured using different distance measurements like Euclidean, cosine or χ^2 .

A good way to mitigate the problems related to similarity within the class is to define a new instance space in such a way that the distance is more related to the class than to the bag membership. The choice of the new space can be obtained by taking into consideration the features that actually discriminate between the classes. The goal is to maximize the distance between the negative instances and the most positive instance of each positive bags. Different methods include the integrated selection of weighting functions or mechanisms.

Finally, feature learning methods project instances into a dimensionless space in which class discrimination is applied at bag level. Usually this means maximizing the distance between negative instances and the most positive instance of each positive bag in the projection space.

- *Instance co-occurrence.* When instances share a semantic relationship, they co-occur within the bags. This type of correlation appears when it is more probable that the subject of an image is placed in a certain context rather than in another, that is when

some objects are often found together (for example, knife and fork). Referring to specific application cases, the co-occurrence represents an opportunity to be seized to have greater precision. In certain cases the co-occurrence represents a necessary condition to obtain a correct classification. One of the most reported examples to explain this concept is due to Foulds and Frank [94] and concerns the classification of images of beaches, even considering images containing only “desert” and only “sea”.

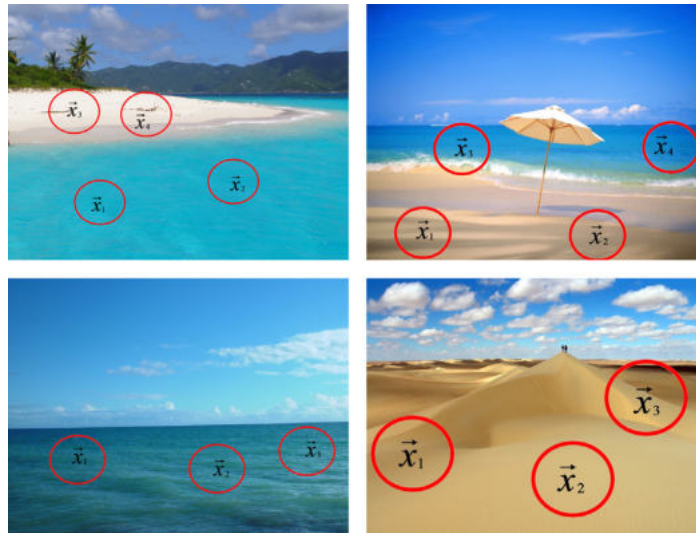


FIGURE 2.2: Classification of images into beach (top row) and non-beach (bottom row) [94].

One image will belong to the *beach* class, if instances of the class sea and sand are in the same bag. Conversely, if only one of these classes is present in the image, then the class is *non-beach*. This type of data occurs rather frequently in image classification activities. Most methods that work under the collective assumption [94] naturally exploit co-occurrence. Many of these methods represent the bags as distributions of instances that indirectly explain the recurrence. Although useful for classifying scholarships, for example classification problems, the recurrence of instances can confuse the student. If a given positive instance often coincides with a given negative instance, it is more likely that the algorithm considers the negative instance positive. This occurrence would lead to a higher false positive rate (FPR).

- *Instance and bag structure.* Some problems refer to data characterized by an underlying structure between the instances extracted from the same bag or between different bags [111]. The term “structure” means a more complex concept of simple recurrence, in the sense that the instances are significantly correlated or follow a certain order. Understanding the structure is possible through the analysis of the considered domain, and can lead to better classification performances [112], [113]. In general, structure can be spatial, temporal, relational or even causal. For example, when the bag is a video sequence, all the frames are naturally ordered both in time and space. Capturing the relationships of a particular context is not easy. To facilitate this task, dedicated graphic models have been proposed that exploit the structure to model the relationships between stock exchanges, instances or both [114], [111]. The structure, both temporal and spatial, between the instances can be modeled in different ways. In Bag of Words

(BoW) models for computer vision, this can be achieved by splitting images [115] or videos [112] into different both spatial and temporal regions of interest (ROI). Each zone is individually characterized and the final representation is the concatenation of all ROI feature vectors.

2.3.3 Data distributions

Similar to traditional supervised learning, different MIL works are proposed under the assumption that a representative training set is available for a proper learning of the classifier. That is to say, the training data can appropriately describe the distribution of positive and negative data in the testing set. However, this assumption may not be always satisfied. In real-world MIL applications, the negative data in the training set may not sufficiently represent the distribution of negative data in the testing set. When a representative training set is not available, learning an appropriate MIL class becomes crucial for real applications.

The choice of design of MIL algorithms must take into account some delicate aspects concerning the nature of the overall distribution of the data.

Multimodal distributions of positive instances

Some MIL algorithms work by assuming that positive instances are found in a single cluster or in a region of feature space. This assumption has been used by several early methods such as Axis-Parallel hyper-Rectangles (APRs) [82], which tries to determine a hyper-rectangle that maximizes the inclusion of instances from positive bags, excluding instances from negative ones or methods such as Diverse Density (DD) [107] that follow a similar idea using a framework to learn a simple description of a person from a series of images (bags) containing that person. These methods look for the point in the function space closest to the instances in positive bags, but away from the instances in negative bags. This point is considered the “positive concept”.

Some more recent methods also follow the assumption of the single cluster. In [97], the classifier is a sphere that includes at least one positive instance from each positive bag excluding instances from negative bags. In [97] the authors proposed SDB-MIL, a Sphere-Description Based approach for Multi Instance Learning. SDB-MIL identifies an optimal separation sphere by determining a large margin between instances. SDB-MIL also guarantees that each positive bag has at least one instance inside the sphere and all negative bags are outside the sphere.

Enclosing at least one instance from each positive bag in the sphere enables a more desirable MIL classifier when the negative data in the training set cannot sufficiently represent the distribution of negative data in the testing set. The single cluster hypothesis is reasonable in some applications but problematic in many other contexts. In image classification, the target concept can correspond to many clusters.

This assumption provided the starting point for the realization of MIL methods capable of learning multimodal positive concepts. SVM-based methods of space-instances such as *mi-SVM* [95] can handle disjoint regions of positive instances using a kernel. Two novel formulations of multiple-instance learning in terms of maximum margin problem are presented in [95]. These two extensions of the SVM learning approach lead to mixed integer quadratic programs that can be solved heuristically.

The proposed generalization of SVMs makes a state-of-the-art classification technique, including non-linear classification via kernels.

In [116], Amores pays special attention to vocabulary-based approaches: the empirical comparison includes seven databases from four heterogeneous domains, implementations of eight popular MIL methods, and a study of the behavior under synthetic conditions. Furthermore, the methods that model the distribution of instances in bags such as vocabulary-based methods naturally deal with data sets containing multiple concepts.

Multiple Instance Learning (MIL) has been used successfully in various applications, including image classification. Existing MIL methods, however, do not address the problem where positive instance distributions are multi-modal. This is a limitation for MIL models in many real world applications. In order to solve this problem, in [117], the authors propose a MIL novel discriminative data-dependent mixture-model method (MM-MIL) for image classification. MM-MIL addresses the multi-target problem through a data dependent mixture model, which allows positive instances to come from different clusters.

Furthermore, the kernelized representation of the proposed model allows effective and efficient learning in high dimensional feature space.

Non-representative negative distribution

In classic multiple instance learning (MIL), positive and negative bags are usually needed to learn a prediction function. In some studies, such as [99], it is pointed out that the learning of the concepts of instance assume that the distribution of the data is identical both in the testing and training phase. This assumption is satisfied for positive needs, but not for negative data where training data cannot always be dedicated to the distribution of the negative instance. If sufficient training data has been provided, the algorithm can learn by keeping the target object. However, some objects can be found in various contexts; in these cases it is impossible to entirely model the negative class distribution. Vice versa, in some applications the object of the research can find itself alone in contexts requirements and can be modeled using a finite number of samples. However, a high human cost is needed to know the label of each positive or negative bag.

Our lenses are contained only in positive bags, while negative bags contain noise or background. In this context it is reasonable not to invest too many resources to label negative bags. Surprisingly though, many existing MIL methods require enough negative bags in addition to the positive ones. For this reason, some methods limit themselves to modeling only the positive class, managing appropriately several negative distributions in the test phase.

In most cases, these methods look for a region that includes the positive concept. In APR [82] this region is a hyper-rectangle, while in many others it is a collection of hyper spheres [97], [118], [119]. These methods perform the classification based on the distance from a point (concept) or from a region in the space of the features: the determination of this point can coincide for example with the center of the instances, be they positive or negative, or be differently defined. Anything that is far enough from the point, or that lies outside the positive region, is considered negative. In this way, the form of the distribution of the negative class is not important.

Some non-parametric methods, such as Citation-kNN [120], exploit a similar approach by measuring the distance with respect to positive instances, instead of respect to positive concepts. The MIL problem can still be modeled as a problem of a class, in which the object

class of interest is defined by positive instances. Several methods using one-class SVM have been proposed. In particular, in [121], the authors propose an algorithm called “Positive Multiple Instance” (PMI), which learns a classifier given only a set of positive bags, which does not require the annotation of negative bags. PMI uses the assumption that the unknown positive instances in positive bags are similar to each other and form a compact cluster in the feature space such that negative instances are all outside of it. The experimental results show that PMI provides comparable performances, using a reduced number of training bags respect the ones requested by traditional MIL algorithms.

2.3.4 Label Ambiguity

The ambiguity of the label is an aspect that must be taken into consideration in the weak supervision. With the MIL approach, this ambiguity can take different forms depending on the assumption with which the problem is formulated. If the standard MIL assumption is adopted, there are no ambiguities on the labels of the instances extracted from the negative bags. If relaxed MIL hypotheses are adopted, there may be sources of ambiguity such as noise on labels and different label spaces for instances and bags.

Label noise

The MIL algorithms that adopt the standard assumption require that the bags are correctly labeled. Even a single negative instance that is close to the positive concept can affect the performance [122], as well as a bag mistakenly labeled as positive, would lead to a high False Positive Rate (FPR) [105].

For example, in artificial vision applications, an image can contain many objects and it could happen that it is not correctly labeled. Even in text classification there may be similar problems, perhaps due to the use of analogies. The methods that adopt the collective assumption can effectively managed label noise, as these methods assign the label exclusively verifying the presence of positive instances belonging to several classes. Another strategy to deal noise is to establish a threshold for positive classification, where the threshold value is based on the number of positive instances in bags. The method proposed in [123] uses both the threshold and the media strategies. First of all, the instances of a bag are classified from the most positive to the least positive, and the bags are represented by the average of the highest level instances and by the average of the lowest level instances.

Label spaces

In some formulations of MIL problems, instances and bags have not the same label space. Occasionally, these different spaces will correspond to different levels of detail. For example, a bag labeled as a diagnostic image of the abdomen will contain labeled instances related to the various organs. In other cases instead, the instance labels have not immediate semantic meanings; referring for example to a leopard-print dress, whose instances could be similar to those extracted from photos of real animals.

Methods that adopt the standard MIL assumption are unsuitable when the instances cannot be assigned to a specific class. Therefore, in these cases, it is necessary to use collective assumption by using vocabulary-based methods [116]. These MIL methods associate instances to words discovered from instance distribution. The bags are thus represented by distributions on these words.

All the peculiarities presented in this paragraph refer to a series of MIL problems, which must be specifically addressed. The choice of the best model for a given application is really complex.

2.4 MIL Paradigms

There are many contexts where problems can be effectively formulated using the multi-instance learning approach: pharmacy, text classification, image classification, speakers identification and bankruptcy prediction, to name a few. Starting from a training set of bags with known labels, the classification task in MIL consists in the prediction of the class label of unseen bags (see Figure 2.3).

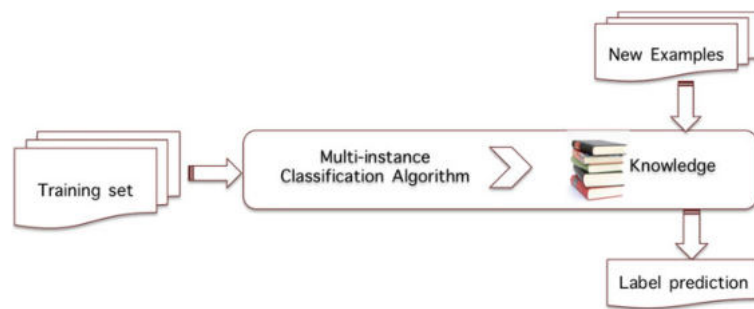


FIGURE 2.3: Procedure of MIL classification problem

The MIL algorithm exploits the training data to learn a classifier, which is subsequently used to predict the class label of new examples. Traditional single-instance classification represents each learning pattern with one instance in the form of a feature vector. Thus, in an image classification problem that tries to classify an image of the category zebra, the image would be represented by one instance as a feature vector. Each observation has an associated class label (label zebra or label no zebra). In the MIL representation each bag is represented with several instances or feature vectors (instances).

The bag as a whole has an associated label, but class labels of instances inside it are unknown. We know that if the bag label is zebra, at least one instance contains a zebra. However, it is not known which instance contains the learning object nor whether there is more than one instance containing it.

2.4.1 Taxonomy of MIL paradigms

In MIL terminology, a *bag* is a set $X = \{x_1, \dots, x_N\}$, where the elements x_i are feature vectors called *instances*, and the cardinality N can vary across the bags. All the instances x_i are vectors in a d -dimensional feature space, \mathbb{R}^d , called *instance space*. As we have introduced before, MIL classification problem aims at learning a model, at training time, that can be used to predict the class labels of unseen bags. Focusing on the binary classification problem, a bag X can be either positive or negative, and MIL model have to estimate a classification function $g(X) \in [0, 1]$ that provides the likelihood that X is positive.

In order to learn $g(X)$, MIL model uses a training set with M bags and their corresponding labels, $T = \{(X_1, y_1), \dots, (X_M, y_M)\}$, where $y_i \in \{-1, 1\}$ is the label of X_i and in particular, $y_i = -1$ if X_i is negative, and $y_i = 1$ if it is positive. Over the bag-level classification

function $g(X)$, many methods are focused on learning an instance-level classification function $f(x_i)$ that operates directly on the instances x_i .

Nowadays, the proposed MIL algorithms for classification problems cover all categories of machine learning methods. From decision rules and tree methods to the most sophisticated evolutionary methods or support vector machines, all of these algorithms have been reformulated according to a MIL approach. The great interest aroused by the MIL approaches for a series of application contexts, and the growing number of MIL algorithms has raised the need to create their own taxonomy that differentiates them according to distinctive features. As in the case of traditional single-instance learning, various proposals have been made to create an exhaustive and exclusive taxonomy. The taxonomy currently most followed is that proposed by Amores in [83]. This taxonomy, classifies the MIL methods for Classification (MIC) problems based on the way in which the existing information in the data is used, into the following three groups:

- **Instance Space (IS):** algorithms that determine the discriminating functions within the instance space. The bag label is derived using a multiple instance assumption which links the instance labels to that of the bag. In IS paradigm, a discriminating instance level classifier $f(x)$ is trained to allow the separation of instances in positive bags from those in negative ones. Considering a new bag X , the $g(X)$ bag classifier is obtained by aggregating the evaluations at the instance level $f(x)$, where $x \in X$. The Instance-Space paradigm does not look at global characteristics of the bag, exploiting local information through the use of instances.
- **Bag Space (BS):** methods that work in the bag space and define the similarity through distance measurements between bags, allowing them to determine the spatial relationships between bags and classes. In the BS paradigm, each bag X is treated as a whole entity and the learning process discriminates between bags. It follows that a discriminative bag-level classifier $g(X)$ uses the information of the entire bag X to establish the class of X . With the BS paradigm, the decision is made by analyzing the entire bag, rather than aggregating decisions at the local level. Since the bag space is non-vector, BS methods adopt non-vector learning techniques for which the definition of a distance function $D(X, Y)$ is provided. Through the use of distance $D(X, Y)$ it becomes possible to compare two bags X and Y namely two non-vectorial entities. The distance function $D(X, Y)$ can be used in any standard distance-based classifier such as K-Nearest Neighbor (K-NN) or in any kernel-based classifier like SVM.
- **Embedded Space (ES):** algorithms that transform the original input space into an embedded space, in which the bags are described by single-attribute vectors. Single instance algorithms can be applied in the induced space. In the ES paradigm, each bag X bag is mapped onto a single feature vector. The original bag space is mapped onto an vectorial embedded space, where the classifier is learned. In this way the original classification problem is transformed into a standard supervised learning problem, in which each feature vector has an associated label. Thus, it becomes possible to use any standard classifier like those seen in Section 1.5. The ES paradigm is also based on global information at the bag level; in fact the generic bag X is represented by a function vector v which contains all the information of interest of the bag. Given this

feature vector, it is worth for the bag-level classifier that $g(X) = f(v)$, where f is a discriminant classifier that operates on the vector v representing the entire bag.

Practically, the taxonomy proposed in [83] includes two main categories: instance-based methods and bag-based methods, in turn divided into two categories that differ in their use of an embedded space or not.

Basically, the taxonomy introduced by Amores classifies the methods based on whether they focus on information at the instance level (IS paradigm) or global information at the bag level and, differentiating in this latter case the methods that implicitly extract the relevant information (BS Paradigm) or those that extract information explicitly (ES paradigm). It is worth emphasizing that both ES and BS paradigms exploit global information at the bag level, but differ in the way information is extracted. In the BS paradigm, information is implicitly extracted by defining a distance function or kernel. In the ES paradigm, the extraction of information from the whole bag is carried out explicitly by defining a mapping function that represents the relevant information in a single vector v . In addition, there is also a computational cost characteristic for each paradigm.

2.5 Instance Space Models

The aim of this thesis is related to the classification problems of medical images using instance space Multiple Instance Learning optimization methods. In particular we refer to the instance space methods. In this section we report some of the literature models to which we have referred and inspired.

2.5.1 Support Vector Machines for Multiple Instance Learning

In [95], Andrews, Tsochantaridis and Hofmann presented an innovative classification technique, which lends itself through the kernel transforms to be used also in the non-linear classification. In particular, this innovative generalization of the SVM techniques has been presented in two multi-instance learning formulations as a maximum margin problem leading to mixed integer quadratic programs that can be heuristically solved.

The classic SVM method is modified and extended to allow the resolution of MIL problems. The first approach explicitly treats the labels of the model as unobserved integer variables, subject to the constraints defined by the (positive) labels of the bag. The aim becomes to maximize the *soft-margin*, namely the usual pattern margin, jointly on hidden label variables and a linear or kernelized discriminant function. The second introduces a generalization of the notion of margin to the bags of which it aims to maximize the margin. The first approach is particularly indicated when it is necessary to obtain an accurate classifier at the model level, the second one when it is necessary to classify new test bags. In the case of singleton bags, i.e. when the instance coincides with the entire bag, both methods are identical and refer to the standard SVM soft margin formulation.

The *mi-SVM* model

The first formulation of MIL presented in [95] is a mixed integer formulation known as *mi-SVM*.

Let J^+ be a set of m positive bags of instances and J^- be a set of k negative bags of instances:

$$\text{Positive Bags } J^+ = \{J_1^+, \dots, J_m^+\}$$

$$\text{Negative Bags } J^- = \{J_1^-, \dots, J_k^-\}$$

Let's indicate by $x_j \in \mathbb{R}^n$ the j -th instance belonging to a generic bag, both positive or negative. In classic *instance based SVM*, we would look for a hyperplane H , separating the instances belonging to the negative bags from those belonging to the positive ones:

$$H \triangleq \{x \in \mathbb{R}^n | w^T x + b = 0\}$$

More precisely, defining the couple of *shifted* hyperplanes $H^- \triangleq \{x | w^T x + b = -1\}$ and $H^+ \triangleq \{x | w^T x + b = 1\}$, we would require, for all instances belonging to negative bags and to positive bags, respectively:

$$w^T x_j + b \leq -1, \quad (2.5)$$

and

$$w^T x_j + b \geq 1, \quad (2.6)$$

Consequently, we would associate, to each couple hyperplane-instance, the classification error

$$\max\{0, 1 + (w^T x_j + b)\}, \quad (2.7)$$

if the instance belongs to a negative bag, and

$$\max\{0, 1 - (w^T x_j + b)\},$$

otherwise.

In practical applications such an approach appears restrictive whenever (e.g. in image classification) negative and positive bags exhibit a significant degree of similarity, which results in bag overlapping in the feature space.

In the approach proposed in [95], instead, we look for a hyperplane H such that:

i) all negative bags are contained in the set $S^- \triangleq \{x | w^T x + b \leq -1\}$;

ii) at least one instance of each positive bag belongs to the set $S^+ \triangleq \{x | w^T x + b \geq 1\}$.

Since it is impossible to know in advance whether or not such a hyperplane exists, an optimization model is introduced (see [95]), where the decision variables are the couple $(w \in \mathbb{R}^n, b \in \mathbb{R})$ defining the possibly separating hyperplane H and the labels $y_j \in \{-1, 1\}$ to be assigned to all instances of the positive bags.

The twofold objective consists of minimizing the classification error and of maximizing the separation margin, defined as the distance between the shifted hyperplanes H^- and H^+ . The model is the following:

$$mi\text{-SVM} \begin{cases} z^* = \min_{w,b,y} f(w,b,y) \\ \sum_{j \in J_i^+} \frac{y_j + 1}{2} \geq 1, \quad i = 1, \dots, m \\ y_j \in \{-1, 1\}, \quad j \in J_i^+, \quad i = 1, \dots, m, \end{cases} \quad (2.8)$$

where

$$f(w,b,y) \triangleq \frac{1}{2} \|w\|^2 + C \sum_{i=1}^k \sum_{j \in J_i^-} \max\{0, 1 + (w^T x_j + b)\} + C \sum_{i=1}^m \sum_{j \in J_i^+} \max\{0, 1 - y_j(w^T x_j + b)\}$$

with $C > 0$ representing the trade-off between the margin and the classification error objectives. A discussion on the structure of function f is in order. It is the sum of three terms:

1. $\frac{1}{2} \|w\|^2$. Minimization of the norm of w ([36]) leads to maximization of the margin, which in fact can be expressed as $\frac{2}{\|w\|}$.
2. $\sum_{i=1}^k \sum_{j \in J_i^-} \max\{0, 1 + (w^T x_j + b)\}$. This term (see (2.7)) represents the total classification error of the negative bags;
3. $\sum_{i=1}^m \sum_{j \in J_i^+} \max\{0, 1 - y_j(w^T x_j + b)\}$. This term represents the total classification error of the instances belonging to positive bags.

Note that for each such instance x_j satisfying (2.6), the error can be driven to zero by simply setting the corresponding $y_j = 1$. Note also that for those of such instances falling on the *wrong side*, that is satisfying condition (2.5), the error can be driven to zero as well by setting $y_j = -1$.

On the other hand constraints

$$\sum_{j \in J_i^+} \frac{y_j + 1}{2} \geq 1, \quad i = 1, \dots, m \quad (2.9)$$

impose that at least one instance of each positive bag is labelled by $y_j = 1$. Consequently if *all* instances of a positive bag fall in the complement to S^+ , then the error associated to such bag is strictly positive. Summing up, the classification error is equal to zero if and only if all negative bags are contained in the set S^- , at least one instance of each positive bag belongs to the set S^+ and no instance of any positive bag falls in the area where $|w^T x + b| < 1$.

In [95] problem P (2.8) has been tackled by means of two different heuristic techniques, based on solving successive SVM quadratic programs.

The mi-SVM formulation refers to mixed integer programming problem that meets the requirements of the MIL formulation; *mi-SVM* aims both to find the optimal labeling and to determine the optimal discriminant.

In *mi-SVM* formulation the y_i labels of the x_i belonging to the positive bags are treated as unknown integer variables. In this way, a soft margin criterion is maximized jointly on possible label and hyperplane assignments in *mi-SVM*.

In the formulation *mi-SVM* (2.8), the objective function f is non-smooth. Introducing the variables ξ_j , the model can be rewritten linearly as follows:

$$mi\text{-SVM} \begin{cases} z^* = \min_{w,b,y,\xi} f(w,b,y,\xi) \\ \xi_j \geq 1 - y_j(w^T x_j + b) \quad j \in J_i^+, \quad i = 1, \dots, m, \\ \xi_j \geq 1 + (w^T x_j + b) \quad j \in J_i^-, \quad i = 1, \dots, k, \\ \xi_j \geq 0 \quad j \in J_i^+, \quad i = 1, \dots, m, \\ \xi_j \geq 0 \quad j \in J_i^-, \quad i = 1, \dots, k, \\ \sum_{j \in J_i^+} \frac{y_j + 1}{2} \geq 1, \quad i = 1, \dots, m \\ y_j \in \{-1, 1\}, \quad j \in J_i^+, \quad i = 1, \dots, m, \end{cases} \quad (2.10)$$

where:

$$f(w,b,y,\xi) \triangleq \min_{w,b,y,\xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^k \sum_{j \in J_i^-} \xi_j + C \sum_{i=1}^m \sum_{j \in J_i^+} \xi_j.$$

The MI-SVM model

In [95] an alternative way to apply maximum margin ideas has been proposed following the MIL approach. In particular, the notion of margin has been extended from single patterns to sets of patterns.

For the mi-SVM formulation, the margin of each pattern in a positive bag is important, although it is possible to have the freedom in choosing the label variables aiming at margin maximization. In the bag-centered formulation, only one model for positive bag counts, since it will determine the margin of the bag. As introduced in the formulation of the previous model, let J^+ be a set of m positive bags of instances and J^- be a set of k negative bags of instances:

$$\text{Positive Bags } J^+ = \{J_1^+, \dots, J_m^+\}$$

$$\text{Negative Bags } J^- = \{J_1^-, \dots, J_k^-\}$$

The j -th instance belonging to a generic bag, both positive or negative is indicated by $x_j \in \mathbb{R}^n$. The goal is to determine a hyperplane H , separating the negative bags from positive ones:

$$H \triangleq \{x \in \mathbb{R}^n \mid w^T x + b = 0\}$$

The definition of a MIL version of the soft-margin classifier, using the notion of a *bag margin*, descends:

$$MI\text{-SVM} \begin{cases} \min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^k \max\{0, 1 + \max(w^T x_j + b)\} \\ + C \sum_{i=1}^m \max\{0, 1 - \max_{j \in J_i^+}(w^T x_j + b)\} \end{cases} \quad (2.11)$$

where k is the number of negative bags and m is the number of positive bags.

Some considerations should be emphasized in relation to errors on positive and negative bags.

The error on positive bags is evaluated through:

$$\max\{0, 1 - \max_{j \in J_i^+}(w^T x_j + b)\}$$

It follows that when the error occurs in the positive bags, it appears that:

$$1 - \max_{j \in J_i^+}(w^T x_j + b) > 0$$

that is:

$$\max_{j \in J_i^+}(w^T x_j + b) < 1$$

An error occurs on a positive bag if the hyperplane values in all points are < 1 , or if the max of these values is smaller than 1.

The error on negative bags is evaluated through:

$$\max\{0, 1 + \max_{j \in J_i^+}(w^T x_j + b)\}.$$

It follows that when the error occurs in the negative bags, it appears that:

$$1 + \max_{j \in J_i^+}(w^T x_j + b) > 0$$

that is:

$$\max_{j \in J_i^+}(w^T x_j + b) > -1$$

An error occurs on a negative bag if at least the value of the hyperplane at a point is greater than -1 , or if the max of the hyperplane values in all points of the bag is greater than -1 .

2.5.2 The Mangasarian and Wild Model

In the previous section we provided two different formulations of *mi-SVM* and *MI-SVM* models proposed by Andrews et al. [95]. These models actually extend the use of the SVM to MIL problems.

In *mi-SVM* they use integer variables to select the class of points in positive bags. On the contrary, with the formulation proposed in [124], continuous variables are introduced to represent the convex combination of each positive bag that is expected to be positioned on the positive side of the separation hyper-plan leading to the following optimization problem:

$$MICA \begin{cases} z^* = \min_{w, b, \lambda} f(w, b, \lambda) \\ \sum_{j \in J_i^+} \lambda_j^{(i)} = 1 \quad i = 1, \dots, m, \\ \lambda_j^{(i)} \geq 0 \quad j \in J_i^+, \quad i = 1, \dots, m, \end{cases} \quad (2.12)$$

where:

$$f(w, b, \lambda) \triangleq \min_{w, b, \lambda} \frac{1}{2} \|w\|_1 + C \sum_{i=1}^k \sum_{j \in J_i^-} \max\{0, 1 + w^T x_j + b\} + C \sum_{i=1}^m \max\{0, 1 - w^T (\sum_{j \in J_i^+} \lambda_j^{(i)} x_j) - b\}$$

In the above formulation, the objective function f is non-smooth. We linearize the classification error by introducing the variables v_i and ξ_i obtaining:

$$MICAL \left\{ \begin{array}{l} z^* = \min_{w, b, \lambda, \xi, v} f(w, b, \lambda, \xi, v) \\ v_i \geq 1 - w^T (\sum_{j \in J_i^+} \lambda_j^{(i)} x_j) - b, \quad i = 1, \dots, m, \\ \xi_j \geq 1 + w^T x_j + b \quad j \in J_i^- \quad i = 1, \dots, k, \\ \sum_{j \in J_i^+} \lambda_j^{(i)} = 1 \quad i = 1, \dots, m, \\ \lambda_j^{(i)} \geq 0 \quad j \in J_i^+, \quad i = 1, \dots, m, \\ v_i \geq 0 \quad i = 1, \dots, m, \\ \xi_i \geq 0 \quad i = 1, \dots, k \end{array} \right. \quad (2.13)$$

where:

$$f(w, b, \lambda, \xi, v) \triangleq \min_{w, b, \lambda, \xi, v} \frac{1}{2} \|w\|_1 + C \sum_{i=1}^k \sum_{j \in J_i^-} \xi_j + C \sum_{i=1}^m v_i$$

The model introduced by Mangasarian and Wild [124] is characterized by two fundamental points:

1. treats the positive bags in terms of instances convex hulls;
2. adopts the L_1 norm which in the case of classic SVM returns a linear programming problem instead of a quadratic problem, also providing a more sparse solution.

The first point implies that the coefficients of the instances convex hulls become variables that must be managed in the model. The adoption of the L_1 norm has positive implications in features selection applications, allowing to identify the significant features of the faced problem.

2.6 An overview of MIL literature

Many problems of interest lend themselves to being formulated as MIL problems: this is the reason why in the literature there is a great variety of MIL algorithms proposed by various scientific communities. Against this, there are few studies that carry out general MIL investigations. In this section, we will refer to a series of publications that refer to general characteristics of MIL problems or to specific contextualization related to the use of these models in the field of image analysis.

The first survey on MIL is a technical report written in 2004 [125]. In multi-instance learning, the training set includes labeled bags made up of unlabeled instances and the task is to provide the labels of unidentified bags. Starting from applications for Drug Activity Prediction, the developments on the study of learning, learning algorithms, applications

and extensions of multi-instance learning were examined. In particular, this document constitutes a first attempt at a unified vision in which multi-instance learning algorithms are examined.

One of the common incurred errors is to apply MIL algorithms being guided by the similarity of the data sets in the considered context, or by the size of the data. Any unobserved properties influence the performance of MIL algorithms: this is why some authors suggest comparing MIL with other supervised learning settings.

Ray and Craven [100] have found that supervised methods often produce better results, and that in general, algorithm performance depends on a number of factors.

There have been many proposals for Multiple Instance Learning related to content-based image retrieval (CBIR). The intended purpose is to classify all images using only a small data set with a label. Most MIL algorithms use images only for data test ignoring the use for learning process too.

In [126] the authors presented MISSL, a semi-supervised learning framework able to transform multiple instance problems into an input for a single-instance semi-supervised learning method graph-based. MISSL codified the multiple-instance aspects of the problem by operating both at the bag level and at the instance level. Unlike most prior MIL learning algorithms, MISSL made use of the unlabeled data, and showed potential to make use of the large amount of unlabeled data available for CBIR applications.

In [109], Bunescu et al., present a novel MIL approach useful when the positive bags contain sparse positive instances. The proposed approach enforced the constraint related to the assumption that at least one of the instances in a positive bag is positive showing effective accuracy when positive bags are sparse. Experimental results showed that the proposed approach overcomes the previous SVM methods on image data sets and behaves competitively on other types of MIL data.

In 2008 Babenko published a report [127] containing an updated survey on the main families of MIL methods and highlighted two types of ambiguity in the MIL problems. Supervised traditional learning requires a set of training data where the label is known for each data item. In many applications, it is an hard task to accurately assign labels to the inputs: the MIL learning paradigm enables the definition of a classifier from ambiguously labeled data. The ambiguities highlighted in this work concerned polymorphism, in which each instance is a distinct entity or a distinct version of an entity (i.e. conformations of a molecule), and the partial ambiguity in which all instances are parts of the same object (i.e. segments of an image). As in the case of supervision, the performance of various algorithms depends on the data sets and there is no algorithm that can be called to be the best in an absolute way.

Foulds and Frank [94] have reviewed the hypotheses on which the MIL algorithms are based; these hypotheses influence the functioning of the algorithms on different types of data sets. In particular, they discovered that the algorithms that work assuming the collective assumption work well also with data sets corresponding to the standard MIL assumption.

In [128], Babenko et al., faced the object tracking problem by an adaptive model. The aim of the proposed solution is to obtain a good performance at real-time speed, training a classifier in online mode to separate the object from the background through “tracking by detection” technique. This type of contribution will prove to be interesting, when, as in the context of medical imaging, it will be appropriate to isolate the lesion to be examined from the rest of the tissue.

The presented classifier uses the current tracker status to extract positive and negative examples from the current frame. Mistakes in the labeling phase can be caused by slight inaccuracies in the tracker: the adoption of MIL instead of traditional supervised learning soothes these aspects and makes the model more robust.

Further developing the themes presented in [128], Babenko et al., addressed the problem of tracking an object in a video given its location in the first frame and no other information using the just proposed tracking techniques called “tracking by detection” [129].

These methods train a discriminative classifier in an online manner to separate the object from the background. In particular, accurate experimental results were presented on a series of video clips using a new technique to update an adaptive model of the tracking system: use of MIL approach to train the appearance classifier insures more robust tracking respect to partial occlusions, and various appearance changes.

Works such as the one presented show further challenges to face. Adaptive appearance models cannot avoid the problems that appear when the object is completely occluded for a long time or the object is not present in the scene: in coherence with these situations an adaptive appearance model start learning from incorrect examples losing track of the object. Another challenge is to draw articulated objects that cannot be easily outlined except through a part-based approach, and again, a possible use of online algorithms for learning multiple instances could be useful to manage areas outside of visual.

Bergeron et al., presented a bundle algorithm for multiple-instance classification and ranking useful for many problems that have a special structure [130]. The functions of multiple instance loss are generally non-fluid and non-convex and are often addressed by converting them into non-convex optimization problems solved in an iterative way. Inspired by linear gradient-based methods for support vector machines, the authors optimized the target directly using a non-convex bundle method. The computational results showed that this method is linearly scalable, without sacrificing the accuracy of the generalization, also facilitating kernel modeling. This work present, for the first time, a linearly scalable algorithm to solve multi-instance learning problems using a non-convex non-regular bundle algorithm. Although its computational complexity has not been formally established, the MIL bundle showed to have a linear time scalability in the sample size.

As a result, MIL problems can be studied in more detail, allowing models to be created from larger samples, using more features and allowing more complex tasks, including feature selection. From this point of view, the authors highlight the opportunity inherent the possibility of using Big Data information in sensitive sectors such as computational chemistry and other applications.

Sabato and Tishby [131] analyzed MIL sample complexity discovering that the statistical performance of the MIL depends only slightly on the number of instances inside bag. In this work a unified theoretical analysis for MIL was provided, which applies to any underlying hypothesis class, regardless of a specific application or domain of the problem. This allows to think on applications that can take into consideration an appropriate number of instances for single image, and raises from the question that instances randomly taken can generate incorrect labels.

Other works focus on specific classes of algorithms and are related to specific application contexts. For classification task, many works analyze SVM solutions that provide better performance for the classification of instances or bags, depending on the properties of the method and the size of the data set [132]. The context of pattern analysis has attracted many

researchers, enticed by the possibility of implementing frameworks that can be used in sensitive contexts such as the medical one.

In [133] Xu et al., presented in 2014 an interesting overview on high-level tasks solved with a minimum of manual annotation showing good feature representations for medical images. More precisely, the experimental section refers to a dataset consisting of colon cancer histopathology images. In medical image analysis, objects like cells, organs and lesion are characterized by significant clinical features. Identifying the features of interest in a given application context is a very difficult task and requires skills in choice, in formulation and in software implementation of the selected features. With the aim of bypassing this arduous step, the authors study automatic extraction of feature representation through deep learning (DNN). Furthermore, detailed annotation of objects is often an ambiguous and challenging task: the use of MIL framework in classification training with deep learning features can mitigate this issue. Experiments on image features representation allow us to draw interesting conclusions [133]:

- (1) automatic features learning is better than the manual one;
- (2) the performance of unsupervised approach are not too different rather full supervised ones;
- (3) in supervised deep learning features, the MIL performance is better than the supervised performance.

For an effective operation, the proposed framework requires the availability of a lot of data. In fact, if the data set is not sufficiently numerous, the features that the model learns do not guarantee the best performances, resulting worse than solutions based on supervised learning. A series of skepticisms regarding the use of DL techniques remain open concerning the lack of clarity on the way the model works and on its selection criteria.

Alpaydin et al. [93] have carried out a study related to application of instance-space and bag-space classifiers on synthetic and real world data. It has been shown, that for data sets with few bags, it is preferable to use an instance space classifier, and that if the instances provide partial information on the bag labels, it is preferable to use a bag-space representation.

In [134], the similarities between reference datasets MIL were studied. As for the supervised classification, several classifiers and different data sets are available in the MIL context. A comparison of different MIL classifiers is possible only if the differences in the data sets used for the comparison are really understood. The authors of [134] provide an overview of the available reference datasets and of some popular MIL classifiers, using a measure of dissimilarity of the data set. The proposed measure is based on the differences between the ROC curves obtained from different classifiers. The obtained results show that data sets conceptually similar can behave very differently and vice versa. The indications that derive from this recommend an accurate analysis of the characteristics of the data set when comparisons are made between existing and new MIL classifiers.

Among the frameworks adopted for the analysis of medical images, computer assisted diagnosis system (CAD) are attracting interest. CADs often adopt supervised machine learning techniques, that on one hand are capable of providing good performance, but on the other hand involve a long-standing annotation phase by the experts. The localization of annotations on medical images invalidates the use of the entire image. The MIL techniques

lends itself well to setting these limits, allowing the bags to be labeled based on the maximum value of the labels of the instances contained in the same bag. Kandemir and Hamprecht in [135] use MIL techniques for evaluating their performance on two CAD applications, the first relating to the diagnosis of Barrett's cancer and the second concerning the diagnostic screening of diabetic retinopathy. From the experiments conducted, it emerged that the most accurate diagnosis is obtained using the same algorithm that works at bag-level. This result is interesting consider the very different visual outlook of the data set.

Similarly, for instance-level prediction, the algorithm that best performs is invariant between the two applications considered.

In 2016, Herrera et al. [136], published a dedicated book that aims to present an understandable overview of the MIL paradigm, providing a formal definition and dealing with the sub-paradigms of the proposed approach with reference to the most relevant algorithms and the most interesting applications.

Wei e Zhou [137] studied the utility of several image bag generators, among which, k-meansSeg, Blobworld, WavSeg, JSEG-bag and SIFT. Thought experiments bag generators look to better perform against other strategies. For image classification problems, the standard MIL assumption of learning algorithms is not efficient. Vice versa the methods that use the collective assumption work better for image classification task. Always for image analysis, the authors found that modeling intra-bag similarities was a good strategy for bag classification.

Recently, Quéllec et al. [88] have written a review on MIL methods focused on medical images and videos analysis. For this particular context, MIL approach has proved to be particularly interesting overcoming the performance of single-instance learning algorithms. Based exclusively on class labels globally assigned to images or videos, MIL algorithms detect relevant patterns locally in images or videos, which may be used for classification at global level. Because supervision occurs considering global labels, manual segmentation is superfluous to train MIL algorithms, unlike what happens in traditional single-instance learning (SIL). Also in quite [88], existing strategies for modeling medical imaging problems are examined as MIL problems, illustrating the structure of algorithms useful for various applications. Experiments carried out on medical image and video data sets show that, in addition to being cheaper than SIL solutions, MIL algorithms are also more accurate in many cases. In other words, MIL is a great opportunity for many medical image and video analysis tasks.

The abundant availability of digital images has emphasized the increase in demand for their analysis, and has given impetus to frameworks such as computer-aided diagnosis systems that use machine learning techniques.

In [138], Komura and Ishikawa address some specific problems related to the applications of digital analysis of pathological images with machine learning algorithms. In particular, the authors emphasize how the recognition of digital histopathological images can be effectively addressed through machine learning. Some problems are highlighted, such as the presence of foreign bodies in the images, or the presence of rare tumors with respect to which the classifier has not been trained: the classification of this type of images in one of the predefined categories thus leads to an inaccurate diagnosis. To solve the problem, the scientific community is interested in algorithms for the outliers detection, such as principal components analysis (PCA) and some methods based on deep learning techniques that use reconstruction errors. The use of deep learning in the medical field is accompanied by a

lot of skepticism. This reservation depends on whether its decision-making process is not understandable and looks like a black box. Even the decision-making process of the specialist is not always clear and the diagnoses and the therapies are often influenced by his experience: in this case it is the patient who would like greater clarity on the identification and management of his own pathology. Topics such as joint learning could enable a better osmosis between automatic analysis and human expertise.

In [139], Cheplygina et al., present an overview of semi-supervised, multiple instance, and transfer learning in medical imaging. The strong impact of machine learning algorithms in health care areas, and the greater availability of medical images, highlight a new challenge for supervised ML algorithms regarding their effective functioning with unlabeled data. As a result, various methods that can learn with less/other types of supervision, have been proposed.

Esteva et al.[140], present a guide on deep-learning techniques applied in computer vision, natural language processing, and reinforcement learning. In particular, the authors describe how these computational techniques can impact a few key areas of medicine showing how to design end-to-end systems. The discussion of computer vision focuses largely on medical imaging, and the described applications of natural language processing refer to domains such as electronic health record data. Similarly, reinforcement learning is discussed in the context of robotic-assisted surgery, and generalized deep-learning methods for genomics are reviewed.

2.7 Take away

The studies we have referred to in this chapter can be summarized as follows:

- Some MIL problems can also be solved using standard supervised methods.
- In the modeling phase, when the number of bags is low, the use of an instance-based method is advisable.
- When it is necessary to model the combinations of instances to infer the labels of the bags, the methods of bag space and inclusion work better.
- Generally the best classifier at bag level is not the best classifier at the instance level and vice-versa.
- The similarity between the instances of the same bag affects classification performance.
- The performance of MIL models depends only slightly on the number of instances per bag but depends by several properties of the data set.
- Multiple Instance Learning approaches are particularly effective for Medical Image and Video Analysis.

All these indications are related to one or more characteristics typical of the MIL problems. Identifying these characteristics, by gaining a better understanding of their impact on algorithm performance, is an important step towards the advancement of MIL research and towards a more aware adoption of suitable methods for specific application contexts.

PART II

THE ADVANCES

Chapter 3

Classification via spherical separation

"Est quoque cunctarum novitas carissima rerum"

[Novelty is the most welcome among all things]

– Ovid, letters from Pontus, 3, 4, 5

The problem of binary classification is aimed at discriminating between two finite sets of points in the n -dimensional space by using a given separation surface that has to minimize the classification error.

Among the different mathematical approaches proposed for binary classification, the support vector machine technique (SVM) [36], has been and still is one of the most widely used. As we have discussed in Section 1.5.1, with this technique a hyperplane is identified in order to separate the points of the two considered classes. It is also possible to obtain non-linear separation surfaces by adopting kernel transformations, thus mapping the data into larger dimensional instance space.

In some application contexts, classic separation scheme is too demanding and, consequently, correct classification is hard to be achieved. Suppose we want to categorize a given set of medical images into two classes, assuming that all of them are related to the same type of organ or of a tissue, in presence/absence of a certain pathology. In this case both the positive and negative images are related to the same organ type, thus they are likely to exhibit a relevant similarity degree.

New methods involving non-linear classification surface and inspired by the SVM have been introduced. First of all it is necessary to remember the Support Vector Domain Description (SVDD) [141]. This method uses a particularly effective minimum volume sphere as separation surface useful for applications concerning the detection of novelties or anomalies. In particular, with this model it is possible to obtain descriptions of the higher order limit without additional calculation costs. SVDD, through the use of various kernels, allows flexible and accurate data descriptions.

Another very interesting model that uses separation by means of a sphere is the one proposed in [142], in which the center of the sphere is fixed. Starting from this simple assumption, this model, under the hypothesis of a careful choice of the center of the sphere, allows good separation results. For these reasons, the model proposed in [142] can be profitably adopted in the management of very large data sets, and is still suitable for modern mobile applications.

Spherical classification is also used for bag classification. In particular, in [143] the authors face the problem of finding a sphere such that negative bags are *entirely* left outside it, while at least one instance of each positive bag is kept inside the sphere. The authors leverage on the combinatorial nature of the problem. In fact, by picking exactly one instance from each positive bag, the problem reduces to finding a spherical separation where all such instances stay inside the sphere, while the negative bags are confined outside. Of course the number of spherical separation problems to be solved depends, in principle, on the number of possible choices of a single instance for each positive bag.

The concept of separation margin is used in classification through spherical separation. In [144], it is proved that a full enumeration algorithm is polynomial in the dimension of the feature space. However, for dealing with large size problem, a Variable Neighborhood Search (VNS) metaheuristic is designed.

In this chapter we will focus on a new MIL spherical approach. To this aim, in the next section, we first recall the concept of spherical separation adopted in the supervised learning.

In the rest of the chapter we indicate by $\|\cdot\|$ the Euclidean ℓ_2 -norm and by $a^\top b$ the inner product of the two vectors $a, b \in \mathbb{R}^n$. Moreover, we denote by

$$S(w, r) \triangleq \left\{ x \in \mathbb{R}^n, \|x - w\|^2 = r^2 \right\}$$

a sphere in \mathbb{R}^n with center $w \in \mathbb{R}^n$ and radius $r \in \mathbb{R}$.

3.1 Spherical models for classification problems

Let A and B be two non-empty and disjoint finite sets of sample points in the n -dimensional space having m and k dimensions respectively.

The problem of classifying with spherical separators has been defined in [142], as the problem of identifying a sphere of minimum volume that encloses all the points of A and no point of B . Both the center of the sphere belonging to \mathbb{R}^n and the radius belonging to \mathbb{R} , must therefore be selected.

The problem of the separation of the set A from the set B of points can be tackled by searching for a sphere of minimum volume. Since the sphere that perfectly separates A and B may not exist, then a sphere is sought which minimizes the classification error.

The sets A and B will be separated by the sphere $S(w, r)$ if:

$$\begin{cases} (a_i - w)^\top (a_i - w) \leq r^2 \quad \forall a_i \in A \quad (i = 1, \dots, m) \\ (b_l - w)^\top (b_l - w) \geq r^2 \quad \forall b_l \in B \quad (l = 1, \dots, k) \end{cases} \quad (3.1)$$

It is possible to define the classification error associated to the decision variables (w, r) for any $a_i \in A$ and for $b_l \in B$, as follow:

$$\begin{cases} \mathcal{E}_i^- = \max\{0, (a_i - w)^\top (a_i - w) - r^2\} \quad \forall i = 1, \dots, m \\ \mathcal{E}_l^+ = \max\{0, r^2 - (b_l - w)^\top (b_l - w)\} \quad \forall l = 1, \dots, k \end{cases} \quad (3.2)$$

The problem of minimizing both the volume of the sphere and the classification error can be formulated as follows:

$$\min_{w,r} r^2 + C \sum_{i=1}^m \max\{0, (a_i - w)^T(a_i - w) - r^2\} + C \sum_{l=1}^k \max\{0, r^2 - (b_l - w)^T(b_l - w)\} \quad (3.3)$$

where C is a positive constant used to balance the two objectives.

The problem just formulated requires the minimization of a non-smooth and non-convex function and is represented by the sum of several functions of the maximum type, apart from the smooth quadratic term r^2 . To eliminate non-smoothness, the additional variables \mathcal{E}_i^- , \mathcal{E}_i^+ are introduced. In this way it is possible to transform the unconstrained problem into a constrained optimization problem:

$$\begin{cases} \min_{w,r,\mathcal{E}_i^-, \mathcal{E}_i^+} r^2 + C \left(\sum_{i=1}^m \mathcal{E}_i^- + \sum_{l=1}^k \mathcal{E}_l^+ \right) \\ r^2 - (a_i - w)^T(a_i - w) + \mathcal{E}_i^- \geq 0 \quad \forall i = 1, \dots, m \\ (b_l - w)^T(b_l - w) - r^2 + \mathcal{E}_l^+ \geq 0 \quad \forall l = 1, \dots, k \\ \mathcal{E}_i^- \geq 0 \quad \forall i = 1, \dots, m \\ \mathcal{E}_l^+ \leq 0 \quad \forall l = 1, \dots, k \end{cases} \quad (3.4)$$

In the particular case where the center of the sphere is fixed, the resolution of the described problem implies only the minimization of the radius. A minimization problem is thus obtained where the single-variable of objective function is non-smooth and convex.

In [142], the authors proposed a fixed-center spherical separation algorithm with kernel transformations for classification problems.

By fixing the center of the sphere a simplification of the problem (3.4) is obtained. In particular, the center of the sphere w is assumed equal to some centroid of the set A . By the change of variable $z = r^2$ where $z \geq 0$, and through the definition of the following variables c_i and d_l

$$\begin{cases} c_i \triangleq (a_i - w)^T(a_i - w) \leq 0 \quad \forall i = 1, \dots, m \\ d_l \triangleq (b_l - w)^T(b_l - w) \geq 0 \quad \forall l = 1, \dots, k \end{cases} \quad (3.5)$$

the problem (3.3) can be written as follows:

$$\min_{z \geq 0} + C \left(\sum_{i=1}^m \max\{0, (c_i - z)\} + \sum_{l=1}^k \max\{0, (z - d_l)\} \right) \quad (3.6)$$

which is a convex, piecewise affine minimization problem in the scalar variable z , where z is non negative. The problem (3.6) thus obtained can be solved through the usual techniques of univariate minimization. The authors in [142] presented an efficient algorithm that provides an exact solution in the time $O(p \cdot \log p)$, where $p = \max\{m, k\}$.

3.2 Multiple instance classification via spherical separation

In this section we present an original contribution concerning spherical classification for Multiple Instance Learning [145].

We adopt spherical separation as a classification tool and come out with an optimization model which is of DC (Difference of Convex) type. We tackle the model by resorting to a specialized non-smooth optimization algorithm, recently proposed in the literature which is based on objective function linearization and bundling. The results obtained by applying the proposed approach to some benchmark test problems are also reported.

In this section, although we adopt for MIL a spherical separation paradigm similar to that introduced in [143], we state the problem in a quite different way by introducing a non-smooth non-convex continuous optimization formulation. In particular we define a classification error function depending on center and radius of the sphere and we come out with an optimization model to minimize a combination of the volume of the sphere and of the classification error. We provide a DC (Difference of Convex) decomposition of the objective function which allows us to resort to a number of effective algorithms available in the literature, see [146]–[150].

The continuation of the chapter is organized as follows. In subsection 3.2.1 we introduce our spherical separation model and in subsection 3.2.2 we describe the DC decomposition of the objective function of the related optimization problem. Some explanations of the algorithm adopted for solving the DC optimization problem are in subsection 3.2.3, while in the subsection 3.2.4 we recall a brief description of the data sets used to evaluate the performance of our proposed algorithm. Finally, the numerical results we have obtained are discussed in subsection 3.2.5.

3.2.1 Problem statement

We assume that a set of instances $X = \{x_1, \dots, x_N\}$ is given in the sample space \mathbb{R}^n , which is partitioned into $m + k$ subsets $X_1^+, \dots, X_m^+, X_1^-, \dots, X_k^- \subset X$, named *bags*. Hence, each bag is constituted by a set of instances (i.e., points in the sample space \mathbb{R}^n), and each instance belongs to exactly one bag. The subsets X_1^+, \dots, X_m^+ are referred to as the *positive bags*, while X_1^-, \dots, X_k^- are referred to as the *negative bags*. We denote by J_1^+, \dots, J_m^+ the instance index-sets of the positive bags X_1^+, \dots, X_m^+ , by J_1^-, \dots, J_k^- the instance index-sets of the negative bags X_1^-, \dots, X_k^- , and we let

$$J^+ \triangleq \{J_1^+, \dots, J_m^+\} \quad \text{and} \quad J^- \triangleq \{J_1^-, \dots, J_k^-\}.$$

As stated earlier, our aim is to find a sphere $S(w, r) \subset \mathbb{R}^n$, of center $w \in \mathbb{R}^n$ and radius $r \in \mathbb{R}$, separating the two classes of bags. In the following definition we state that in order to separate the positive bags X_1^+, \dots, X_m^+ from the negative ones X_1^-, \dots, X_k^- , a sphere must have a nonempty intersection with each positive bag, while leaving outside all the instances belonging to negative bags.

Definition 3.2.1. Let a sphere $S(w, r)$ of center $w \in \mathbb{R}^n$ and radius $r \in \mathbb{R}$ be given. $S(w, r)$ is called a *separating sphere* if for every $i \in \{1, \dots, m\}$ it holds:

$$\|x_j - w\|^2 - r^2 \leq 0 \quad \text{for some } j \in J_i^+ \quad (3.7)$$

and for every $i \in \{1, \dots, k\}$ it holds:

$$\|x_j - w\|^2 - r^2 \geq 0 \quad \text{for every } j \in J_i^-. \quad (3.8)$$

A pictorial example of spherical separation is presented in Figure 3.1, where the sphere $S(w, r)$ separates the negative bags X_1^- , X_2^- , and X_3^- from the positive bags X_1^+ and X_2^+ . In particular, we remark that while the bags depicted in Figure 3.1 are spherically separable, they are not separable by any hyperplane in the sense of [95], [96], [130], the instances belonging to the positive bag X_2^+ being inside the convex hull of the instances belonging to all the negative bags.

According to Definition 3.2.1 any negative bag X_i^- , with $i \in \{1, \dots, k\}$, is said *misclassified* with respect to a given sphere $S(w, r)$ if there exists $j \in J_i^-$ such that $r^2 - \|x_j - w\|^2 > 0$. Likewise, any positive bag X_i^+ , with $i \in \{1, \dots, m\}$ is said *misclassified* with respect to $S(w, r)$ if $\|x_j - w\|^2 - r^2 > 0$ for every $j \in J_i^+$.

Based on the latter remark we introduce an optimization model whose aim is to look for a separating sphere, if any, by minimizing a measure of all the classification errors of both the negative and the positive bags.

In fact, with respect to the decision-variable vector $(w, r) \in \mathbb{R}^{n+1}$ and the related sphere $S(w, r)$, we define for every negative bag X_i^- , with $i \in \{1, \dots, k\}$, the classification error $\mathcal{E}_i^-(w, r)$ as

$$\mathcal{E}_i^-(w, r) \triangleq \max \left\{ 0, \max_{j \in J_i^-} \left\{ r^2 - \|x_j - w\|^2 \right\} \right\},$$

and for every positive bag X_i^+ , with $i \in \{1, \dots, m\}$, the classification error $\mathcal{E}_i^+(w, r)$ as

$$\mathcal{E}_i^+(w, r) \triangleq \max \left\{ 0, \min_{j \in J_i^+} \left\{ \|x_j - w\|^2 - r^2 \right\} \right\}.$$

Putting together the classification errors of all the positive and negative bags, we obtain the following spherical MIL error function $\mathcal{E}(w, r)$

$$\mathcal{E}(w, r) = \sum_{i=1}^k \mathcal{E}_i^-(w, r) + \sum_{i=1}^m \mathcal{E}_i^+(w, r), \quad (3.9)$$

and we note that $\mathcal{E}(w, r) \geq 0$, where $\mathcal{E}(w, r) = 0$ if and only if $S(w, r)$ is a separating sphere according to Definition 3.2.1.

We are ready now to define a Spherical MIL problem (SMIL) as the following unconstrained optimization problem

$$\min_{(w, r) \in \mathbb{R}^{n+1}} f(w, r) \triangleq r^2 + C\mathcal{E}(w, r), \quad (3.10)$$

which combines, by introducing a trade-off parameter $C > 0$, the two objectives of minimizing the radius of the sphere and the classification errors of all the negative and positive bags. Here the radius minimization is aimed at reducing the false positive phenomenon when the calculated sphere is used as a classification tool. We remark that the error function $\mathcal{E}(w, r)$ is inherently nonconvex and nonsmooth. In the next section we introduce a decomposition of $f(w, r)$ as the difference of two convex nonsmooth functions that will allow us to tackle the problem by adopting nonsmooth DC algorithms of the type described in [148].

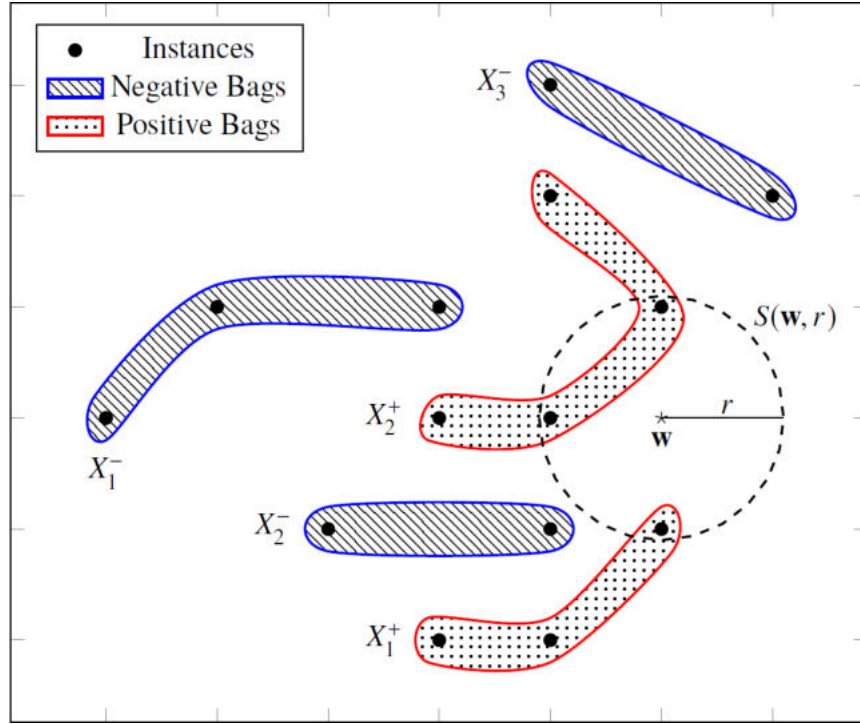


FIGURE 3.1: Spherical separation with three negative bags and two positive bags

3.2.2 A DC decomposition of SMIL

We first recall that any nonconvex function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is DC if it can be put in the form

$$f(y) = f_1(y) - f_2(y), \quad (3.11)$$

where both $f_1, f_2 : \mathbb{R}^d \rightarrow \mathbb{R}$ are convex. We assume that f_1 and f_2 are finite in \mathbb{R}^d and not necessarily differentiable.

In order to obtain a DC decomposition of the objective function of the SMIL problem (3.10), we focus on appropriately rewriting some of the terms appearing in the definition of the error function $\mathcal{E}(w, r)$. Focusing first on the definition of $\mathcal{E}_i^-(w, r)$ we obtain the following expression of the innermost maximum term

$$\begin{aligned} \max_{j \in J_i^-} \{r^2 - \|x_j - w\|^2\} &= \max_{j \in J_i^-} \left\{ r^2 + \sum_{l \in J_i^- \setminus \{j\}} \|x_l - w\|^2 - \sum_{l \in J_i^-} \|x_l - w\|^2 \right\} = \\ &= \max_{j \in J_i^-} \left\{ r^2 + \sum_{l \in J_i^- \setminus \{j\}} \|x_l - w\|^2 \right\} - \sum_{j \in J_i^-} \|x_j - w\|^2. \end{aligned} \quad (3.12)$$

Next, focusing on the innermost maximum term in the definition of $\mathcal{E}_i^+(w, r)$, we obtain that:

$$\begin{aligned}
\min_{j \in J_i^+} \{ \|x_j - w\|^2 - r^2 \} &= - \max_{j \in J_i^+} \left\{ r^2 - \|x_j - w\|^2 \right\} = \\
&= \sum_{j \in J_i^+} \|x_j - w\|^2 - \max_{j \in J_i^+} \left\{ r^2 + \sum_{l \in J_i^+ \setminus \{j\}} \|x_l - w\|^2 \right\}. \quad (3.13)
\end{aligned}$$

Now, by recalling that, for any given couple of numbers a and b , it holds

$$\max\{0, a - b\} = \max\{a, b\} - b, \quad (3.14)$$

and taking into account equation (3.12), we can rewrite, for every $i \in \{1, \dots, k\}$, the classification error $\mathcal{E}_i^-(w, r)$ as follows

$$\begin{aligned}
\mathcal{E}_i^-(w, r) &= \max \left\{ 0, \max_{j \in J_i^-} \left\{ r^2 + \sum_{l \in J_i^- \setminus \{j\}} \|x_l - w\|^2 \right\} - \sum_{j \in J_i^-} \|x_j - w\|^2 \right\} = \\
&= \max \left\{ \sum_{j \in J_i^-} \|x_j - w\|^2, \max_{j \in J_i^-} \left\{ r^2 + \sum_{l \in J_i^- \setminus \{j\}} \|x_l - w\|^2 \right\} \right\} + \\
&\quad - \sum_{j \in J_i^-} \|x_j - w\|^2. \quad (3.15)
\end{aligned}$$

Similarly, taking into account equation (3.13), we can rewrite, for every $i \in \{1, \dots, m\}$, the classification error $\mathcal{E}_i^+(w, r)$ as

$$\begin{aligned}
\mathcal{E}_i^+(w, r) &= \max \left\{ 0, \sum_{j \in J_i^+} \|x_j - w\|^2 - \max_{j \in J_i^+} \left\{ r^2 + \sum_{l \in J_i^+ \setminus \{j\}} \|x_l - w\|^2 \right\} \right\} = \\
&= \max \left\{ \max_{j \in J_i^+} \left\{ r^2 + \sum_{l \in J_i^+ \setminus \{j\}} \|x_l - w\|^2 \right\}, \sum_{j \in J_i^+} \|x_j - w\|^2 \right\} + \\
&\quad - \max_{j \in J_i^+} \left\{ r^2 + \sum_{l \in J_i^+ \setminus \{j\}} \|x_l - w\|^2 \right\}. \quad (3.16)
\end{aligned}$$

Finally, we observe that now both $\mathcal{E}_i^-(w, r)$, for every $i \in \{1, \dots, k\}$, and $\mathcal{E}_i^+(w, r)$, for $i \in \{1, \dots, m\}$, are expressed as the difference of two convex nonsmooth functions.

Summing up, a DC decomposition of the classification error function $\mathcal{E}(w, r)$ in (3.9) can be easily obtained by setting

$$\mathcal{E}(w, r) = \check{\mathcal{E}}(w, r) - \hat{\mathcal{E}}(w, r) \quad (3.17)$$

where the functions

$$\begin{aligned} \check{\mathcal{E}}(w, r) \triangleq & \sum_{i=1}^m \max \left\{ \sum_{j \in J_i^+} \|x_j - w\|^2, \max_{j \in J_i^+} \left\{ r^2 + \sum_{l \in J_i^+ \setminus \{j\}} \|x_l - w\|^2 \right\} \right\} + \\ & + \sum_{i=1}^k \max \left\{ \sum_{j \in J_i^-} \|x_j - w\|^2, \max_{j \in J_i^-} \left\{ r^2 + \sum_{l \in J_i^- \setminus \{j\}} \|x_l - w\|^2 \right\} \right\} \end{aligned} \quad (3.18)$$

$$\hat{\mathcal{E}}(w, r) \triangleq \sum_{i=1}^m \max_{j \in J_i^+} \left\{ r^2 + \sum_{l \in J_i^+ \setminus \{j\}} \|x_l - w\|^2 \right\} + \sum_{i=1}^k \sum_{j \in J_i^-} \|x_j - w\|^2 \quad (3.19)$$

are both convex. As a consequence, by letting

$$f_1(w, r) \triangleq r^2 + C\check{\mathcal{E}}(w, r)$$

and

$$f_2(w, r) \triangleq C\hat{\mathcal{E}}(w, r),$$

and observing that $f_1, f_2 : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$ are convex nonsmooth functions, then the nonconvex nonsmooth optimization problem SMIL presented in (3.10) can be reformulated as the following unconstrained nonsmooth DC program (DC-SMIL)

$$\min_{(w, r) \in \mathbb{R}^{n+1}} f(w, r) \triangleq f_1(w, r) - f_2(w, r). \quad (3.20)$$

3.2.3 Solving the DC-SMIL model

In the last decades nonsmooth DC functions (3.11) have been a subject of intense research activities both on the theoretical and the algorithmic side. We mention here the survey [151] and the seminal papers [152] and [153] where necessary and sufficient conditions for local and global optimality were established in a general nonsmooth setting, based on properties of the subdifferential and of the ϵ -subdifferential of the two convex functions f_1 and f_2 . We recall, in passing, that for a convex function $f : \mathbb{R}^d \rightarrow \mathbb{R}$, given $\bar{y} \in \mathbb{R}^d$ and $\epsilon > 0$, the subdifferential $\partial f(\bar{y})$ and the ϵ -subdifferential $\partial_\epsilon f(\bar{y})$ at \bar{y} are defined, respectively, as

$$\begin{aligned} \partial f(\bar{y}) & \triangleq \{g \in \mathbb{R}^d : f(y) \geq f(\bar{y}) + g^\top(\bar{y} - y), \forall y \in \mathbb{R}^d\}, \\ \partial_\epsilon f(\bar{y}) & \triangleq \{g \in \mathbb{R}^d : f(y) \geq f(\bar{y}) + g^\top(\bar{y} - y) - \epsilon, \forall y \in \mathbb{R}^d\}. \end{aligned}$$

As for practical applications of DC programming we mention here as examples [57], [154].

From the algorithmic point of view a relevant contribution was provided by the methods based on the linearization of function f_2 (see, e.g., the DCA method presented in [155] and references therein), where the problem is tackled via successive convexifications of function f . The basic idea behind such methods is the following. Taking the current estimate y_t of a (local) minimum of f in an iterative descent procedure, the following approximation $\tilde{f}_t(y)$ of f is built

$$\tilde{f}_t(y) = f_1(y) - (f_2(y_t) + g_t^{(2)\top}(y - y_t)), \quad (3.21)$$

where $g_t^{(2)} \in \partial f_2(y_t)$.

The model function $\tilde{f}_t(y)$, which is based on replacing f_2 by its linearization rooted at y_t , enjoys the following properties:

- i) $\tilde{f}_t(y)$ is convex;
- ii) $\tilde{f}_t(y) \geq f(y), \forall y \in \mathbb{R}^d$.

The next iterate y_{t+1} is then constructed by minimizing the convex function $\tilde{f}_t(y)$ and letting

$$y_{t+1} = \arg \min_{y \in \mathbb{R}^d} \tilde{f}_t(y).$$

Note that, in case it is $\tilde{f}_t(y_{t+1}) < \tilde{f}_t(y_t)$, from ii) it follows

$$f(y_{t+1}) \leq \tilde{f}_t(y_{t+1}) < \tilde{f}_t(y_t) = f(y_t),$$

that is a decrease in the model function guarantees decrease in the objective function as well.

Some approaches recently introduced to solving nonsmooth DC programs, see [147]–[149], retain the basic feature of DCA of keeping an affine (or piecewise affine) approximation of f_2 , but they replace function f_1 in (3.21) by its cutting plane approximation [156]. Moreover, a proximity term, borrowed from the well established class of bundle methods [157], is added to the model function.

We recall that the cutting plane model $h_t(\cdot)$ of any convex function $h : \mathbb{R}^d \rightarrow \mathbb{R}$ is defined, starting from any finite set of couples $\{(y_s, g_s) : s = 1, \dots, t\}$, where $y_s \in \mathbb{R}^d$ and $g_s \in \partial h(y_s)$, by letting

$$h_t(y) = \max \{h(y_s) + g_s^\top (y - y_s) : j = 1, \dots, k\},$$

and obtaining the next iterate $y_{t+1} = \arg \min_{y \in \mathbb{R}^d} h_t(y)$.

To validate our MIL method we have adopted the DC Piecewise-Concave Algorithm (DCPCA) for minimizing a nonsmooth DC function introduced in [148]. DCPCA can be sketched as follows:

- Two cutting plane models for f_1 and f_2 , respectively, are kept and updated throughout an iterative procedure;
- The next iterate is obtained by line-searching along a direction provided by the solution of one strictly-convex quadratic program, which embeds the two models and, for numerical stability purposes, also a quadratic *proximity* term. Under appropriate conditions a second auxiliary quadratic program is solved too, aiming at improving the approximation of f provided by the model;
- Whenever no descent is achieved along the search direction, a better approximation of f_1 close to the current estimate of the solution becomes available, this being the fundamental feature of any cutting-plane algorithm.

For the many technicalities of Algorithm DCPCA the interested reader is referred to [148].

3.2.4 Data sets

We have performed experiments on various data sets to evaluate the proposed technique and to compare it to other methods for MIL.

Musk Data Set

The MUSK data sets are the benchmark data sets used for testing in virtually all previous MIL approaches and have been described in detail in [82]. Both data sets, *Musk1* and *Musk2*, consist of descriptions of molecules that use multiple low-energy conformations. In these data sets each conformation deriving from surface properties is represented by a vector of 166 dimensional features. More precisely, *Musk1* contains on average about 6 conformations for each molecule, while *Musk2* has on average more than 60 conformations inside each bag.

These data sets are used to test MIL methods on the prediction task of new molecules in relation to the fact they will be musk or non-musk. The 166 features through which molecules are described depend on the exact shape and conformation of the molecule. By virtue of the rotation that can affect the bonds, each single molecule can present itself in different forms. The generation of these data sets was obtained by taking into consideration the low-energy conformations of the molecules suitably filtered to eliminate very similar conformations between them.

In more detail, the data set *Musk1* describes a set of 92 molecules evaluated by specialists including 47 musk and 45 non-musk molecules. This left 476 conformations.

The *Musk2* data set describes instead a set of 102 molecules always evaluated by specialists of which 39 musk and 63 non-musk. For Musk 2 all the low energy consumption conformations of the molecules were generated, obtaining 6.598 conformations.

Function vectors describing each conformation have been extracted for both Musk 1 and Musk 2. During the training of a classifier on these data sets, the classifier will indicate the *musk* class for a generic molecule if any of its conformations is classified as musk, viceversa a molecule will be classified as *non musk*.

Elephant, Fox and Tiger

One of the most important problems in computer vision is retrieving images from large data set using the image content as a search criterion. In [95] a new MIL data set was generated for an image annotation task. This data set was derived from original data consisting of color images taken from Corel data set. In the seminal work [158] the Blobworld system was presented which represents images by applying a transformation from raw pixel data to a subset of image regions characterized by similar color and texture. Using the pre-processing and segmentation techniques described in [158], in [95] each image was decomposed into a set of segments characterized by color, texture and shape descriptors.

The three categories *Elephant*, *Fox*, *Tiger* were randomly extracted from a pool of photos of other animals, each of which characterized by 100 positive and 100 negative example images.

The reduced accuracy of image segmentation, the small number of region descriptors and the small size of the training set, make these data sets very difficult due to a classification problem. This justifies why these categories are generally used to test MIL classification algorithms.

Trec Data set

The task to be faced is related to the identification of documents relevant for user's needs. The reference MIL data sets for text categorization that we have used for our numerical experimentation, has been obtained Starting from the publicly available TREC9 data set,

also known as OHSUMED, through splitting documents into passages using overlapping windows of maximal 50 words each.

In the original data set [159] a text filtering system flows on information with the aim of satisfying the needs of persistent user profiles which represent a need for long-term information. Through the feedback expressed by users, the system learns from the best profile, increasing its performance over time.

The TREC filter track tests to simulate time-critical online text filtering applications, where the value of a document quickly decays with time. This means that potentially relevant documents must be submitted immediately to the user.

The filter operates in a different way than the classic research, as the documents arrive in time sequence. The TREC filter uses three subtasks: adaptive filter, batch filter and routing. With adaptive filter, the system filters documents taking into account only a user profile and a very small number of relevant documents. Each recovered document is judged relevant and the system adaptively updates the filtering profile. With batch and routing filters, the system considers a wide range of evaluated documents useful for building the search profile. Through the batch filter, the system decides to accept or reject each document, while the routing phase returns a classified list of documents.

The original data set consists of several years of selected MEDLINE articles [159]. We have considered the data set obtained with the 1987 data set used as training data in the TREC9 filtering task which consists of approximately 54,000 documents. MEDLINE documents are annotated with MeSH terms (Medical Subject Headings), each defining a binary concept.

3.2.5 Numerical results and final remarks

We have assessed the practical performance of Algorithm DCPCA applied to the DC-SMIL formulation (3.20), by testing it on a set of five medium-size benchmark problems extracted from [82], [95], and on a set of seven large-size benchmark problems extracted from [159].

The relevant characteristics of each problem are reported in Table 3.1 for medium size data sets and in Table 3.2 for the large size ones. In these tables we list the problem size n (i.e., the number of features), the number of instances N , the number of positive bags m , and the number of negative bags k .

Data sets	n	N	m	k
Elephant	230	1391	100	100
Fox	230	1320	100	100
Tiger	230	1220	100	100
Musk 1	166	476	47	45
Musk 2	166	6598	39	63

TABLE 3.1: Characteristics of Medium Size Data sets

The two-level cross-validation protocol, as adopted in [96], has been used in order to tune parameter C and next to train the classifier. In fact, at the higher level, every data-set has been randomly partitioned into 10 pieces of equal size, according to the tenfold cross-validation protocol. Such pieces are grouped into 10 different blocks (the training sets) each containing 9 out of 10 pieces. Every block is then used to train the classifier by running DCPCA, next using the left out piece as the testing-set that returns the percentage of correctly classified

Data sets	n	N	m	k
TST01	6668	3224	200	200
TST02	6842	3344	200	200
TST03	6568	3246	200	200
TST04	6626	3391	200	200
TST07	7037	3367	200	200
TST09	6982	3300	200	200
TST10	7073	3453	200	200

TABLE 3.2: Characteristics of Large Size Data sets

bags (test correctness). Before proceeding with the training phase, a suitable value of parameter C in the set $\{2^{-7}, 2^{-6}, \dots, 1, \dots, 2^6, 2^7\}$, is selected by means of a lower-level five fold cross-validation protocol on each training set (the model-selection phase). The selected C value, for each training set, is the one returning the highest average test-correctness in the model-selection phase.

The selection of an appropriate starting point is a key issue to ensure good performance for a local optimization algorithm like DCPCA. For each training set, denoted by \bar{w}_+ the barycenter of all the instances belonging to positive bags, and by \bar{w}_- the barycenter of all the instances belonging to negative bags, we have selected the starting point (w_0, r_0) by setting

$$w_0 = \lambda \bar{w}_+ + (1 - \lambda) \bar{w}_- \quad (3.22)$$

with $\lambda \in \mathbb{R}$, namely, w_0 is an affine combination of \bar{w}_- and \bar{w}_+ , and choosing r_0 as the smallest radius such that each positive bag has all the instances inside the sphere $S(w_0, r_0)$.

We have adopted the Java implementation of Algorithm DCPCA, running the computational experiments on a 3.50 GHz Intel Core i7 computer. The QP solver of IBM ILOG CPLEX 12.8 [160] has been used to solve the quadratic subprograms. The following set of parameters, according to the notation introduced in [148], has been selected: the optimality parameter $\theta = 0.7$, the subgradient threshold $\eta = 0.7$, the approximate linesearch parameter $m = 0.01$, the agreement rate $\rho = 0.95$, the step-size reduction parameter $\sigma = 0.01$, and the linearization-error threshold $\epsilon = 0.95$.

Furthermore, we have selected $\lambda = 100$ to generate the starting point according to (3.22), we have limited the computational budget \bar{N}_f , in terms of number of evaluations of the objective function, for every execution of DCPCA. We have selected $\bar{N}_f = 500$ for medium-size problems, and $\bar{N}_f = 200$ for large-size problems. We have restricted the size of the bundle to $\bar{N}_f/5$ elements adopting an appropriate bundle-restart strategy.

The numerical results, in terms of the percentage test-correctness averaged over the 10 folds, are reported in Table 3.3 and Table 3.4, respectively for medium and large-size benchmark data sets, where we also report, for a comparison, the corresponding results available in the literature regarding the following methods: DC-MIL [161], MIL-RL [96], mi-SVM [95], MI-SVM [95], MICA [124], MIC^{Bundle} [130], and mi-SVM* [96].

The best performance have been boldfaced and underlined in Tables 3.3 and 3.4, where we see that DC-SMIL combined with DCPCA slightly improves on the existing results of two out of five medium-size test problems, and of two out of seven large-size test problems, while keeping reasonably good correctness for the remaining ones.

TABLE 3.3: Computational results: Average test-correctness on Medium Size Literature Data sets (%)

Data sets	DC-SMIL (%)	DC-MIL (%)	MIL-RL (%)	mi-SVM (%)	MI-SVM (%)	MICA (%)	MIC ^{Bundle} (%)	mi-SVM* (%)
Elephant	84.0	84.0	83.0	82.2	81.4	80.5	80.5	82.5
Fox	59.0	57.0	54.5	58.2	57.8	58.3	58.3	56.5
Tiger	77.5	84.5	75.0	78.4	84.0	82.6	79.1	77.5
Musk 1	80.0	74.5	80.0	87.4	77.9	84.4	75.6	76.7
Musk 2	80.0	74.0	73.0	83.6	84.3	90.5	76.8	77.0

TABLE 3.4: Computational results: Average test-correctness on Large Size Literature Data sets (%)

Data sets	DC-SMIL (%)	DC-MIL (%)	MIL-RL (%)	mi-SVM (%)	MI-SVM (%)	MICA (%)	MIC ^{Bundle} (%)	mi-SVM* (%)
TST01	94.0	94.3	95.5	93.6	93.9	94.5	-	95.5
TST02	81.3	80.0	85.5	78.2	84.5	85.5	-	86.3
TST03	87.0	86.5	86.8	87.0	82.2	86.0	-	83.8
TST04	84.0	86.0	81.0	82.8	82.4	87.7	-	83.0
TST07	82.0	79.8	83.5	81.3	78.0	78.9	-	79.0
TST09	71.5	68.3	68.8	67.5	60.2	61.4	-	61.0
TST10	81.0	78.0	77.5	79.6	79.5	82.3	-	76.3

TABLE 3.5: Computational results: Average cpu-time (s) and correctness (%) of DC-SMIL training-phase: medium-size problems

Data sets	Avg Train-Correctness (%)	CPU time (s)
Elephant	87.72	12.8
Fox	70.72	6.5
Tiger	83.00	6.4
Musk 1	92.65	1.7
Musk 2	88.04	16.3

TABLE 3.6: Computational results: Average cpu-time (s) and correctness (%) of DC-SMIL training-phase: large-size problems

Data sets	Avg Train-Correctness (%)	CPU time (s)
TST01	96.69	177.9
TST02	90.44	162.6
TST03	91.44	160.3
TST04	90.55	184.9
TST07	91.33	154.6
TST09	83.77	170.4
TST10	90.53	184.2

Besides, such encouraging results are obtained in a short amount of computation time, as we show in Tables 3.5 and 3.6, where we report the cpu time (measured in seconds) spent by DCPCA in the training phase, averaged over the 10 training folds. We report in the same tables the results related to training correctness, from which we can verify how the model appropriately generalizes on the training data. Upcoming research directions are about understanding the role played by the starting-point selection in the performance of the approach, and about facing large-scale problems, focusing in particular on appropriately tailoring DCPCA to deal with higher size problems.

Further research may also involve the application of different nonsmooth DC solvers, see [150], [162], whose features allow the improvement of the search for better critical points. Although such approaches would probably return improved performance in terms of train-correctness, it remains an open issue to understand the extent to which the test-correctness can be influenced by the goodness of a critical point.

Chapter 4

Machine Learning and Automated Melanoma Detection

"There is only one question: 'What is really most important, the whole or its parts?'"

– Friederich Holderlin, Letter to Karl (1801)

In chapter 2 and chapter 3, we have treated Multiple-Instance Learning (MIL), a recent machine-learning paradigm which proves to be very efficient for image analysis tasks [163]. In fact, we have highlighted that the MIL algorithms are able to detect relevant local patterns using only the class labels assigned globally to the data, both in the cases of images and videos classification. The troublesome problem of manual data segmentation is effectively overcome by the fact that supervision is based on global labels. Solutions based on MIL approach are of particular interest also considering that they are not only more lightweight in the learning phase but also more precise than traditional single-instance learning ones. In the particular context of medical image and video analysis, the use of solutions that adopt MIL approach is desirable [88]. From the literature, it emerges how MIL approaches have been proposed to support the diagnosis of aggressive pathologies. Our attention has focused on skin cancers and in particular on melanoma.

4.1 Statistics on Melanoma

Melanoma is currently one of the most important types of cancer for deaths in the world, and is the most deadly type of skin cancer (see Figure 4.1 and Figure 4.2).

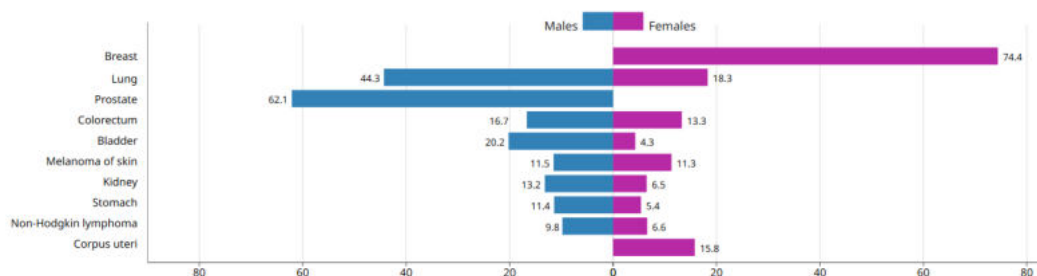


FIGURE 4.1: Incidence rates per sex, top 10 cancers [1]

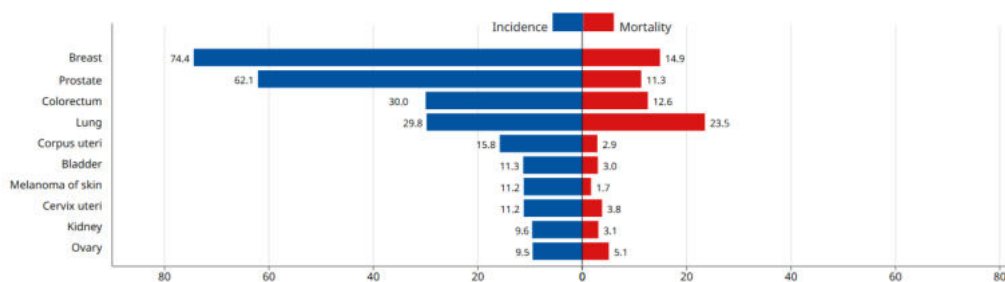


FIGURE 4.2: Incidence and mortality rates, top 10 cancers [1]

According to the latest report of the World Health Organization, we refer to some statistics drawn from Global Cancer Observatory (GCO) relating to the onset of new cases, the incidence and the mortality rate at 5 years divided by gender and geographical area [1].

The Global Cancer Observatory (GCO) is an interactive web-based platform presenting global cancer statistics to inform cancer control and cancer research.

The platform focuses on the visualization of cancer indicators to illustrate the changing scale, epidemiological profile, and impact of the disease worldwide, using data from several key projects of IARC's Section of Cancer Surveillance (CSU), including:

- GLOBOCAN
- Cancer Incidence in Five Continents (CI5)
- International Incidence of Childhood Cancer (IICC)
- Cancer Survival in Africa, Asia
- the Caribbean and Central America (SurvCan).

In Figure 4.3 we report specific data concerning the number of new cases, the number of deaths, the incidence, the mortality and the 5-year prevalence for both sexes at 2018.

The data presented in the Global Cancer Observatory are the best available for each country worldwide. However, caution is needed when interpreting the data, recognizing the current limitations in the quality and coverage of cancer data, particularly in low- and middle-income countries.

In Figure 4.4 the values of age standardized incidents rates of melanoma, both for males and females, are reported.

The darker colors indicate areas most affected by melanoma. Generally, the reported statistics have similar values for both males and females. The following Figures 4.5 and 4.6 show the values, differentiated by sex, of the incidence and the mortality rates of melanoma specified by geographical area.

In recent decades, malignant melanoma has become one of the most aggressive cancers and it is spreading rapidly in many areas of the world. Populations living in Europe, North America and Australia above all are strongly affected by this type of skin cancer.

All over the world, in 2018 melanoma has caused over 60.000 deaths and over 280.000 new cases of melanoma have been diagnosed [1]. More than one million non-melanoma skin cancers and more than 250.000 of melanoma skin cancers occur each year.

If we focus on the most affected areas, the number of cases and the incidence rates of melanoma are even more worrying. As reported in Figure 4.7, in the U.S.A., for example, melanoma ranks 5-th for expected new cases in 2019 both for males and females [164].

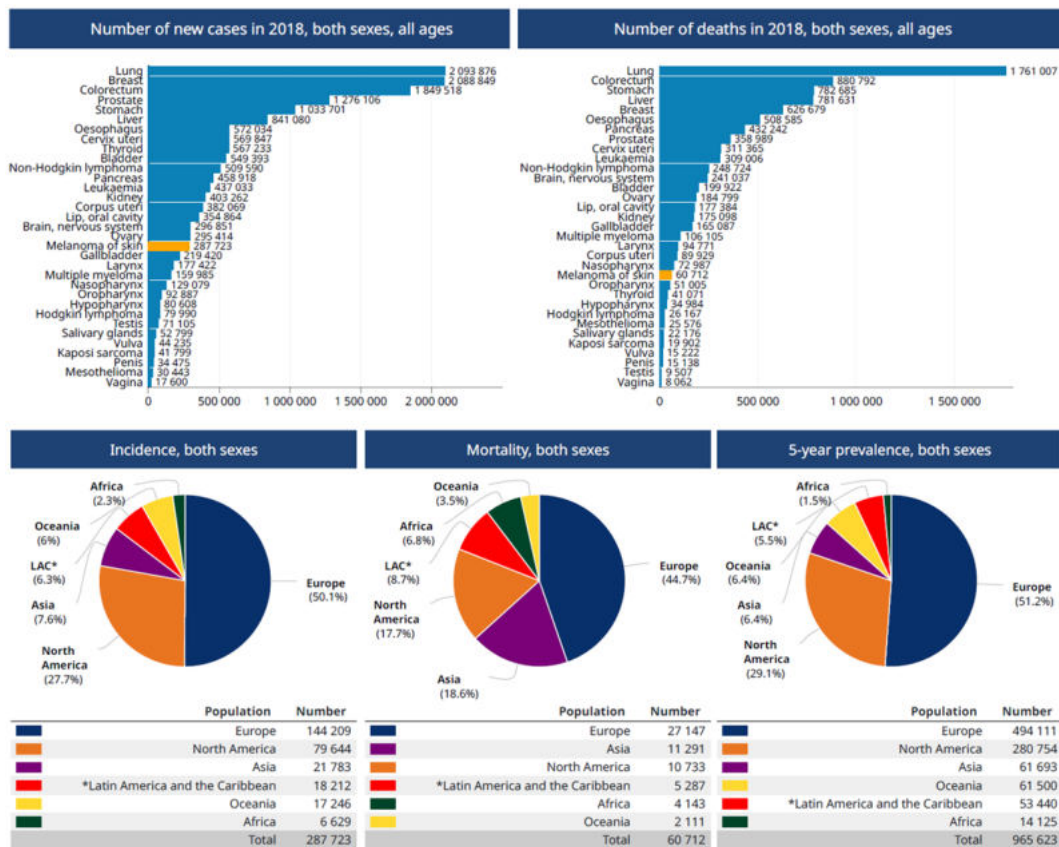


FIGURE 4.3: Statistics on Melanoma in 2018 [1]

Still referring to the U.S.A., Figure 4.8 shows that melanoma’s incidence rate is characterized by a positive trend among the tumors responsible for the greatest number of deaths [164].

Despite the ever increasing diffusion and its aggressiveness, melanoma is a type of curable cancer when it is identified by an early diagnosis (see Figure 4.9). Some clinical protocols such as the “ABCDE” rule [165] and the 7-PCL [166] have been established to facilitate the task of specialists in identifying the lesion from the initial phase. These clinical protocols take into consideration certain lesion features such as asymmetry, irregular edges, colors, diameters greater than 6 mm and evolving stages.

On one hand the massive spread in many areas and the aggressiveness of this type of skin cancer, and on the other hand the possibility that an early diagnosis followed by an excision allow the survival of the individual, have led us to experiment our proposed classification methods focusing on the automatic diagnosis of melanoma. In the next section we refer to the pathogenetic mechanisms of skin cancer to help the reader’s understanding of visual differentiation.

4.2 Skin layers

The *epidermis* is the most external layer of skin and it’s visible to the naked eye; although the epidermis is extremely thin, it’s an impermeable barrier that protects against bacteria and

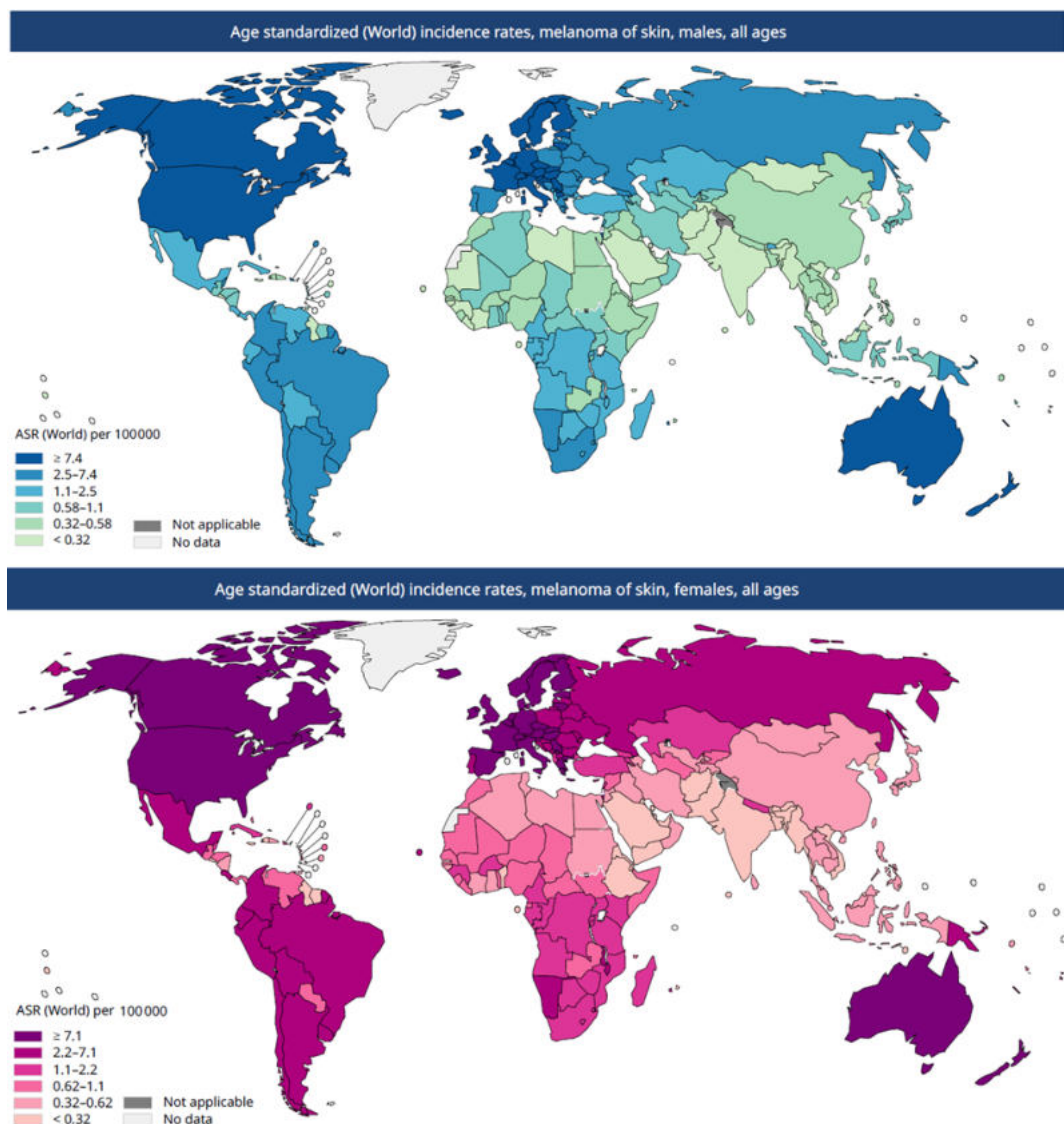


FIGURE 4.4: Melanoma age standardized incidents rates per sex [1]

other microorganisms present in the surrounding environment. Various cells are located inside the epidermis:

- Keratinocytes that constitute the outermost part of the epidermis and are rich in a protein called keratin which makes them resistant; they are interconnected to form a barrier impermeable to water.
- Basal cells that form a large part of the basal layer of the epidermis. They are the only cells in the epidermis that divide and create new cells called keratinocytes.
- Melanocytes that are located in the basal layer of the epidermis. They are distributed regularly in the middle of the basal cells; the melanocytes produce melanin, a pigmented protein that gives color to the skin and hair, and provides protection against the damage of ultraviolet rays.

Cancer incidence statistics worldwide and by region						
	Incidence					
	Both sexes		Males		Females	
	New cases	Cum. risk 0-74 (%)	New cases	Cum. risk 0-74 (%)	New cases	Cum. risk 0-74 (%)
Eastern Africa	2 244	0.12	956	0.10	1 288	0.13
Middle Africa	1 037	0.15	475	0.16	562	0.14
Northern Africa	979	0.05	473	0.05	506	0.05
Southern Africa	1 234	0.22	635	0.28	599	0.18
Western Africa	1 135	0.07	426	0.05	709	0.09
Caribbean	549	0.10	305	0.12	244	0.09
Central America	3 490	0.19	1 562	0.19	1 928	0.19
South America	14 173	0.29	6 899	0.32	7 274	0.27
North America	79 644	1.40	46 635	1.67	33 009	1.18
Eastern Asia	10 228	0.04	5 346	0.05	4 882	0.04
South-Eastern Asia	2 827	0.05	1 499	0.06	1 328	0.05
South-Central Asia	5 359	0.03	2 465	0.03	2 894	0.04
Western Asia	3 369	0.15	1 745	0.17	1 624	0.14
Central and Eastern Europe	26 220	0.59	11 546	0.63	14 674	0.57
Western Europe	62 821	1.97	31 719	1.99	31 102	1.96
Southern Europe	23 831	0.94	12 229	1.01	11 602	0.88
Northern Europe	31 337	1.82	15 674	1.83	15 663	1.83
Australia and New Zealand	16 978	3.72	9 988	4.41	6 990	3.05
Melanesia	242	0.34	109	0.29	133	0.38
Polynesia	26	0.46	12	0.43	14	0.48
Micronesia	0	0	0	0	0	0
Low HDI	4 448	0.10	1 957	0.09	2 491	0.10
Medium HDI	9 704	0.04	4 568	0.04	5 136	0.04
High HDI	33 672	0.11	16 434	0.11	17 238	0.10
Very high HDI	239 719	1.09	127 645	1.20	112 074	1.00
World	287 723	0.35	150 698	0.39	137 025	0.31

FIGURE 4.5: Detailed melanoma incidence per sex and by region [1]

Cancer mortality statistics worldwide and by region						
	Mortality					
	Both sexes		Males		Females	
	Deaths	Cum. risk 0-74 (%)	Deaths	Cum. risk 0-74 (%)	Deaths	Cum. risk 0-74 (%)
Eastern Africa	1 495	0.08	625	0.07	870	0.09
Middle Africa	757	0.11	351	0.13	406	0.10
Northern Africa	532	0.03	285	0.03	247	0.02
Southern Africa	574	0.11	303	0.14	271	0.08
Western Africa	785	0.05	300	0.04	485	0.07
Caribbean	162	0.03	98	0.03	64	0.02
Central America	870	0.05	480	0.06	390	0.04
South America	4 255	0.08	2 440	0.11	1 815	0.06
North America	10 733	0.15	7 053	0.21	3 680	0.10
Eastern Asia	5 050	0.02	2 765	0.02	2 285	0.02
South-Eastern Asia	1 633	0.03	927	0.04	706	0.02
South-Central Asia	3 361	0.02	1 842	0.02	1 519	0.02
Western Asia	1 247	0.05	718	0.07	529	0.04
Central and Eastern Europe	9 180	0.19	4 622	0.24	4 558	0.15
Western Europe	8 054	0.18	4 750	0.23	3 304	0.14
Southern Europe	5 194	0.16	3 059	0.21	2 135	0.12
Northern Europe	4 719	0.21	2 812	0.27	1 907	0.15
Australia and New Zealand	2 062	0.36	1 372	0.50	690	0.23
Melanesia	47	0.06	27	0.06	20	0.06
Polynesia	2	0.03	2	0.07	0	0
Micronesia	0	0	0	0	0	0
Low HDI	2 970	0.07	1 302	0.06	1 668	0.07
Medium HDI	5 674	0.03	3 028	0.03	2 646	0.02
High HDI	12 313	0.04	6 943	0.04	5 370	0.03
Very high HDI	39 743	0.15	23 551	0.19	16 192	0.11
World	60 712	0.07	34 831	0.08	25 881	0.05

FIGURE 4.6: Detailed melanoma mortality rates per sex and by region[1]

The second layer, the *dermis*, is of thicker consistency and is located below the epidermis. It contains blood and lymphatic vessels, nerve endings, muscle fibers, sebaceous and sweat glands and hair follicles (see Figure 4.10). The dermis in turn is divided into two layers:



Estimated New Cases						
			Males	Females		
Prostate	174,650	20%			Breast	268,600 30%
Lung & bronchus	116,440	13%			Lung & bronchus	111,710 13%
Colon & rectum	78,500	9%			Colon & rectum	67,100 8%
Urinary bladder	61,700	7%			Uterine corpus	61,880 7%
Melanoma of the skin	57,220	7%			Melanoma of the skin	39,260 4%
Kidney & renal pelvis	44,120	5%			Thyroid	37,810 4%
Non-Hodgkin lymphoma	41,090	5%			Non-Hodgkin lymphoma	33,110 4%
Oral cavity & pharynx	38,140	4%			Kidney & renal pelvis	29,700 3%
Leukemia	35,920	4%			Pancreas	26,830 3%
Pancreas	29,940	3%			Leukemia	25,860 3%
All Sites	870,970	100%	All Sites	891,480 100%		

FIGURE 4.7: Ten Leading Cancer Types for the Estimated New Cancer Cases by Sex, in United States in 2019 [164]

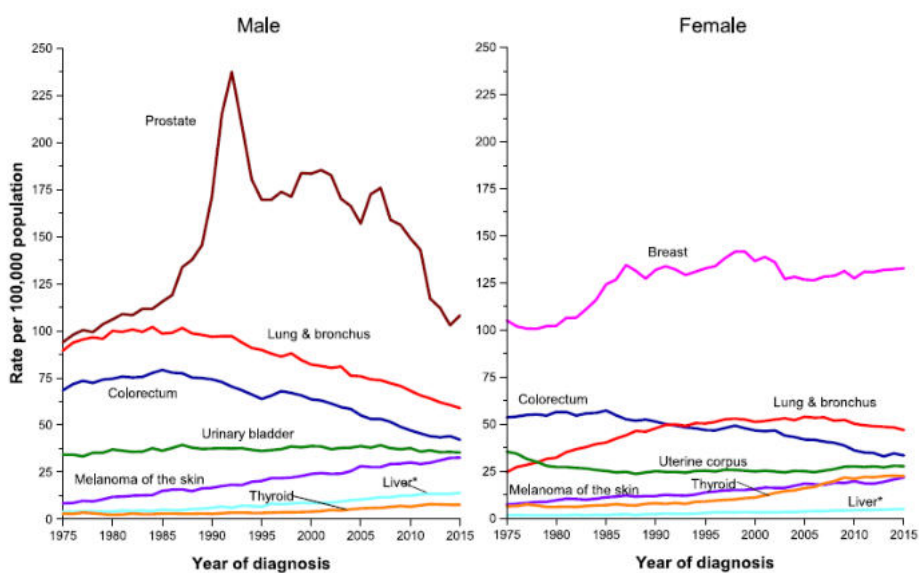


FIGURE 4.8: Trends in Incidence Rates for Selected Cancers by Sex, United States, 1975 to 2015 [164]

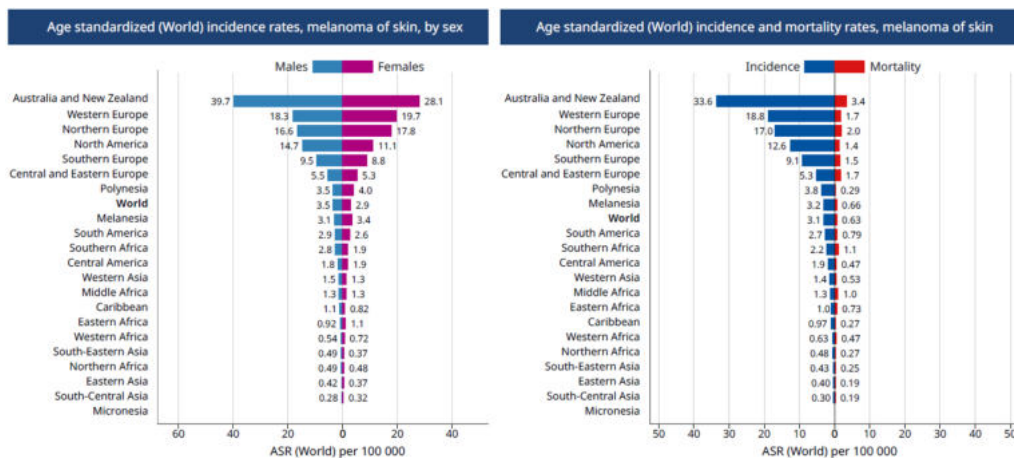


FIGURE 4.9: Melanoma age standardized incidents and mortality rates per sex

- Dermal papillary, the upper layer consisting of soft connective tissue, blood vessels and nerves. Digitiform projections called papillae connect the dermis to the epidermis and provide essential nutrients.
- Reticular dermis, is the thicker and inferior layer; is a network of collagen fibers and dense connective tissue that gives the skin its strength and elasticity and that contains a rich supply of blood vessels, as well as lymphatic vessels, glands and hair follicles.

Finally there is the subcutaneous layer consisting of a thick layer of fat and connective tissue located under the skin. Like the dermis, it contains numerous blood and lymphatic vessels and Isola and conserves body heat, cushions the strokes so as to protect the underlying tissues and internal organs from damage caused by possible trauma, and is a resource and a reserve of energy [167].

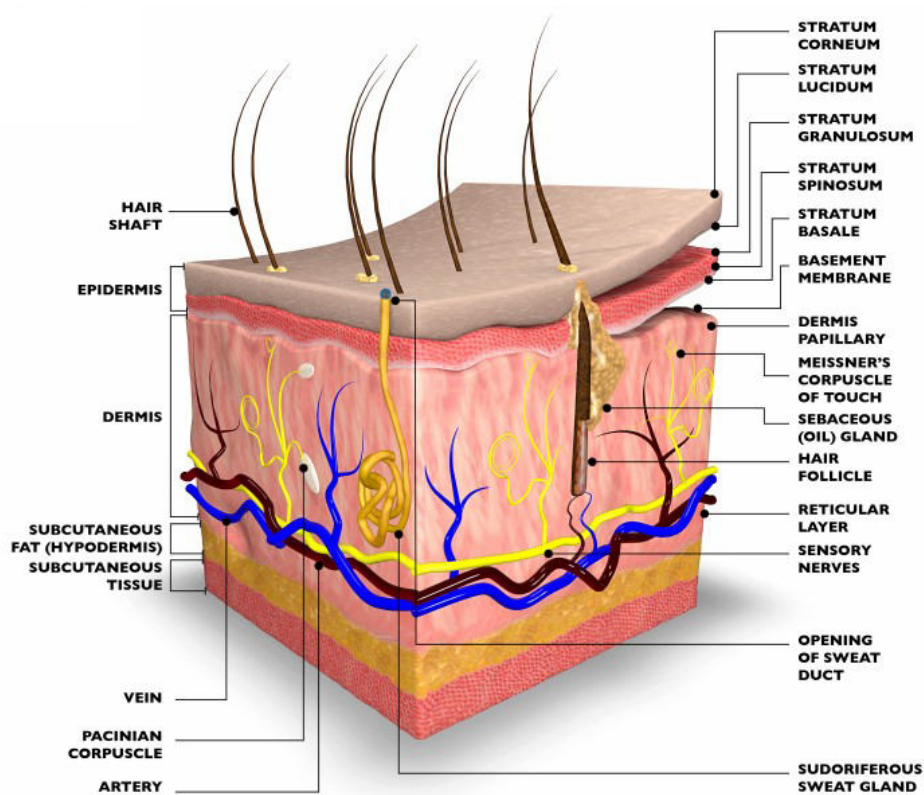


FIGURE 4.10: Skin Layers

The layers of the skin based on their composition have distinct optical properties. A source of white light reflected on the skin penetrates into the superficial layers of the skin, and while a part of it is absorbed, much is reflected; this allows it to be recorded using appropriate digital equipment.

The epidermis is largely composed of connective tissues, and also contains melanocytes that produce melanin cells. Melanin is a pigment that strongly absorbs light in the blue part of the visible spectrum and UV acts as a protective filter against UV radiation. All the light not absorbed by the melanin can potentially cross the dermis that contains sensors, receptors, blood vessels and nerve ends.

Pigmented skin lesions appear as patches of darker color on the skin. In most cases, the cause is an excessive concentration of melanin in the skin. In benign lesions (common nevi),

melanin deposits are normally found in the epidermis. In malignant lesions (melanoma), the melanocytes reproduce the melanin at a high and abnormal rhythm and, however, with optical properties similar to those of the highly pigmented normal skin.

The presence of melanin in the dermis is the most significant sign of melanoma even if some benign nevi also have dermal deposits, albeit with more regular spatial patterns than melanoma (Figure 4.11).

The thickening of collagen fibers in the papillary dermis (fibrosis), the increase in blood supply to the periphery of the lesion (erythematic reaction) and the lack of blood inside the lesion are further signs indicative of a potential *in situ* melanoma.

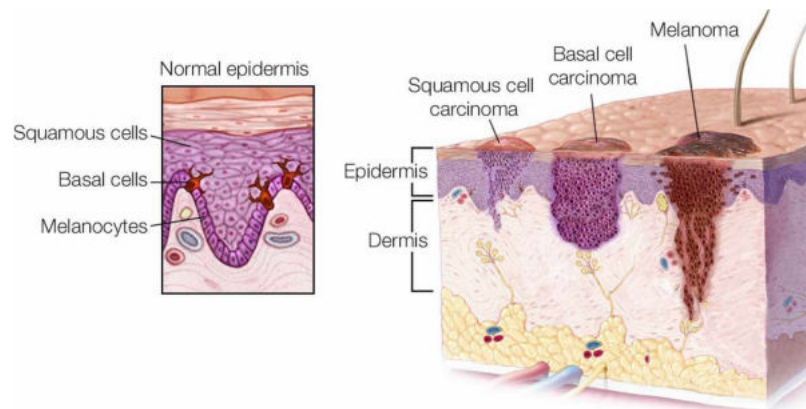


FIGURE 4.11: Types of skin Lesion

Even the colors of the malignant lesions usually show characteristic shades that are not found in other skin conditions. This provides an important diagnostic indication: if the visual approach corroborates a suspicion of skin cancer, histology is necessary to make an explicit diagnosis.

4.3 Computer Aided Diagnosis Systems

The medical scientific community looks with increasing interest at Computer Vision Systems for skin lesion characterization. A Computer Vision System, also referred to as Computer Aided Diagnosis (CAD) System, is a mix of technologies that enables a computing device to inspect, evaluate and identify still or moving images.

As we saw in Section 4.1, skin cancer is one of the most insidious and aggressive tumors. The growing spread of this particular type of cancer is also a cause for concern. Fortunately, a diagnosis in the initial stages often allows a successful treatment of this type of skin lesion. Early detection and removal turn out to be a decisive treatment when the tumor is still small and thin. This justifies the need to provide tools that allow early and accurate diagnosis, both facilitating the work of specialists and allowing the release of low-cost solutions for effective self-diagnosis.

Currently the diagnosis of melanoma through fully automated analysis is far from being considered stable and the diagnostic accuracy is still dependent on the experience of dermatologists [168]; a desirable solution would be the use of systems able to offer a second opinion to the specialist. In clinical evaluation, dermatologists adhere to some reference protocols that better than others allow to identify the tumor lesions of the skin.

The physicians typically refer to the ABCDE protocol [169](which stands for lesion asymmetry, border irregularities, color variation, diameter and evolution), the 7-point checklist [166] and the Menzies method [170]. If the specialists consider the case examined to be suspicious, they proceed with further investigation on a portion of tissue taken by biopsy. The pitfall lies is the fact that in the initial stages melanoma appears similar to other benign lesions and it is difficult to identify even for expert dermatologists; the tendency of physicians to underestimate melanoma in the initial phase should not be overlooked [171].

Medical imaging has transformed the way physicians perform assessment, diagnosis, and disease monitoring. Digitized medical images contain an enormous amount of numerical data that simple visual observation, the so-called "qualitative analysis", cannot process. Visual analysis manages to extract only a small part of the information contained in a digital medical image. If these images are processed and analyzed through evolved CAD Systems, it is possible to obtain significant quantitative data, able to provide information on the underlying physiopathological phenomena; thus, it's possible to support diagnostic and surgical intervention. The visual nature of skin diseases makes digital imaging extremely useful in everyday practice.

The design of computer vision systems for the diagnosis of skin lesions encompasses image acquisition, image processing, segmentation, features extraction and, at the end, classification step (see Figure 4.12). In the present Section we will focus on peculiarities of each of these phases, referring to applications related to skin lesions.

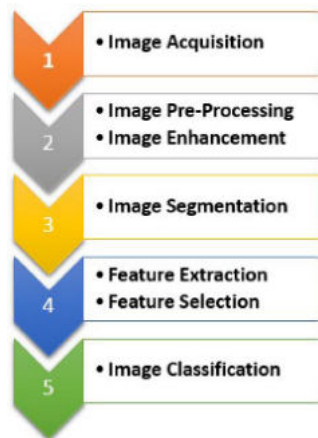


FIGURE 4.12: Main steps in biomedical image processing

4.3.1 Image Acquisition methods

Imaging in dermatology plays a fundamental role in the assessment and monitoring of skin cancer. Although physicians rely primarily on their eyes, numerous tools have been developed to improve melanoma detection. Imaging can be at the level of *total body photography* to detect changes in the size, shape or color of individual lesions, but also at the subcellular level with techniques such as *confocal reflectance microscopy* used to visualize atypical cells.

The various imaging techniques are used upstream of computer vision systems with the aim of providing support for the identification of specific lesions within a field of lesions of similar appearance.

In particular, Epiluminescence Microscopy (ELM, or Dermoscopy) makes the epidermis translucent and allows an in-depth analysis of the surface texture of skin lesions. Transmission Electron Microscopy (TEM), returns detailed information on the structure of the lesion in the dermis, and it is therefore used to monitor the evolution of melanoma. Through the lateral transillumination, the light source points towards the lesion with a certain angle, making translucent both the superficial and the sub-surface layers of the skin. In this way, it's possible to appreciate the variations of blood flow and vascularization of the lesion also analyzing the colorimetric variations of the pigmentation. Computed tomography (CT) images have also been used for detection of melanomas monitoring the progress of the disease and patient's response to possible treatments.

TECHNIQUE	ADVANTAGES
PHOTOGRAPHY Digital Photography, Total Body Photography, UV Photography	<ul style="list-style-type: none"> • Economical, long-term data storage, easy management. • Facilitates skin self-examination. • Long term Follow Up. • 3D-TBP generates a 3D avatar, allowing for enhanced visualization of body surface. • Possibility to evaluate sun damage.
DERMOSCOPY (ELM)	<ul style="list-style-type: none"> • Magnifies skin 20× to facilitate diagnosis. • Monitors skin lesions over time. • Establish criteria for diagnosing skin cancer that correlates well with histopathologic features. • Diagnoses pigmented and non-pigmented skin cancer with better sensitivity, specificity and correctness compared to ones made with naked eyes
REFLECTANCE CONFOCAL MICROSCOPY (RCM)	<ul style="list-style-type: none"> • High magnification 30×, which allows for imaging of microscopic structures. • Allows for imaging to a depth of 200μm down to papillary dermis • Non-invasive and may reduce the need for biopsy. • Low power laser without tissue damage. • Facilitates diagnoses of equivocal features, allows for delineation of surgical margins, and useful for long-term monitoring.
OPTICAL COHERENCE TOMOGRAPHY (OCT)	<ul style="list-style-type: none"> • Non-invasive and may reduce the need for biopsy • High resolution of 3 – 15μm allows for imaging of microscopic features. • Depth of 1.5 mm is better than RCM • Generates 2D and 3D images. • Wide applications for imaging lesions, aging skin, skin moisture and engineered tissue. • Possible use with other techniques, including Doppler to enhance diagnostic correctness.

TABLE 4.1: Advantages of most prominent current imaging techniques in Dermatology

Positron emission tomography (PET) involves the use of fluoro-deoxyglucose [172], and is a useful diagnostic method for examining the potential metastatic of cutaneous melanoma. The absorption of fluoro-deoxyglucose has been correlated with the rate of proliferation and

therefore the degree of malignancy of a given tumor. Alternative techniques like multifrequency electrical impedance [173] or Raman spectra [174] are other useful screening methods. The electrical impedance of a biological material is associated with the momentary physical properties of the tissue. Raman spectra are obtained by aiming a laser beam at a sample of skin lesion. The laser beam excites the molecules in the sample and a dispersion effect returns useful information on the physical structure of the tissue sample. The various image acquisition techniques used in Dermatology present specific characteristics that make it more or less suitable depending on the case. The pros and cons of some of the most widespread techniques are shown in Table 4.1 and in Table 4.2.

TECHNIQUE	LIMITATIONS
PHOTOGRAPHY Digital Photography, Total Body Photography, UV Photography	<ul style="list-style-type: none"> • Only superficial morphological analysis of the skin. • Traditional 2D-TBP takes time and may be uncomfortable for the patient. • Heavy management of images to ensure patient privacy.
DERMOSCOPY (ELM)	<ul style="list-style-type: none"> • Proper training is needed. • Interpretation of results is subjective. • Limited magnification restricts its applications.
REFLECTANCE CONFOCAL MICROSCOPY (RCM)	<ul style="list-style-type: none"> • Proper training is needed (associated learning curve). • Unable to image lesions beyond papillary dermis, thus cannot reliably evaluate tumor invasion.
OPTICAL COHERENCE TOMOGRAPHY (OCT)	<ul style="list-style-type: none"> • Expensive and requires proper training and experience. • Strong scattering limits the depth to thin tumors and cannot reliably evaluate tumor invasion. • Cannot differentiate between benign and malignant lesions effectively due to limited resolution

TABLE 4.2: Limitations of most prominent current imaging techniques in Dermatology

One of the increasingly adopted techniques is epiluminescence microscopy (ELM, or Dermoscopy). Dermoscopy is a non-invasive technique that uses incident light beams and possibly baths in special oil to inspect the sub-surface structures of the skin. Although the detection of melanoma by Dermoscopy is better than the non-assisted one, the training of the dermatologist remains the discriminating element for an accurate diagnosis. Distinguishing a melanoma from a melanocytic nevus is not easy, especially in the initial phase, even when expert dermatologists operate with the aid of dermoscopy.

CAD Systems are able to extract information, in terms of color variation, asymmetry, texture features, which may not be easily perceived by human eyes. Dermoscopy is also one of the cheapest ways through which identify and classify skin cancer. The classical methodology would require the removal of a piece of tissue from the patient's body in order to perform histologist analyzes. Biopsy is an invasive examination that involves costs for both the patient and the health system.

CAD Systems are designed to detect the presence of cancer cells in the image. These frameworks typically use digital image processing techniques obtainable, for example, through

a dermatoscope and automatic learning techniques such as SVM for the final classification of images. These solutions allow early diagnosis of skin cancers without the need to apply oil on the lesion: clear images are obtained with a quick and clean approach. In this way, solutions that exploit image magnification together with artificial intelligence approaches allow the automatic diagnosis of melanoma directly from dermoscopic images.

Among the advantages of Dermoscopy (ELM), it should not be overlooked that this imaging technique eliminates superficial skin reflex, allowing better diagnostic accuracy compared to standard photography. Malignant melanoma lesions are asymmetrical and have an irregular and serrated edges. The accurate observation of skin lesions is of fundamental importance, considering that atypical moles could be benign. These moles with an unusual appearance are also known as *dysplastic nevi*. Dysplastic nevi can resemble melanoma and people exhibiting them are at great risk of developing melanoma in a mole or other parts of the body.

Figure 4.13 shows a sample of images of common nevi, dysplastic nevi and malignant skin lesions (melanomas) taken from PH^2 which is a dermoscopic dataset [3].

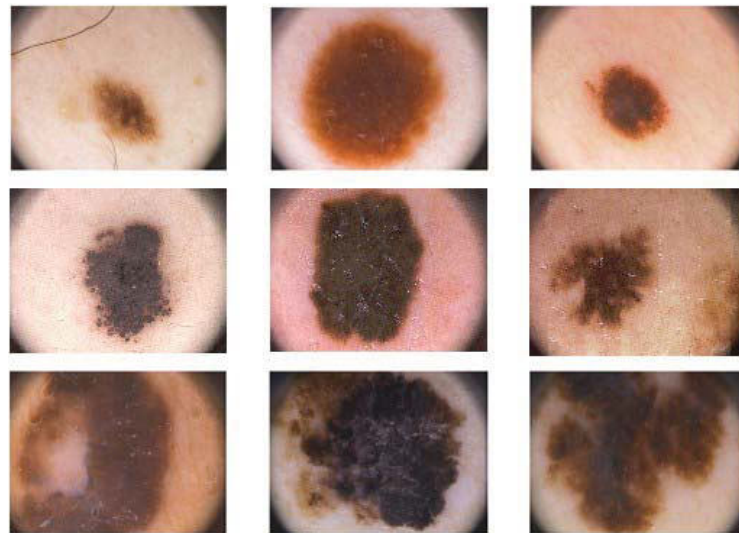


FIGURE 4.13: A collection of images from PH^2 database: common nevi (1st row), dysplastic nevi (2nd row) and melanomas (3rd row).

The advancement of camera technology even on wearable devices such as smartphones, as well as the new paradigms of the Internet of things, opens new horizons to the possibilities of creating self-diagnosis systems on accessible skin lesions [175]. These solutions will be possible, also thanks to development of new image classification algorithms (see [145], [176], [177]), characterized by streamlined learning phases that allow to overcome the need of experts manual segmentation of images.

More and more attention is being paid to the solutions that intend to implement new health care models, involving and driven by the patient, supporting decision and responsibility processes, considering predictive and preventive aspects [10].

4.3.2 Image Pre-processing

CAD systems makes possible a better monitoring of patients at risk avoiding unnecessary biopsies as well as identifying the sites of previous excisions; in this way it is possible to

improve the reliability of the diagnosis. Pre-processing is the first stage of automated detection used to improve the quality of images. Depending on the technique adopted for the generation of images, there is a need for customized pre-processing steps. A first goal is to separate the lesion area from healthy skin. This operation is difficult due to the presence on dermoscopic images of irregular lighting, frames, gelatinous material and intrinsic skin characteristics that make difficult edges detection of lesion. Therefore, anything that could affect the image must be localized and then removed, masked or replaced. Many approaches have been used in literature, including image resizing, masking, cropping, hair removal and the conversion from RGB to gray-scale images.

As we saw in the previous Section, Dermoscopy is a useful technique to investigate skin lesions. Dermoscopy is increasingly considered a preferential approach to promote both correctness and accuracy in diagnosis. Skin cancer images obtained with this technology are used in CAD Systems to screen melanoma, verifying also stages of evolution in a timely and accurate way. Also dermoscopic images may be influenced by some artifacts, including the gradual transition between the lesion and the skin, the presence of hair, the eventual transition effects of gel and the water bubble, plus colored lesions and specular color reflections.

The presence of these artifacts can generate incorrect assessments of the lesion and, therefore, of the classification of skin cancer. It is necessary to provide pre-processing steps to properly manage these artifacts, by removing unrelated and excess parts present in the background. Image pre-processing techniques were first used to limit the search for outlier [178]. Over time, other needs emerged, such as the need of removing hair or reconstructing images processed after a pre-processing phase. It is possible to summarize the results of the pre-processing phase in three main categories: image improvement, image restoration and hair removal (see Figure 4.14).

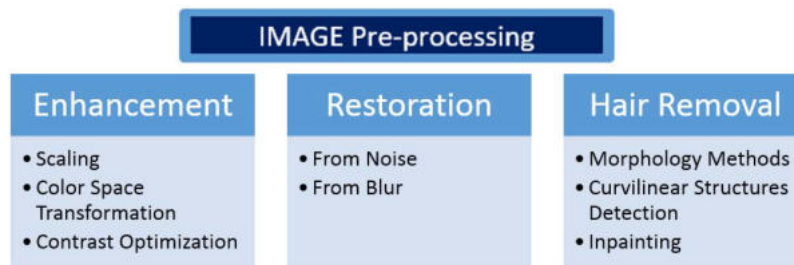


FIGURE 4.14: Image pre-processing for automated skin detection

The techniques used by researchers for image pre-processing useful for CAD systems for melanoma detection are reported below.

Image Enhancement

Image enhancement improves the specialists' image interpretation, providing also "better" inputs for other image processing techniques. Nowadays, many techniques have been proposed which can improve a digital image without degrading it. These methods are very problem-oriented: the choice of parameters and the way they are modified are strongly related to the specific interest. Typically image enhancement methods can be divided into:

- Spatial domain
- Frequency domain.

With spatial domain techniques, the pixels of a certain image are directly manipulated to achieve the desired improvement. Vice-versa with frequency domain methods, the Fourier transform of the image is computed to have a representation in the frequency domain. Once the optimizations are made on the Fourier transform of the image, the inverse transformation is performed to get the resulting image back. In this way, the image parameters are improved, such as image brightness, contrast or gray level distribution. Image Enhancement is defined as provider of the “better” transform representation for further automated steps of detection [179]. Thus, image enhancement can be contextualized in three categories:

Image scaling

In computer graphics, image scaling refers to resizing a digital image, while magnification is known as up-scaling or improved resolution. When resizing a vector graphic image, the graphic primitives that make up the image can be resized by geometric transformations, without loss of the image’s quality. When resizing a raster graphic, a new image must be generated with a larger or smaller number of pixels. If the number of pixels decreases (scaling down), there may be a loss of visible quality. Image scaling techniques are applied due to the lack of equal and standard image sizes. Because skin cancer images can be collected from different sources, the first step is to resize images, for example, to get fixed width pixels but varying height sizes [180].

Many algorithms have been proposed for image scaling. We report Nearest-Neighbor interpolation [181]. This algorithm uniforms the color of the image replacing the values of some pixels with the nearest pixel values. This can preserve sharp details, but also introduce jaggedness in previously smooth images. Much interest is laid on deep convolutional neural networks, that typically uses machine learning for more detailed images such as photographs and complex artwork [182].

Color Space Transformation

A color space is a geometrical and mathematical representation of color. There is no general method that is applicable to all domains; the number of variables involved make for complexity such that a complete theoretical analysis is not feasible in most practical applications. In any problem of color quantification, the first step toward a solution is to define the color space. Historically, many different representations have been defined, but each was developed for a specific purpose. Since color information plays an inevitable role in skin cancer detection systems, researchers try to extract the more closely related color of images for further processing.

Generally, the common color spaces include RGB, HSV, HSI, CIE LAB and CIE-XYZ. RGB is a color space which comprises the red, green, and blue spectral wavelength. The most frequent presentation of colors in image processing is RGB. Since RGB color space has some limitation in high level processing, other color space representations have been developed [183]. HSV and HSI color spaces imitate the human visual perception of color in terms of hue, saturation and intensity which are respectively the average wavelength of the color, the amount of white in the color and the brightness. CIE-LAB is a widely used color space which has been proposed to provide uniformity. CIE-XYZ is another color space which can produce every color with positive tristimulus values [184]. The human eye has photoreceptors for the display of medium and high brightness colors with sensitivity peaks in short wavelengths (S, 420-440 nm), averages (M, 530-540 nm), and long (L, 560-580 nm).

Thus, the sensation of color is described by three parameters. These tristimulus values of a color are the sum of the 3 primary colors in a color model with 3 additive components (see Figure 4.15).

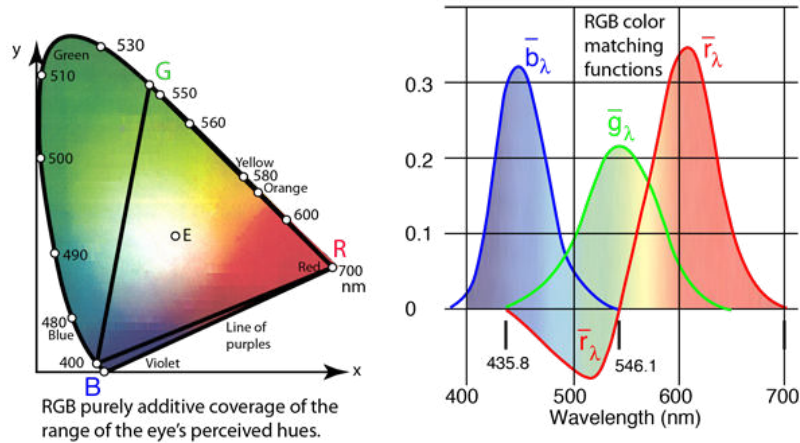


FIGURE 4.15: The color features of the CIE standard

To effectively detect the edges of the lesions it is useful to obtain high-level variations between intensities; one popular way to guarantee this result is to convert color image to gray-scale version. Since LAB is one of the useful color models that represent each color through three luminance components, red/green and blue/yellow, it may be useful to transform the RGB into LAB passing through an intermediate transformation using the XYZ color space. Luminance would present a gray-scale skin image [185].

One of the steps that must be taken into account optimizing an image, is a correct evaluation of the color space most appropriate to the context of interest. Image transformations in different color spaces may be necessary.

Contrast Enhancement

Image enhancement techniques have been widely used where the subjective quality of images is important for human interpretation. The contrast is given by the difference in luminance reflected by two adjacent surfaces, that is the difference in visual properties that makes an object distinguishable from other objects and from the background. When the contrast of an image is concentrated on a specific region, information can be lost in areas that are uniformly concentrated. Contrast Enhancement plays a fundamental role in increasing the quality of an image, both for sharpening the edge and for improving the difference in brightness between background and foreground.

In general, the most widely adopted techniques of contrast enhancement are classifiable in "Linear" and "Nonlinear" [186]:

- *Linear contrast enhancement techniques* refer to contrast stretching techniques. The contrast of the input image is emphasized by varying the values of the gray level so that the histogram extends to the entire interval [187]. These techniques are mainly used in remote sensing images.
- *Nonlinear contrast enhancement techniques* are commonly used in the medical field [188]. Contrast enhancement is obtained, through equalization steps and using algorithms to generate the histograms associated with images.

For automated skin cancer detection, the local details of the region where melanoma is located are more important than those of the background of healthy skin. Techniques such as the equalization histogram (HE), the equalization of the adaptive histogram (AHE) and unsharp masking are often used together [180]. On the other hand, although HE can also sharpen the image, it has the drawback of reducing the surrounding details. For this reason the researchers are still investigating the issues of contrast enhancement. Delgado et al. [189], proposed a contrast enhancement method, based on independent histogram research (IHP). The presented framework foresees the linear transformation of the original RGB image into a decorrelated chromatic space, in order to completely separate the lesion and the background skin.

Edge detection is then performed on transformed images, characterized by high contrast, using a simple clustering algorithm. A technique known as “independent histogram pursuit” consists in finding a combination of spectral bands that enhance the contrast between healthy skin and lesion [190].

Image Restoration

Image restoration is defined as the procedure to recover degraded, blurred and noisy images [191]. It is possible to restore images affected by one or more noises using to different ways. Image noise is a random variation (not present in the photographed object) of brightness or color information in images, and is usually an aspect of electronic noise. It can be produced by the sensor and the various circuits of a scanner or a digital camera.

The noise of the image can also come from the grain of the film and in the inevitable noise of the shot of an ideal photon detector. Image noise is an unwanted by-product of image acquisition that adds incorrect and foreign information [191]. Since the degraded images may imply incorrect diagnoses, it is necessary to intervene. Noted the noises present in the image it is possible to obtain an effective restoration by applying specific filters and algorithms.

Restoration from noise

Image reduction is an essential step in preprocessing an image. It is difficult to apply a de-noising method without considering the specificity of the application. The goal is to pursue both the suppression of noise and the preservation of sharp edges [192]. Among the noises most frequently present in the dermoscopic images, we recall those of Gaussian, Salt and Pepper, Poisson and Speckle [193].

The presence of a noise may depend from the adopted image acquisition technology. In what follows, we have a brief description of the noises that may be present in dermoscopic images. To complete the description, Figure 4.16 gives a graphic representation of the noises on a photo.

- *Gaussian noise*

The standard model of Gaussian noise is independent in intensity of signal at each pixel. In some color cameras used for digital acquisition of skin lesions, there is more amplification in the blue channel, which therefore may present more noise. Color features are particularly important in the diagnosis of skin cancer, where the presence of bluish structures is a strong indicator of the presence of the malignant melanoma.

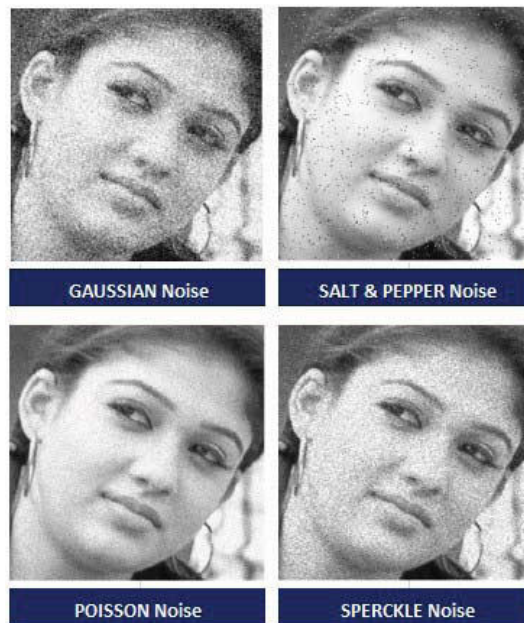


FIGURE 4.16: Visual Noise Effect

- *Salt-and-pepper noise*

It consists in the presence of light pixels in dark regions ("salt") and vice versa ("pepper"), thus obtaining the characteristic contrast between the pixels affected by noise and the surrounding ones. This type of noise can be caused by dead pixels, analog-to-digital converter errors, data transmission errors between the camera's sensor and its image processing units. This can be largely eliminated by subtracting the dark frame and interpolating around dark/bright pixels.

- *Poisson noise*

Poisson noise or shot noise is characterized by the granular characteristics of the electric charges. There is a shot noise in the thermionic emission of a cathode or in the current flowing through the junction of a semiconductor. Statistically it is appropriately described by a Poisson process. Shot noise is white up to frequencies of the order of the inverse of the electron transit time [194].

- *Speckle noise*

Speckle noise is a granular noise that inherently exists and degrades the image quality of the active radar and synthetic aperture radar. Speckle noise results from random fluctuations in the return signal from an object that is not larger than a single image processing element. As a consequence of this noise the average gray level of a local area is increased. The impact is generally severe, causing difficulties for the interpretation of the image.

The basic methods for de-noising an image can be classified as Spatial Filtering and Transform Domain Filtering [195]. Spatial filtering includes a predefined operation that modifies the grey value of each pixel according to the pixel values of square neighborhood centered at that pixel [196].

In order to provide a useful starting point for further analysis, we recall filters [193], [197] most used in pre-processing steps on dermatologic images. Mean filters (see Table 4.3), work best with Gaussian noise and could be effective for salt and pepper noise. Although these filters reduce the noise, they blur the image reducing sharp edges. Statistic filters (see Table 4.4) work well on speckle noise, and they allow the removal of noise while preserving edges.

<i>MEAN Filter</i>	<i>Characteristics</i>
Arithmetic Mean Filter (ArMF)	<ul style="list-style-type: none"> • This is the simplest of mean filters. • ArMF process computes the average value of the corrupted image in a particular area S. The value of the restored image at any point (x, y) is simply the arithmetic mean computed using the pixels in the region defined by S. • ArMF simply smoothes local variations in a image. Noise is reduced as a result of blurring: it can uniform the noise and works well with Gaussian noise.
Geometric Mean Filter (GMF)	<ul style="list-style-type: none"> • Each restored pixel is given by the product of the pixels in the sub-image window, raised to the power $1/m$. • A geometric mean filter achieves smoothing comparable to ArMF, but it tends to lose less image detail in the process.
Harmonic Mean Filter (HMF)	<ul style="list-style-type: none"> • HMF works well for Salt noise but fails for pepper noise. • HMF works well also with other types of noise like Gaussian noise.
Contra-harmonic Mean Filter (CMF)	<ul style="list-style-type: none"> • CMF is well suited for reducing the effects of salt and pepper noise. It cannot do both simultaneously. • CFM reduces to ArMF if $Q = 0$, and to HMF if $Q = 1$, where Q is the order of the filter. • It can preserve the edge and remove noise much better than ArMF.

TABLE 4.3: Mean Filter

Adaptive filters (see Table 4.5) work best when the noise is constant-power (“white”) additive noise like Speckle noise.

A second classification of de-noising methods is based on wavelet transforms resuming the Fourier transform. Wavelet transforms are defined as mathematical functions that analyze data based on scale or resolution [198]. In the particular context of digital image analysis on skin cancer, the most common filters applied by researchers to optimize images in the pre-processing phase are median filter, adaptive filter, middle filter and Gaussian smoothing filter [197], [199].

Restoration from blur

Many techniques and algorithms for image restoration are available, each of them with its own characteristics. In general, image restoration techniques are classified into two categories, blind image restoration and non-blind restoration [200], [201] depending on how well the noise degrading the image is known. Medical images can be affected by noise or blurring, making a correct diagnosis difficult for physicians. Restoring the degraded medical images becomes an important issue in doing reliable diagnosis. The blur is a kind of degradation which occurs when there is movement of the camera [202].

<i>STATISTIC Filter</i>	<i>Characteristics</i>
Median filter (MF)	<ul style="list-style-type: none"> • MF works by moving through the image pixel by pixel, replacing each value with the median value of neighbouring pixels. • MF is less sensitive to the extreme values. So, it can remove the outlier without reducing the sharpness of an image. • MF is an effective filter for salt and pepper noise: the original value of the pixel is included in the computation of the median. • MFs are quite popular because they provide excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size. • MF is widely used as it is very effective at removing noise while preserving edges.
Max and Min Filter (MMF)	<ul style="list-style-type: none"> • MMF blurs the image by replacing each pixel with the difference of the highest pixel and the lowest one respect to the intensity, within the specified window size. • This filter is useful to find the darkest points of an image.
Mid Point Filter (MPF)	<ul style="list-style-type: none"> • MPF is the best for random distributed noises such as speckle noise.
Gaussian smoothing Filter (GF)	<ul style="list-style-type: none"> • GF is used to blur images and remove noise and detail. • GF is a useful filter for smoothing and sharpening the image. • GF removes "high-frequency" components from the image (low pass filter).

TABLE 4.4: Statistic Filter

<i>ADAPTIVE Filter</i>	<i>Characteristics</i>
Adaptive local noise reduction filter (ALNF)	<ul style="list-style-type: none"> • ALNFs are capable of de-noising images that have abrupt changes in intensity. • ALNFs adjust its parameters during scanning the image to match the image generating mechanism. • ALNFs work better than mean filters and they can be used for random noises.
Adaptive Median Filter (AdMF)	<ul style="list-style-type: none"> • It can preserve the details of smoothing non impulse noise as the traditional median filter is not able to do. • AdMF performs spatial processing to determine which pixels in an image have been affected by impulse noise. • AdMF classifies pixels as noise by comparing each pixel in the image to its surrounding neighbor pixels. The size of the neighborhood is adjustable, as well as the threshold for the comparison.

TABLE 4.5: Adaptive Filter

There are several deblurring techniques such as the Lucy-Richardson algorithm technique, the Inverse filter, the Wiener filter and the neural network approach [201], [202].

In medical applications, the Wiener filter is one of the most powerful de-blurring techniques also used to remove noise. Wiener filter achieves noise reduction with some integrity loss of the speech signal. However, few efforts have been successful in showing the relationship between noise reduction and speech distortion [203].

Hair Removal

Recently, there has been an increase in the number of studies using imaging techniques to analyze melanocytic lesions and for mapping the total mole. An important issue regards pre-processing of dermatoscopic images, concerning the removal of hair. In fact, although the thin blood vessels and the cutaneous lines can be smoothed using the restoration filters referred in the previous section, the presence of short hair in the automated analysis of small skin lesions is an impediment capable of infecting the segmentation phase, and contributing to an inaccurate final diagnosis.

To remove thick hair in skin cancer images, researchers proposed methods based on mathematical morphology, on the detection of the curvilinear structure, on a inpainting based method, on Top Hat transformations combined with bicubic interpolation. In [204], the authors present an interesting review on hair removal techniques.

The objective pursued by the pre-processing phase of skin cancer detection systems, provides that the resulting images are distinguishable from the initial ones and are ready to feed the segmentation phase. In [205] DullRazor, a software tool to remove hair from images, was presented. DullRazor cleans the image using the following steps:

1. Identify dark hair positions using a generalized grayscale morphological closure operation.
2. Check the shape of the hair pixels as a thin and long structure and replace the verified pixels with a bilinear interpolation.
3. Smooth the replaced hair pixels with an adaptive median filter.

It has been tested on real dermatoscopic images with satisfactory results. Figure 4.17 (a) shows a lesion covered by thick hair and Figure 4.17 (b) shows the result after hair removal step.

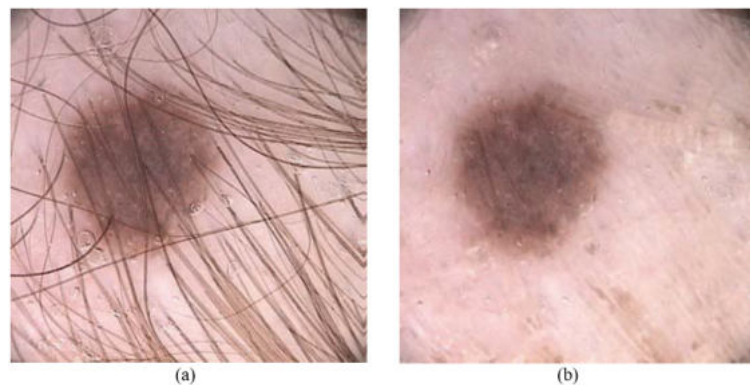


FIGURE 4.17: An image of nevus: (a) with hair, (b) after pre-processing step

An interesting comparative study of advanced hair repair methods is [204], where the authors propose an algorithm which does not disturb the tumor's pattern when repairing the hair pixels.

An accurate hair removal phase should be included in any Computer Vision Systems for melanoma detection as a pre-processing step, in order to repair the structure of melanoma images making them consistent with human vision.

4.3.3 Image Segmentation

During the segmentation step, the initial objective is to separate skin lesion (region of interest ROI) from healthy skin (Figure 4.18). Without the pretension to be exhaustive, we can report the following four types of segmentation methods:

- *Threshold base*, including methods such as the Otsu method, the local and global threshold, maximum entropy, histogram based methods, and so on.
- *Region-based*, growth of the sown region, watershed segmentation, are examples of this class.
- *Pixel-based*, including methods such as fuzzy c-means clustering, random field Markov, artificial neural network which is reinforcement algorithm.
- *Model-based*, such as the deformable parametric model, layer sets.

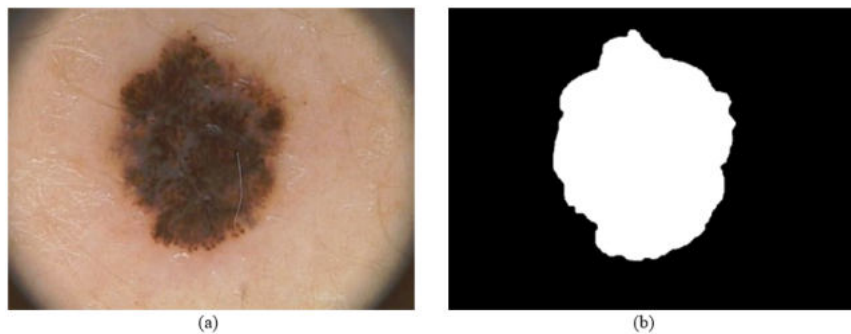


FIGURE 4.18: (a) original image (b) segmented image

The operation to identify the lesion can be done manually, based on the experience of dermatologists, or using software solutions for color segmentation like IMAGEJ and EDISON an edge detection and segmentation system of images. An accurate edge detection of skin lesion is essential to quantify the geometric characteristics of the tumor such as the area, the perimeter and the maximum diameter of the melanoma. Starting from dermoscopic image of a lesion, there are multiple features, both globally and locally, that can indicate whether the lesion is benign or malignant. There are many features used for the prediction of melanoma, including streaks [206], atypical pigment network[207], dots and globules[208], blue-white veil [209], granularity [210]. Even benign dermoscopy features can be equally important in automatic identification [211].

The contextual use of different criteria is the way for early detection of malignant melanoma and other types of skin cancer. Color, structure, shape, relative dimensions, positioning of the lesion, as well as their distribution in the lesion area are used in the segmentation of

clinical features [212]. Pre-processing methods are rigorously examined to find the right combination that best suits the targeted feature detection. We must not leave the possibility of using both visual observation and application of filters for image optimization.

Artifacts, such as dark peripheral regions and color wheels, should be avoided completely by masking them in advance. This can already be achieved in the initial step of lesion segmentation. Choosing the right combination of color channels for segmentation algorithm is just as important as choosing right segmentation approach.

Post processing is also crucial and should be carefully selected, based on the type of filtering necessary to achieve the best results. In fact, it should be kept in mind that the features used for the definitive classification of the lesion will be generated subsequently using the portions of original image.

4.3.4 Features Selection

The segmentation process aims to discard pixels from non-skin regions to simplify the image and to individuate the lesion. After this step, important usable features for melanoma detection are evaluated. Features extraction implies measurements inside image portions representing objects arising from the segmentation step.

In image analysis, the step of features extraction isn't often discussed enough in many publications, especially regarding the definition of features and the objective methods for measuring them. Applications frequently use statistical methods for extracting features or use filters, such as multi-channel filtering. In other applications we can appreciate solutions that provide principal component analysis (PCA) of a binary mask of the lesion, the co-occurrence matrix at the gray level (GLCM) [213], the Fourier spectrum [214], Gaussian derivative kernels [215] the transformation of the wavelet package (WPT) [216], and feature boundary extraction in order to reduce data redundancy [217].

The ABCD system [213], the 7-point checklist [218], 3-point checklist [219], model analysis [220] and Menzies method [221] are protocols that direct in deciding the features that can opportunely be extracted. Reliability in clinical diagnosis is very high for the ABCD rule, also with low computational burden [222]. On the other hand, comparative studies show that solution based on Menzies method achieves best sensitivity performance [223]. Therefore, many automated decision support systems use the ABCD rule or the 7-point checklist for feature extraction step.

Several studies have also demonstrated the effectiveness of the edge shape descriptors for the detection of malignant melanoma both in clinical and in computerized assessment methods [224]. The scientific community is trying to achieve Computer Vision System able to extract the features borrowed from the traditional diagnostic analysis protocols. The aim is to achieve a reliable classification of skin lesions identifying cases of malignant melanoma, dysplastic nevus or common nevus.

Conventionally the following methods are mainly used: ABCD rule of dermoscopy, Seven-point checklist, and Texture analysis. In the following paragraphs, we report the features used for each of these approaches.

ABCD Rule

the ABCD rule investigates the asymmetry (A), the border (B), the color (C) and the differential structures (D) of the lesion and defines the basis for a diagnosis by a dermatologist.

Currently the protocol has been extended to a larger version which also includes change or evolution (E) in color and/or size of lesion.

In Computer Vision System omitting the E-related information there are no significant worsening in terms of classification; so it's a good solution to leave the evolution or change of nevus (E) as a potentially usable feature for the specialist.

- A) *Asymmetry*

Dermatologists evaluate asymmetry comparing the two halves of the lesion according to the principal axis. Stoecker et al. [225] presented an algorithm capable of evaluating an asymmetry index of the lesion, looking at principal axis; in the case of a symmetric lesion, principal axis coincides with the symmetry ones. This index is evaluated defining a percentage of asymmetry, considering the smallest difference between the image area of the lesion and the image of the lesion reflected from the main axis.

Another method [226] considers the center of gravity of the lesion, and calculates the asymmetry index for the differences between the areas defined by the 180 axes. Asymmetry is a quantifiable property that can be used to discriminate and characterize melanomas.

Lengthening index is used to describe the lengthening and the anisotropy degree of the lesion. This index is related to eigenvalues λ' , λ'' of the inertia matrix, and is defined according to the relationship between the moment of inertia around the principal axis λ' and the moment of inertia around the secondary axis λ'' [227].

- B) *Border*

The edge identification of the lesion is a very critical step in the whole process of image segmentation, which aims at the separation of the skin lesion from healthy skin. For these purposes a series of segmentation techniques are referred in section 4.3.3. The most innovative approaches try to combine both color transformation and edge detection techniques, using algorithms able to capture active contours [228].

In [229] we find interesting comparisons of some of the most common features. Among these, we highlight: the larger diameter, area, edge irregularity, thinness ratio [230], circularity index (CIRC), variance of the distance between points of the frontal lesions to the centroid position [231] and symmetry distance (SD) [232].

Aspect ratio

The most common shape factor is aspect ratio, a function of the largest diameter and the smallest diameter orthogonal to it:

$$A_R = \frac{d_{min}}{d_{max}} \quad (4.1)$$

The normalized aspect ratio varies from approaching zero for a very elongated particle, such as a grain in a cold-worked metal, to near unity for an equiaxed grain.

The reciprocal of the right side of the above equation is also used, such that the A_R varies from one to approaching infinity.

Circularity Index

Another very common shape factor is isoperimetric quotient or circularity, defined as function of perimeter P and area A :

$$Circ = \frac{4A\pi}{P^2} \quad (4.2)$$

The circularity (4.2) of a circle is 1, and much less than one for a starfish footprint. The reciprocal of the circularity index is also used, such that f_{circ} varies from one to infinity.

Compactness shape factor

The compactness shape factor is a function of the polar second moment in of a particle and a circle of equal area A [233].

$$f_{comp} = \frac{A^2}{2\pi\sqrt{i_1^2 + i_2^2}} \quad (4.3)$$

The f_{comp} of a circle is one, and much less than one for the cross-section of a beam.

Elongation shape factor

The less-common elongation shape factor [233] is defined as the square root of the ratio of the two second moments of the particle around its principal axes.

$$f_{elong} = \sqrt{\frac{i_2}{i_1}} \quad (4.4)$$

Fractal dimension

Many methods exist to analyze the scale of the edge structures. Different studies have been carried out on images using fractal analysis [234]. This allows the repetition of the structure to be measured at a certain scale, and can be implemented on a grey-scale version of the image. The fractal dimension d is an essential parameter related to the n elements and the dilatation ratio $\frac{1}{k}$. The fractal dimension is given by:

$$n = \left(\frac{1}{k}\right)^{-d} \quad (4.5)$$

and

$$d_{frac} = \left(\frac{\log n}{\log k}\right) \quad (4.6)$$

Simmerty Distance

Simmerty Distance (SD) calculates the average displacement between a number of vertices when a transformation is applied that makes the original shape symmetrical. The symmetrical form closest to the original form P is called the symmetry transform (ST) of P . The SD of an object depends on the effort required to transform the original shape

into a symmetrical shape. It's calculated as follows (4.7):

$$SD = \frac{1}{n} \sum_{i=0}^{n-1} \|P_i - \hat{P}_i\| \quad (4.7)$$

Emphasis is also placed on the features that quantify the transition from the lesion to the skin [234]. Features like minimum, maximum, average, and variance responses of the gradient operator applied on the intensity image along the lesion border are used for these measures.

Thinness Ratio

The thinness ratio and polygon area can be used to define how large or small the gap or overlapping area can be considered a "sliver". Thinness is often used to define the regularity of an object. After calculating the area (A) and the perimeter (P) of an object, we can define the thinness ratio as:

$$TR = 4\pi \frac{A}{P^2} \quad (4.8)$$

This measure takes a maximum value of 1 when a circle is considered. Objects of regular shape have a higher TR than similar irregular ones. Generally, TR 's value is less than 1.

- C) Color

The colors within the lesion are identified: areas of light brown, dark brown, black, red are indicative of vascularized zones, clear areas are relative to healthy skin and areas with slate blue hues are strongly indicative of malignant lesions [235]. Usual situations involves the use of RGB images that can be managed through the individual red, green and blue color channels. Other color channels such as cyan, magenta, yellow (CMY) can also be taken into consideration that is to say the notation hue, saturation, value (HSV):

$$\begin{cases} I = \frac{(R+G+B)}{3} \\ S = 1 - \frac{3}{(R+G+B)} \cdot \min(R + G + B) \\ W = \arccos \frac{R - \frac{1}{2}(G+B)}{\sqrt{(R-G)^2 - (R-B)(G-B)}} \end{cases} \quad (4.9)$$

where hue is defined as follow:

$$H \begin{cases} W \rightarrow \text{if } G > B \\ (2\pi - W) \rightarrow G < B \\ 0 \rightarrow \text{if } G = B \end{cases} \quad (4.10)$$

In [235] the spherical coordinates defined for the pixels belonging to the lesion are used. These are typically defined in the following equations where A and B are angles:

$$\begin{cases} L = \sqrt{R^2 + G^2 + B^2}, \\ A = \cos^{-1}\left(\frac{B}{L}\right), \\ B = \cos^{-1}\left(\frac{R}{L \cdot \sin A}\right) \end{cases} \quad (4.11)$$

The change of colors can be appreciated by measuring the minimum deviations, maximum, averages and standards of the values of the selected channels and through the intensity of the color, enhancing the chromatic differences within the lesion [211], [236]. Another valid method for estimating skin colors based on the normal skin structure model is presented in [237].

- *D) Differential Structures*

The structural components present are identified and counted and a numerical score is assigned and based on their presence. Typically the focus is on the pigment network, on the points, on the globules, and on the strips. Areas without a structure are taken into consideration if they are equal to at least 10% of the surface of the lesion [238].

Seven Point CheckList

The seven-point checklist (7PCL) [166], [239] is a detection protocol made up of seven criteria that take into account both the chromatic characteristics and those related to the shape and structure of the lesion (see Figure 4.19). The dermoscopic image of a melanocytic skin lesion is analyzed in order to highlight the presence of these standard criteria and starting from the assumption that everyone influences the final evaluation with a different weight.

If the resulting total score it's bigger than three, this indicates that the injury is malignant, directing the patient to a specialist's visit.

The 7PCL was revised in 1989 to attribute to three factors a greater weight (change in size, shape and/or color) leaving unchanged the weight of the remaining four; the total score was weighted by considering 2 for the priority factors and 1 for the remaining ones. The threshold value remained at 3 points.

Original 7PCL	Weighted 7PCL
<ul style="list-style-type: none"> • Change in size of lesion • Irregular pigmentation • Irregular border • Inflammation • Itch or altered sensation • Larger than other lesions ($d > 7mm$) • Oozing/Crusting of lesion 	<ul style="list-style-type: none"> • Major features (2 points) <ul style="list-style-type: none"> • Change in size of lesion • Irregular pigmentation • Irregular border • Minor features (1 point) <ul style="list-style-type: none"> • Inflammation • Itch or altered sensation • Larger than other lesions ($d > 7mm$) • Oozing/ crusting of lesion

FIGURE 4.19: Original and Weighted 7-Point Check List

Texture Analysis

Through textures analysis the information concerning the lesion plot is taken. The features extracted relative to the textures are typically statistical and structural. Statistical methods define the plot in terms of local statistics by focusing on the gray levels that, in a particular

area of the lesion, can remain constant or vary more or less slowly. Among the most used textural features, we recall the following [240]:

- *Neighboring gray-level dependence matrix (NGLDM)*

NGLDM is used to detect the presence of the pigmented network on skin lesions [240]. The dissimilarity d is a measure correlated to the contrast by a linear increment of weights, which becomes concrete as we move away from the diagonal of the *co-occurrence matrix of the gray level (GLCM)*. A formulation of dissimilarity is:

$$d = \sum_{i,j=0}^{N-1} P_{i,j} |i - j| \quad (4.12)$$

in which i is the row number, j is the column number, N is the total number of rows and columns of the GLCM matrix, and

$$P_{i,j} = \frac{V_{i,j}}{\sum_{i,j=0}^{N-1} V_{i,j}} \quad (4.13)$$

is the normalization equation in which $V_{i,j}$ is the digital number (DN) value of the cell i, j in the image window (i.e., the current gray-scale pixel value).

- *Angular second moment(ASM)*

ASM, which is a measure related to orderliness, where $P_{i,j}$ is used as a weight to itself, is given by:

$$ASM = \sum_{i,j=0}^{N-1} i \cdot P_{i,j}^2 \quad (4.14)$$

- *Gray-Level Co-Occurrence Matrix (GLCM) mean*

GLCM mean, μ_i , which differs from the familiar mean equation in the sense that it denotes the frequency of the occurrence of one pixel value in combination with a certain neighbor pixel value, is given by:

$$\mu_i = \sum_{i,j=0}^{N-1} i \cdot P_{i,j} \quad (4.15)$$

- *Gray-Level Co-Occurrence Matrix (GLCM) standard deviation*

GLCM standard deviation, σ_i , which gives a measure of the dispersion of the values around the mean, is given by:

$$\sigma_i = \sqrt{\sum_{i,j=0}^{N-1} P_{i,j} (i - \mu_i)^2} \quad (4.16)$$

The researchers that seek to automatically identify skin lesions exploit the available computational capabilities using many of the features stated before, looking also to new ones.

4.3.5 Classification

The continuous proposals of new algorithms and techniques for the classification of dermoscopic images highlight the need to summarize and compare algorithms used for the classification of skin lesions.

We provide a road map of the classification algorithms referring also to new paradigms of artificial intelligence that are interesting for the implementation of increasingly reliable solutions [241]. We refer specifically to the Multiple Instance Learning approaches and the Deep Learning paradigm.

In the classification step, the information extracted in the previous phases is used to produce diagnosis on the dermoscopic images. A first possible response is to provide a probability value, which describes the probability of belonging to a specific class. A second possible result is a dichotomous distinction between the two classes of melanoma and benign nevus.

Models such as logistic regression, artificial neural networks, K-nearest neighbor and decision trees are all members of the first approach, while the models most commonly used for binary classification include supporting vector machines. For a detailed description of a specific classification approach, refer to the cited references.

K-Nearest Neighbour Algorithm.

The K-nearest neighbor classifier [242] is an important non-parametric method used in various contexts for pattern recognition. When considering a lesion belonging to the test set, this algorithm returns the K-vectors closest to the representative vector of the lesion managed in the training set. The unlabeled sample is then assigned to the class represented by most of the nearest neighbors. The operating principle is very simple as long as the classifier can rely on a large and representative training set of the whole range of measurements that can be expected from each class.

Some studies focus on the optimization of the selection procedures, on the definition of the weight and the choice of the features with the aim at improving the performance of the classifier [243]. K-nearest neighbor method uses data directly for classification, without first providing for the creation of a model [244]. The only adjustable parameter in the model is k , which is representative of the number of class members among the closest neighbors. Decreasing k the model becomes more flexible, vice-versa its improvement corresponds to a more rigid model: as well, with this model the tuning phase is particularly delicate and related to the specific application. The choice of k can only be defined empirically taking inspiration from application contexts similar to the one considered. Through this model the specialists may refer to "similar" cases to those at hand, also comparing lesions to be investigated with other similar and already known lesions.

Interesting applications of this classifier are in [245], where a system for computerized analysis of images obtained from Epiluminescence Microscopy (ELM) has been developed to enhance early recognition of malignant melanoma. Through the use of shape features, they have obtained interesting classification performance: sensitivity of 87% with a specificity of 92%.

In [246], K-nearest neighbor was first independently used, and later in combination with other methods making a collaborative decision support system. To make the use of this method effective, it is necessary to define a specific metric for the distance between data

elements. Even, in this case, there are no general rules: the data structure is reflected in the choice of the metric [247].

Decision Trees

As introduced in chapter 1, the decision tree approach is one of the most widely used techniques of supervised machine learning. The model identifies a variable and a threshold value in the domain of the chosen variable, to divide the data set by maximizing the separation using a tree structure [248]. The best choice of variable and threshold is the one that minimizes disparity measurements in the resulting groups. The most common adopted criterion is the "information gain"; this means that entropy reduction due to this division has to be maximized. Various versions of decision trees, such as ADWAT and LMT, are also used for the classification of dermoscopic images.

The use of decision trees in medicine is privileged by the simplicity of the model that may be easily expressed by a set of rules. Thus, the specialist is facilitated in understanding the second opinion provided by a software predictor. The major disadvantage of decision trees is linked to the greedy construction process which, on large training sets, generates complex decision rules that behave well on training data but do not generalize well to unseen data [249]. In such cases, the classifier model highlights the limitation related to excessive overfitting to training data.

In [250], a useful framework was proposed. Texture features, such as Energy, Entropy, Contrast [251], [252], are extracted from the segmented image using the gray level recurrence matrix and are used to detect the presence of skin diseases. The proposed framework classifies images in specific classes such as melanoma, leprosy or eczema, using the decision tree technique that proves to be performing.

In [253], the role of the selected features and their combination to carry out classification of skin lesion is emphasized. The MED-NODE data set is used, first by proceeding with a pre-processing step of the images to remove the artifacts present. Various color features are used and system performance is verified using Naïve Bayes, Decision Tree and K-nearest neighbor. The system achieves an accuracy of 82.35% on the decision tree which is greater than other classifiers.

In [254] the Naïve Bayes and Decision Tree techniques are used to diagnose malignant melanoma using as training set DermIS and DermQuest images. The results show that the accuracy rate of the Decision Tree is over 92%, while Naïve Bayes offers a 98% higher accuracy rate. Comparative results indicated that the proposed techniques have excellent accuracy compared to other techniques in the field of melanoma diagnosis.

Logistic Regression

Logistic regression is an algorithm that constructs the separating hyperplane between two data sets, using the distance from the hyperplane as a probability of belonging to the class. The model works with linear parameters and has found widespread use in medical applications, although it can only be used to calculate linear decision boundaries. The easily use of the model facilitates the interpretation of the results in terms of probability supporting the choice of class membership. Often this model is used to compare the performances of other classifiers also able to determine linear and non-linear separation surfaces [247].

In [255], a study is conducted to associate the risk factors for malignant melanoma with their position on the body. The multivariate poly-symptomatic logistic regression was used to determine whether the risk factors differed among the anatomical sites. The risk factors examined demographic and pigmentary characteristics, factors related to sun exposure, anatomical site-specific sunburn. A history of sunburn at an anatomical site was specifically related to the development of malignant melanoma at that site more than at other sites. Age and gender were the only risk factors that differed significantly between the anatomical sites. The different age was explained by the differences in the histological subtype between the sites. In this case, a fully use of linear regression was made for specific statistical surveys.

Considering degenerative steps in tumor progression and the multi - variable logistic regression, in [256] a prognostic model for survival prediction related to primary cutaneous melanoma has been developed.

The authors used histological criteria, assigning the melanomas to the stages of tumor progression. These phases were the *in-situ* and invasive radial growth phase and the vertical growth phase. After appropriate follow-up phases, 122 invasive tumors in the phase of radial growth without metastases were found. The survival prediction of the affected patients has been improved thanks to the use of the adopted multivariable logistic regression model.

In [257], the authors present a robust "immunoscore model" to predict the "anti-PD1" response of melanoma, that is a dedicated therapy for the treatment of melanoma. Considering that only a subset of patients with melanoma get benefits from PD1 inhibitors, the use of a selection operator (LASSO) logistic regression allows to construct an "immunoscore" based on the fraction of immune subsets and adopted gene set enrichment analysis. This study aims to construct predictors to identify responders to anti-PD1 therapy.

Artificial Neural Network

Artificial Neural Network (ANN) is one of the vital parts of soft computing [258]. ANNs are made up of small processing units, the so-called artificial neurons, capable of processing elementary information, and are organized in highly interconnected schemes training to mimic the human brain.

ANNs are used in various biomedical fields [10] also considering the use of the increasingly numerous data sets today available [259]. This model proves able to learn complex concepts that are potentially too complex for human detection. First implementations required an important effort for tuning to obtain satisfactory results; over time the Bayesian methods [260] and the implementations of faster learning algorithms [261] allowed to use sophisticated methods solving the tuning parameters task. In this regard, various techniques and clustering algorithms based on neural networks are used including the back propagation network (BPN), radial basis function network (RBF) and extreme learning machine (ELM) [262]. The use of neural networks is also widespread for the proposals of Computer Aided Diagnosis system nowadays presented by the scientific community.

In [263], the authors propose a framework to facilitate early recognition of skin cancer by applying the first-order extraction method using contrast, variance, standard deviation, kurtosis, mean and smoothness. The proposed solution uses a multilayer Perceptron neural network (MLP NN): the results show an overall accuracy rate of 83.86%. The most recent best-performing frameworks, like the one presented in [264], are based on Convolutional

Neural Networks (CNNs), a specific ANNs architecture, which represents the state-of-the-art in any field of image processing, and it is characterized by several key properties that decreed its success. More in detail, CNNs are a class of deep, feed-forward artificial neural networks, usually applied, but not limited to, 2d images, or rather to data representable in a grid structure. Taking inspiration from biological processes [265], CNNs mimic the visual cortex, a region of the brain in charge of elaborating electrochemical signals generated by the photo-receptors that constitute the cornea (i.e., our receptive field).

In addition to the advantages of general ANNs, CNNs are shift-invariant or space invariant, thus inherently more efficient thanks to their shared-weights architectural feature and being translational invariant [265], [266].

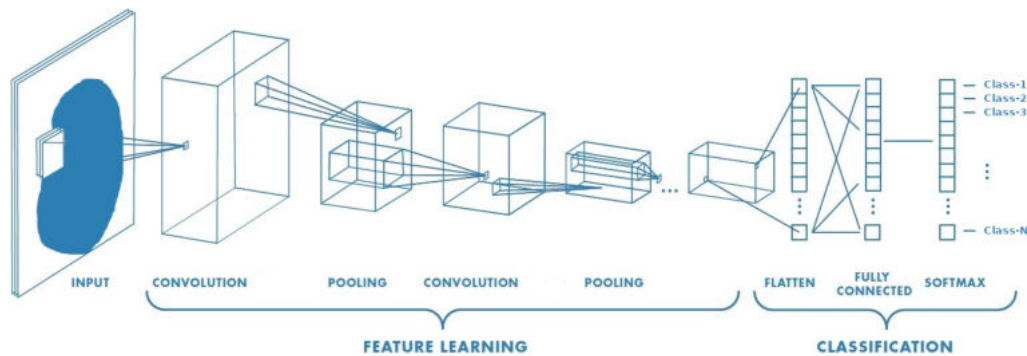


FIGURE 4.20: Example of CNN architecture

In Figure 4.20 CNNs take input images and transform them using convolutional, pooling, and fully connected layers, into flattened vectors. The elements of the output vector (softmax layer) represent the probabilities of the presence of disease. During the training process, the internal parameters of the network layers are iteratively adjusted to improve accuracy. Prediction tasks include both classification of the images as well as localization of medical features such as tumors.

During the last few years, further advances in hardware (i.e., more powerful GPUs), software (e.g., Keras, Tensorflow 2.0, etc.), neural network architectures, and data availability enable researchers to test the capabilities of deep learning models and achieve promising results in classification in medical image processing [267]. In particular, the architecture of choice is based on CNNs, which is based on the main assumption that the input image is compositional, meaning that it is formed of patterns that are local, stationary, and multi-scale. The resulting key properties of CNNs, which ultimately decreed their success, are: the ability to recognize the same pattern in different shapes, positions, and orientations within the image (i.e., translational invariance), the activation of a neural path is based on the detection of a familiar pattern (i.e., locality), and the ability to learn increasingly complex abstract structures in a hierarchical way (i.e., multi-scale) [268], [269].

Leveraging transfer-learning, i.e., loading pre-trained model trained on similar tasks, and then fine-tuning the model on the task at hand, is a common and practical way of reducing training time and achieving better overall performance. For instance, in [269], Liao presented a universal skin disease classification method by selecting models that proved to have very good performance in the ILSVRC-2014 competition (i.e., VGG16, VGG19, and GoogleNet), thus trained on the ImageNet datasets. These pre-trained models are then fine-tuned on the DermNet dataset, which contained more than 23000 skin disease images on a wide variety

of skin conditions. In the experimental evaluation, performed on the DermNet and OLE datasets, the proposed method achieved 73.1% Top-1 accuracy and 91.0% Top-5 accuracy on the Dermnet dataset, and 69.5% Top-5 accuracies on the OLE dataset.

Ercal et al. [270], introduced a skin cancer classification framework based on tree-based hierarchical neural networks and fuzzy logic which leverages morphological features.

Whereas, in [271] they developed a method to detect skin tumors using only color features, achieving the highest accuracy, in the case of “intra-dermal nevus”, and the lowest in the case of “melanoma” images.

Other examples in which transfer learning is being used in order to limit training time and trying to achieve better performance are [272], [273]. In the former, a pre-trained model, trained on the *Kaggle Challenge for Diabetic Retinopathy Detection dataset*, is being used for melanoma screening. However, their experimental evaluation suggests that the experimental design is sensitive to the type of skin lesions (benign or malignant). Similarly, to the previous works, also Kawahara et al. [274] use a pre-trained model for a skin lesions detection task, reducing training time, and achieving 85.8% accuracy in the 5-classes testbed.

A powerful tool in the case of lack of data, or unbalanced datasets, is data augmentation. In [275], the performance of a CNN architecture is compared between a non-augmented dataset and augmented dataset for the classification of skin lesions. In fact, the experimental evaluation showed that the network trained on the augmented dataset achieves a better accuracy rate than the one trained on the non-augmented data.

Conversely, in [276] Fatima et al. rather than relying on deep learning paradigms, they proposed a methodology dubbed MPECS - Multi-Parameter Extraction and Classification System, based on twenty-one handcrafted features, and using six extraction phases. These features are then analyzed based on statistical methods for early detection of skin cancer melanoma. Examples of extracted features are lesion borders, color, symmetry, area, perimeter, and eccentricity. Their study demonstrated singular statistical analysis from extracted features is not sufficient to accurately classify the skin lesions. Therefore, advanced classification methods are needed to achieve good performance for the classification task.

Support Vector Machines

Among the supervised learning models, a special role is played by Support Vector Machines (SVM) which are models used for both classification and regression analysis. Given a set of training examples, considering that each example belongs exclusively to one between two classes, an SVM training algorithm constructs a model capable of assigning the new examples exactly to one of the considered classes. SVM is a linear binary classifier used for dichotomous classification without indicating the probability of belonging. However, in some versions, such as the one with the “Platt scaling”, it may be possible to also have probabilistic classifications [277].

With an SVM model it is possible to get a representation of the examples as points in space. Specific separation surfaces allow to divide the data of a class from those belonging to the other class. Particularly in medical context, the use of SVM methods allows us to obtain benign versus malignant classifications, which are of particular interest for the implementation of software tools supporting diagnostics. Regarding the mathematical model SVM, it creates optimal boundaries of separation between data sets by solving a constrained quadratic optimization problem [36]. In addition to performing linear classification, SVM

can effectively perform a non-linear classification using kernel trick; with this mathematical artifact, input data are mapped into vector spaces with larger dimensions, where a linear classification is possible (see Figure 4.21).

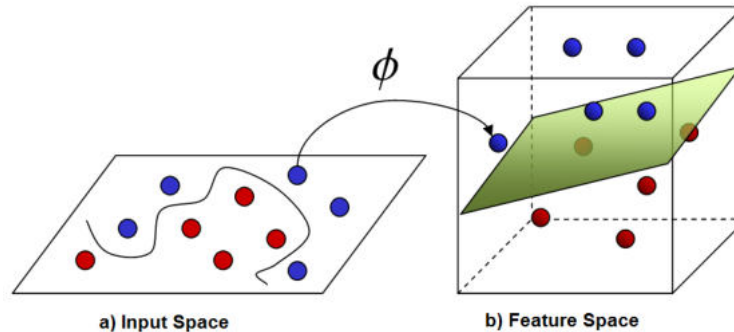


FIGURE 4.21: Kernel trick ϕ : from a) Input Space to b) Feature Space

With unlabeled data an unsupervised learning approach is required. The clustering SVM algorithm, created by Siegelmann and Vapnik [278], is one of the most used clustering algorithms that applies support vector statistics to classify unlabeled data. SVMs have numerous advantages over more classical classifiers such as decision trees and neural networks.

SVMs training mainly involves optimizing a convex cost function. Therefore, there is no risk of getting stuck at local minima as in the case of back-propagation neural networks.

SVMs are based on the principle of structural risk minimization which minimizes the upper bound of the generalization error. Thus, SVMs are less prone to over-fitting than algorithms such as back-propagation neural networks that adopt the empirical risk minimization principle. Another advantage of SVMs is that they provide a unified framework in which different machine learning architectures can be generated through specific kernel choice [226]. Concerning the automatic classification of skin lesions, SVMs are one of the most widely used tools even in combination with other machine learning techniques [279], being able to also benefit from some mathematical techniques, like Lagrangian relaxation, which allow to obtain good classification accuracy rate [280], [281].

Multiple Instance Learning techniques

A MIL problem consists in classifying sets of items. As we have discussed in Chapter 2, in the MIL terminology, such sets are called *bags* and the items inside them are named *instances*. This design problem is referred as the so-called MIL standard assumption, consisting in the following: a bag is positive if it contains at least a positive instance and it is negative if it does not contain any positive instance [83].

The MIL paradigm fits very well with image classification: in fact, to classify an image containing a particular subject, one needs to look only at some subregions (instances) of the image (bag). In this perspective the MIL paradigm works very well with respect to a classical supervised approach because it obtains global information from local detection.

Among the various application contexts, Multiple Instance Learning approaches are particularly effective for diagnostics through medical images and videos in which local analysis is relevant [88].

In [282], a multi-instance learning framework was inserted to solve the problem of recognizing the features of skin biopsy images. Other approaches based on color features have

not been able to directly recognize the characteristics of skin biopsy images due to the color changes present in the images. Through the multi-instance learning approach the authors used texture features to express each instance as a vector expression. Therefore, through the application of multi-instance learning algorithms, the proposed method showed to be effective and acceptable for medical analysis.

In [280], we present an original application to melanoma detection of a MIL approach. Through the application of a MIL algorithm on some clinical data constituted by color dermatoscopic images, we discriminate between melanomas (positive images) and common nevi (negative images). In comparison with standard classification approaches, such as the well known support vector machine, the proposed method performs very well in terms both of accuracy and sensitivity. In particular, using a leave-one-out validation on a data set constituted by 80 melanomas and 80 common nevi, we have obtained good performance of classification (accuracy = 92.50%, sensitivity = 97.50% and specificity = 87.50%).

MIL technique could be at the basis of more sophisticated tools useful to physicians in melanoma detection.

Chapter 5

MIL Models application in Automated Melanoma Detection

*"You see things; and you say 'Why?' But I dream things that
never were; and I say 'Why not?'"*

– George Bernard Shaw

Machine learning (ML) algorithms have a great impact in the field of medical imaging. The counterpart of the growing size of medical imaging data sets, is the significant lack of labelled data.

Manual labeling of images is expensive and time-consuming and such labels may not be necessary in clinical practice, thus restricting the amount of labeled data only to research studies. The lack of labeled data motivates approaches that go beyond traditional supervised learning using other types of data and / or labels that could be more easily accessible.

Machine learning has become very important in the field of medical image analysis. Activity such as segmentation, in which each pixel in an image is assigned to a different anatomical structure or type of tissue and computer aided diagnosis in which a category label for an entire image is expected, now are exclusively done by machine learning methods.

5.1 Classification performances of ML methods on dermo- scopic images

In [139], the authors have pointed out that among the emerging approaches of machine learning methods, semi-supervised learning, multiple instance learning and transfer learning should be included.

The first goal that we tried to achieve in our work has been to verify how a MIL approach can be effectively applied for the classification of medical images related to skin lesions.

In particular, we have considered the case of the diagnosis of melanoma mainly for two reasons: growing importance of the disease on a global scale, and absence in the literature of specific application of MIL algorithms to this particular domain. The study of the dermatological field and of some related optimization models, have guided us along the experimental path that we will describe in Section 5.3.3, where we have applied a very recent MIL model (see Section 5.3.1), to classification of dermatological images.

To highlight the tasks to be addressed we refer to [2], where the classification performances of the most significant ML applications currently in literature are summarized.

The problem of automatic detection of melanoma has been addressed through different approaches. The methods developed were mainly based on two different screening imaging techniques, clinical or dermoscopic images. The clinical imaging modality has been replaced by dermatoscopy (also known as epiluminescence microscopy).

In [2] the authors show a summary of the results of the most significant methods that we summarize by means of Figure 5.1 where for every considered application the following values are reported:

- Sensitivity percentage (in **blue**);
- Specificity percentage (in **black**);
- Size of data set specifying number of melanoma images over total number of images (in **red**).

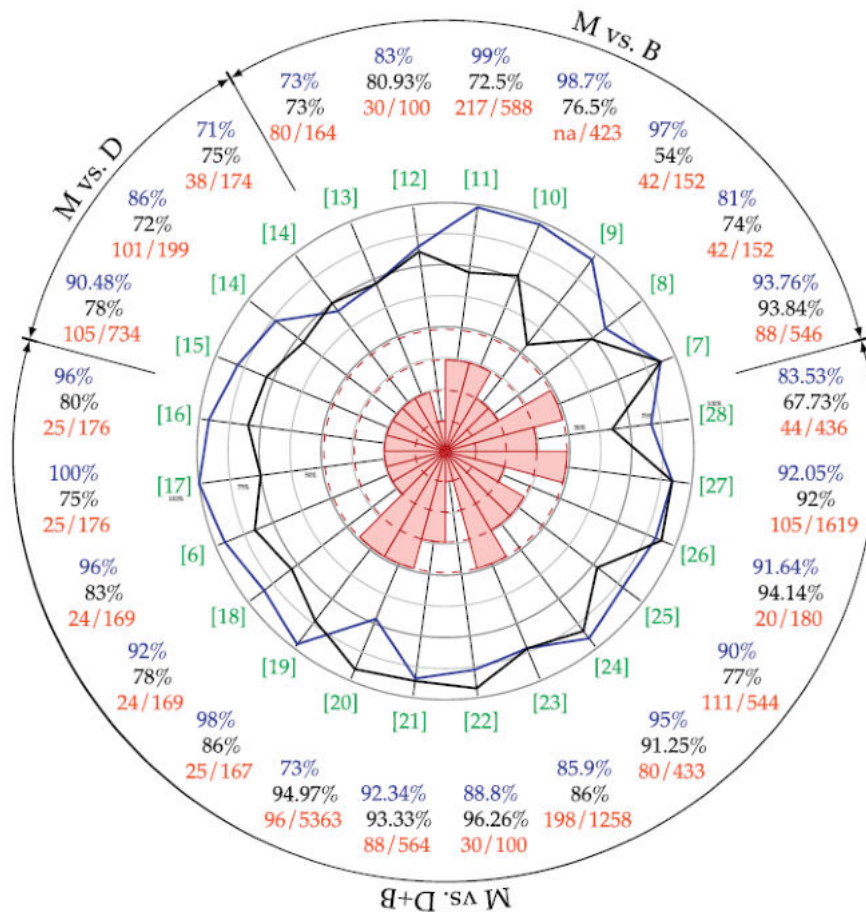


FIGURE 5.1: Summary of the classification performances of the methods reviewed from the dermoscopic imaging literature [2]

In the same figure, a comparison of the size of data set is also presented in the middle of the graph, which contains four levels. For a formal definition of sensitivity and specificity parameters see Appendix A.

As is known, the *sensitivity* refers to the number of cancers correctly identified with respect to the total number of cancers while the *specificity* indicates the number of negative cases with respect to the total number of non-cancer cases. Still in Figure 5.1, the

methods with the relative results can be partitioned in function of their classification task: melanomas against benign ($M vs B$), melanomas against benign and dysplastic ($M vs (B+D)$) and melanomas versus dysplastic nevi ($M vs D$). Figure 5.1 shows the scarce attention that has been paid to the discrimination of melanomas from dysplastic nevi and how the task of dysplastic nevi against common ones has not been evaluated.

These last two tasks are more complex, both for specialists and for automatic detection, given the similarity between the lesions that we want to discriminate.

5.2 An overview of numerical experiments

In the following we describe our experiment performed on two types of data sets:

1. A dermatoscopic data set named PH^2 containing 200 melanocytic lesions images: 80 common nevi, 80 atypical nevi and 40 melanomas. All of them have been obtained in 8-bit RGB color with a resolution of 768×560 pixels [3].
2. A data set of plain photographs publicly available from two online databases <https://www.dermquest.com> and <http://www.dermins.net1>, used in [4].

In particular, we have focused on the following classification tasks:

- Melanomas vs Benign Nevi
- Melanomas vs Dysplastic Nevi
- Melanomas vs Dysplastic and Benign Nevi
- Dysplastic Nevi vs Benign Nevi

We highlight that looking at Figure 5.1, at the best of our knowledge, no results are presented in the literature related to classification of Dysplastic Nevi versus Benign Nevi. For both data sets we have applied two new algorithms, MIL-RL which is described in the next section, and DC-SMIL introduced in Chapter 3. We have used two different approaches in terms of features selection:

- Color features
- Color and Texture features

In most of our experiments we have not used any pre-processing procedure, except for some specific cases aimed at verifying possible improvement of the classification performances [283]. All the reported results are expressed in terms of correctness, specificity, sensitivity, F-score and CPU time. For a formal definition of these measures please refer to Appendix A. We have compared our results with the well-known SVM technique using both the linear and the RBF kernels. All approaches have been implemented in Matlab and the corresponding codes have been run on a Windows 10 system characterized by a 2.21 GHz processor, except for DC-SMIL for which we have adopted the Java implementation of Algorithm DCPCA, running the computational experiments on a 3.50 GHz Intel Core i7 computer.

5.3 The MIL-RL algorithm

In this section we describe a very recent MIL algorithm named MIL-RL, proposed by Astorino et al. in [96], and based on a Lagrangian relaxation approach to the SVM type MIL model introduced by Andrews et al. in [95]. We have chosen this model to verify whether the application of a MIL model can effectively return interesting classification performance for automatic melanoma detection. To the best of our knowledge a MIL approach has never been applied before to this specific task.

The section is organized as follow. In Subsection 5.3.1 we introduce the MIL model at the basis of the MIL-RL approach, while in subsection 5.3.2 we report the main steps of the algorithm and in subsection 5.3.3 we describe the related numerical experiments.

5.3.1 The model

Given p points (the instances) $x_j \in \mathbb{R}^n$, $j = 1, \dots, p$, we indicate by $y_j \in \{-1, 1\}$ the corresponding class label, supposed to be unknown. The set of instances is partitioned in $m + k$ bags, where m is the number of positive bags, identified by the index set $J^+ = \{J_1^+, \dots, J_m^+\}$, and k is the number of negative bags, identified by the index set $J^- = \{J_1^-, \dots, J_k^-\}$. The class label of each bag is known: it is equal to +1 for each positive bag J_i^+ , $i = 1, \dots, m$, and it is equal to -1 for each negative bag J_i^- , $i = 1, \dots, k$.

We recall our MIL assumptions: a bag is positive if it contains at least a positive instance, while it is negative when it contains only negative instances.

The construction of the classifier consists in computing a separating hyperplane

$$H \triangleq \{x \mid w^T x + b = 0\}$$

between the two classes of bags and, at the same time, to assign a class label $y_j \in \{-1, 1\}$ to all the instances in the positive bags. In fact, since a negative bag is defined containing only negative instances, only the class label of the instances inside the positive bags are actually unknown. Taking into account that the objective is to minimize a measure of the classification error of all the instances inside the bags, we come out with the following constrained, nonlinear, nonconvex, mixed integer problem [95]:

$$P \begin{cases} z^* = \min_{w,b,y} f(w,b,y) \\ \sum_{j \in J_i^+} \frac{y_j + 1}{2} \geq 1, \quad i = 1, \dots, m \\ y_j \in \{-1, 1\}, \quad j \in J_i^+, \quad i = 1, \dots, m, \end{cases}$$

where

$$f(w,b,y) \triangleq \frac{1}{2} \|w\|^2 + C \sum_{i=1}^k \sum_{j \in J_i^-} \max\{0, 1 + (w^T x_j + b)\} \\ + C \sum_{i=1}^m \sum_{j \in J_i^+} \max\{0, 1 - y_j (w^T x_j + b)\}.$$

The objective function f is very similar to the classical Support Vector Machine (SVM) objective function and it is the sum of three terms: by the first one we maximize the margin, with the second one we minimize a measure of the classification error of the points belonging to the negative bags and by the last term, we minimize a measure of the classification error of

the points belonging to the positive bags. Note that the presence of the unknown labels y_j in the third term allows the possibility to allocate some points of positive bags in the negative part with respect to the separating hyperplane.

Finally constraints

$$\sum_{j \in J_i^+} \frac{y_j + 1}{2} \geq 1, \quad i = 1, \dots, m \quad (5.1)$$

impose that for each positive bag, at least one point must be labelled as a positive point.

5.3.2 The algorithm

The MIL-RL approach is based on the solution of the Lagrangian relaxation [284] problem of P , obtained by relaxing the linear constraints 5.1.

It holds:

$$P_{LR}(\lambda) \begin{cases} z_{LR}(\lambda) \triangleq \min_{w,b,y} f_\lambda(w,b,y) \\ y_j \in \{-1,1\}, \quad j \in J_i^+, \\ i = 1, \dots, m, \end{cases}$$

with

$$f_\lambda(w,b,y) \triangleq f(w,b,y) + \sum_{i=1}^m \lambda_i \left(1 - \sum_{j \in J_i^+} \frac{y_j + 1}{2} \right),$$

where $\lambda \geq 0$ is the vector of the Lagrangian multipliers in \mathbb{R}^m .

Problem $P_{LR}(\lambda)$, whose optimal solution is denoted in the sequel by $(w(\lambda), b(\lambda), y(\lambda))$, provides a lower bound for problem P , that is:

$$z_{LR}(\lambda) \leq z^*,$$

for any $\lambda \geq 0$. Moreover the Lagrangian dual problem is defined as:

$$P_{LD} \begin{cases} z_{LD} \triangleq \max_{\lambda \geq 0} z_{LR}(\lambda) = \max_{\lambda \geq 0} \min_{w,b,y} f_\lambda(w,b,y) \\ y_j \in \{-1,1\}, \\ j \in J_i^+, \quad i = 1, \dots, m, \end{cases}$$

with, of course, $z_{LD} \leq z^*$.

In [281] it has been proven that the duality gap between the optimal values of the objective functions of problems P and P_{LD} is equal to zero, that is $z_{LD} = z^*$; moreover, in solving P_{LD} one gets also an optimal solution to P .

This important result suggests the use of a dual ascent method for solving problem P_{LD} in order to obtain an optimal solution to problem P . Since problem P_{LD} is a nondifferentiable optimization problem, the use of nonsmooth optimization techniques, such as the subgradient method [285] or any bundle type method [157], [286]–[290] is in order. In such context the evaluation of the objective function $z_{LR}(\lambda)$ requires in turn solution of problem $P_{LR}(\lambda)$. Getting an exact solution would severely impact on the efficiency of the method and, consequently, we resort to a Block Coordinate Descent (BCD) algorithm (see [291]) where, at each iteration, we alternately fix the value of vector y and of the couple (w, b) . Note that, in fact,

for any $\lambda \geq 0$, when variables y_j are kept fixed, problem $P_{LR}(\lambda)$ reduces to solving a classical SVM quadratic program; vice-versa, when the couple (w, b) is fixed, it is possible to solve it, with respect to y , by inspection.

The heuristic approach consists in applying a subgradient strategy to maximize function z_{LR} . We refer to each iteration of such process as an *outer iteration*. As stopping criterion of the subgradient procedure, we adopt the maximum number of outer iterations.

Inside each outer iteration, calculation, even approximate, of z_{LR} is performed by the BCD algorithm, an iterative process that terminates either when substantial improvement in calculation of z_{LR} is no longer achieved, or when a maximum number of inner iterations has been reached.

It is worth noting that whenever BCD provides a solution of $P_{LR}(\lambda)$ infeasible for the original problem, such solution may be easily modified to recover feasibility, thus providing an upper bound on the optimal solution of P too. Such upper bound, appropriately updated, provides the incumbent.

At each outer iteration indexed by l , we indicate by z_L and z_U the correspondent lower bound and upper bound (incumbent), respectively. Moreover we denote by (w^U, b^U, y^U) the feasible solution corresponding to z_U and by $(w^{(l)}, b^{(l)}, y^{(l)})$ the current approximate solution to problem $P_{LR}(\lambda)$ for $\lambda = \lambda^{(l)}$.

We summarize the algorithm as follows:

Algorithm 1.

- Step 0** (Initialization) Fix $\lambda^{(0)} \geq 0$; set $l := 0$, $z_L = -\infty$ and $z_U = +\infty$.
- Step 1** (Solving Lagrangian relaxation) Compute $(w^{(l)}, b^{(l)}, y^{(l)})$ by applying Algorithm BCD to problem $P_{LR}(\lambda^{(l)})$. Set $z^{(l)} := f(w^{(l)}, b^{(l)}, y^{(l)})$. If $z^{(l)} > z_L$, set $z_L := z^{(l)}$.
- Step 2** (Checking feasibility) If $(w^{(l)}, b^{(l)}, y^{(l)})$ is feasible, set $(\bar{w}, \bar{b}, \bar{y}) := (w^{(l)}, b^{(l)}, y^{(l)})$, $\bar{z} := f(\bar{w}, \bar{b}, \bar{y})$ and go to Step 3; otherwise go to Step 4.
- Step 3** (Possible updating of z_U) If $\bar{z} < z_U$, set $(w_U, b_U, y_U) := (\bar{w}, \bar{b}, \bar{y})$ and $z_U := f(w_U, b_U, y_U)$. Go to Step 5.
- Step 4** (Feasibility recovering) From $(w^{(l)}, b^{(l)}, y^{(l)})$ recover a feasible solution $(\bar{w}, \bar{b}, \bar{y})$ and set $\bar{z} := f(\bar{w}, \bar{b}, \bar{y})$. Go to Step 3.
- Step 5** (Updating λ) Compute $\lambda^{(l+1)} := \max\{0, \lambda^{(l)} + t_l g^{(l)}\}$, with $g^{(l)} \in \partial z_{LR}(\lambda^{(l)})$, set $l := l + 1$ and go to Step 1.

At Step 4 a feasible solution $(\bar{w}, \bar{b}, \bar{y})$ of the original problem is obtained from $(w^{(l)}, b^{(l)}, y^{(l)})$ by adjusting first, at the minimum cost, variables y , thus obtaining \bar{y} . Then, the couple (\bar{w}, \bar{b}) is obtained by solving the following SVM optimization problem, where we fix $y = \bar{y}$

$$\begin{aligned} \min_{w,b} \frac{1}{2} \|w\|^2 &+ C \sum_{i=1}^k \sum_{j \in J_i^-} \max\{0, 1 + (w^T x_j + b)\} \\ &+ C \sum_{i=1}^m \sum_{j \in J_i^+} \max\{0, 1 - \bar{y}_j (w^T x_j + b)\}. \end{aligned}$$

The well-known subgradient method [285] with projection onto the non negative orthant is used at Step 5 for updating λ . In particular we take

$$g_i^{(l)} = 1 - \sum_{j \in J_i^+} \frac{y_j^{(l)} + 1}{2}, \quad i = 1, \dots, m \quad (5.2)$$

and

$$t_l = \frac{z_U - z_L}{\|g^{(l)}\|^2}.$$

Note that, because of the nonnegativity of the stepsize t_l , using formula (5.2) implies that $\lambda_i^{(l)}$ increases when the corresponding i -th constraint of the type 5.1 is not satisfied at the point $(w^{(l)}, b^{(l)}, y^{(l)})$. For further details on this approach we refer the reader to [281].

5.3.3 Numerical experiments on image classification task

Since the aim of our work is to evaluate more deeply the impact of *MIL-RL* the proposed algorithm on image classification, we tested the proposed approach on different image data sets, hoping to obtain improvements on classification performance. In this subsection we report the numerical experiments we have done referring also to the articles in which they were published.

Application on Gray Scale Images

In [176], we have applied the proposed approach in the first instance, to some toy gray images data set. In particular, we have generated twenty-two grey level images of 128×128 pixels dimension, divided into two different classes, positive and negative. According to MIL assumption, we have considered positive each image (bag) containing at least one star and negative the images without stars.

Then we have applied a segmentation process, by providing a grey scale of each image and by grouping the pixels in square subregions (blobs) of appropriate dimension.

In this way each image is represented as a bag, while a blob corresponds to an instance of the bag. For each instance, we have considered the following features: the grey scale average of the blob and the difference between the grey scale average of the blob and that of the neighbouring blobs (up, down, right, left), resulting in a five dimensional feature vector. In our experiments we have tested the code *MIL-RL* by separating, according to a MIL paradigm, each single positive image (with at least a star) from each single negative one (without stars), coming out with 121 trials, each of them characterized by a couple of images: one positive and the other negative. The code has been able to correctly classify 110 couples of images and it has failed in 11 cases.

Application on Color Images

In [177] *MIL-RL*, the algorithm presented in subsection 5.3.2, has been adopted also for preliminary color image classification problems. For our experiments we have generated one hundred colour images of 128×128 pixels dimension, divided into two different classes, positive and negative. In particular, we have considered positive each image (bag) containing the yellow colour and negative the images without yellow. The images with yellow color are reported in Figure 5.2, while the ones without yellow color are reported in Figure 5.3.

We have performed a segmentation process by means of some image processing standard Matlab routines. In particular, given a bitmap image, this image is read by the *imread*

Matlab routine, which provides a 128×128 matrix, the indexed image, each element corresponding to a pixel and containing a triplet which represents the RGB (red, green, blue) scale. In fact the first value of each triplet represents the red pixel intensity, the second one represents the green pixel intensity and the last one represents the blue pixel intensity. Once the indexed image has been generated, the successive step consists in converting each indexed image (and the corresponding color-map) into a RGB image by means of the *ind2rgb* Matlab subroutine.

Afterward we proceeded by grouping the pixels in square subregions of appropriate dimension: each image subregion forms the so called "blob". For each blob we have computed the following quantities:

- the average of the RGB intensities of the blob;
- the difference between the average of the RGB intensities of the blob and that of the upper adjacent blob;
- the difference between the average of the RGB intensities of the blob and that of the lower adjacent blob;
- the difference between the average of the RGB intensities of the blob and that of the left adjacent blob;
- the difference between the average of the RGB intensities of the blob and that of the right adjacent blob.

Taking into account that the dimension of each blob has been fixed to 32×32 , we have come out with a database constituted by one hundred bags, four hundred instances (four for each bag) and fifteen features. We have chosen $C = 10$ and we have performed two kinds of numerical experiments.

Tenfold cross-validation (a widely used protocol), which consists in splitting the data set of interest into ten equally sized pieces, has been adopted [292]. For each data set, the method has been run ten times, and each time nine pieces have been used as the training set and the remaining one as the testing set. In the first one we have generated, for ten times, a testing set by choosing ten different images (five positive and five negative), so that each time the remaining ninety (forty-five positive and forty-five negative) have constituted the training set.

In Table 5.1, for each trial, we report the training and the testing correctness percentage, respectively, with the average for both of them in the last row.

In the second type of experiment, we have performed a leave one-out cross validation, obtaining 84.07% and 83% as training and testing correctness, respectively. We have noted that, among the seventeen failures in the testing set, fifteen are positive images and two are negative images. Considering the fifteen misclassified positive images it is worth noting that, referring to Figure 5.2 apart images 1.36 and 1.50, all of them (1.12, 1.20, 1.22, 1.28, 1.34, 1.35, 1.38, 1.40, 1.42, 1.43, 1.47, 1.48, 1.49) contain a very small yellow coloured region.

Moreover, we must underline that in many of these images the parts containing the yellow were close to the edges and were not directly included within the processed blobs.

The Figure 5.4 shows the simple strategy of extracting the blobs that we have adopted. The blobs *A*, *B*, *C* and *D* (in red) with the respective upper, lower, right and left adjacent are highlighted. For each of the images in our data set, only the central four blobs have

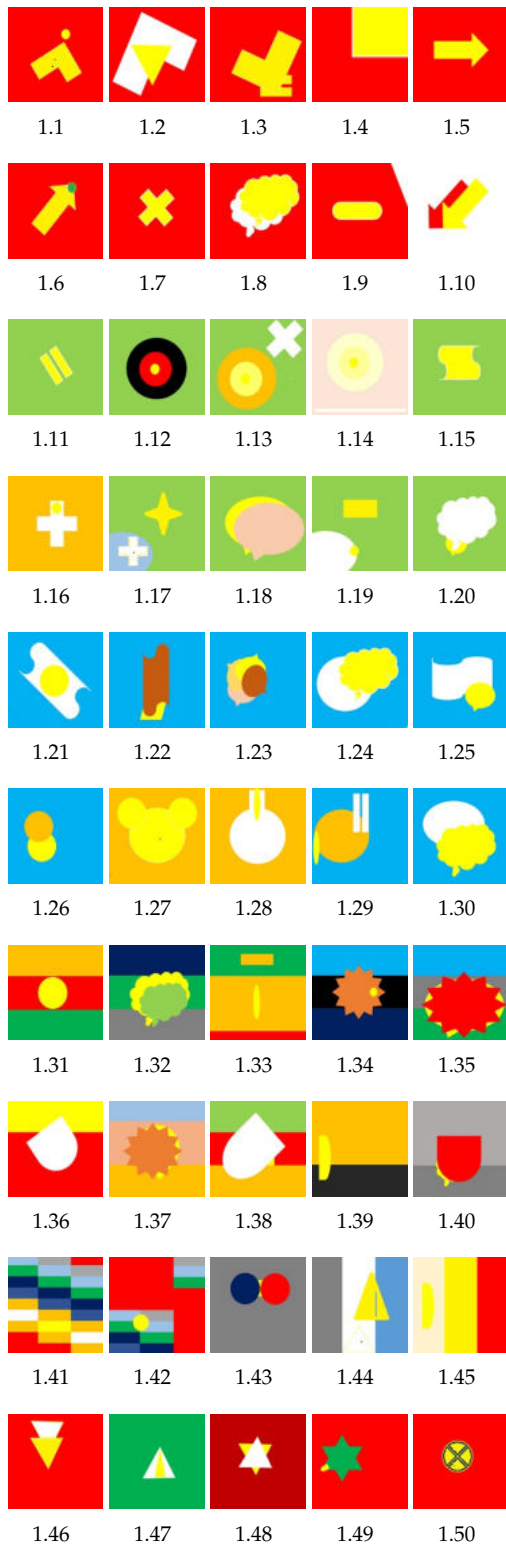


FIGURE 5.2: Images with yellow

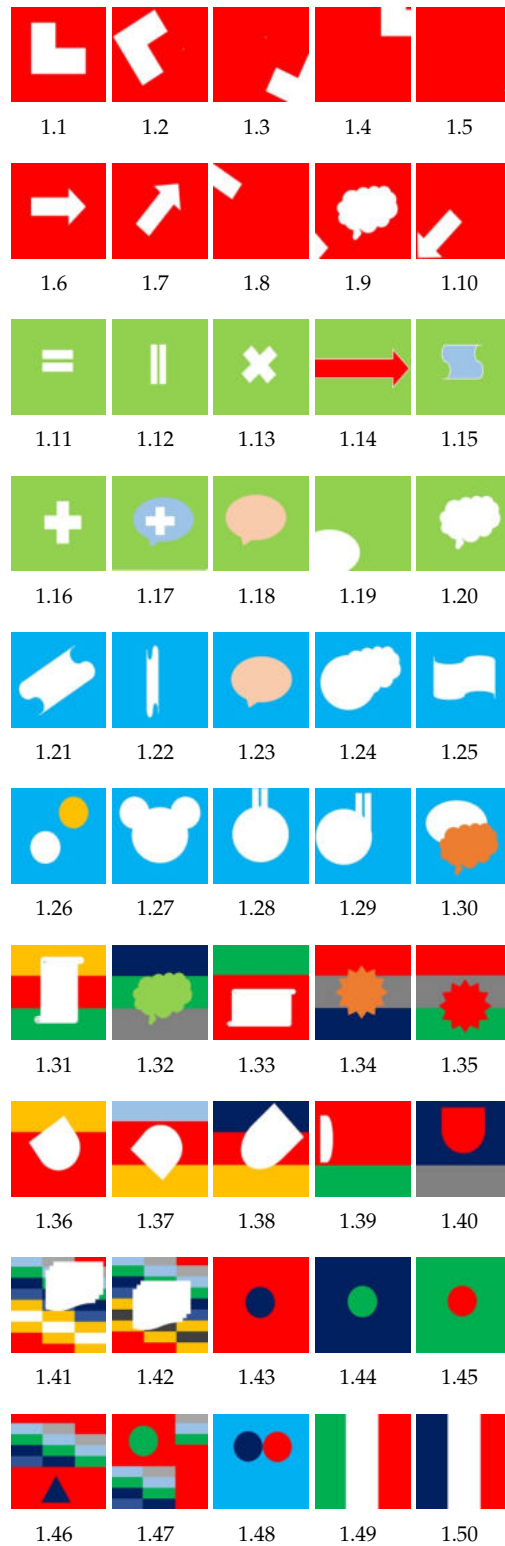


FIGURE 5.3: Images without yellow

#	Training correctness	Testing correctness
1	81.11	100
2	81.11	80
3	84.44	90
4	83.33	80
5	82.22	80
6	81.11	90
7	83.33	80
8	85.56	70
9	84.44	70
10	84.44	70
Average	83.11	81

TABLE 5.1: Results of the first experiment for color image classification



FIGURE 5.4: Sequence of extracted blobs

been extracted. As a consequence, the yellow parts outside the surface interested by blobs extraction have not directly evaluated (see Figure 5.5).

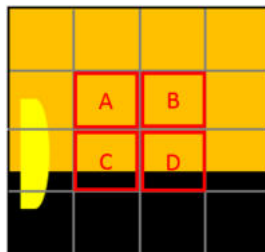


FIGURE 5.5: Extracted blobs from subfigure 1.39 of Figure 5.2

This emphasizes the value of the obtained results, given that they could be improved both by defining more sophisticated extraction strategies for the blobs, and by considering a greater number of adjacent blobs for each one selected.

Application on Dermatoscopic Images

As we have seen in Sections 4.3.1, there are many techniques for skin image acquisition. In particular, the so called *epiluminescence (ELM)* consists in photographing the pigmented skin lesions and in storing the acquired images, in order to compare them with new images of the same lesions taken at a later time (follow-up) [293]. This comparison is not an easy task but it could be performed by means of machine learning techniques able to classify the images on the basis of recurring patterns.

In [280], MIL-RL algorithm has been tested for classification of some medical dermoscopic images drawn from the PH^2 database [3].

The PH^2 database was set up by the Universidad do Porto and Tecnico Lisboa, in collaboration with the Dermatology Service of the Hospital Pedro Hispano (Portugal). The equipment used to acquire these dermoscopic images is the Tubinger Mole Analyzer system, by which it is possible to obtain high resolution images with a magnification factor of $20\times$. The entire PH^2 database contains 200 melanocytic lesions images: 80 common nevi, 80 atypical nevi and 40 melanomas. All of them have been obtained in 8-bit RGB color with a resolution of 768×560 pixels.

Patients from whom the photos have been taken correspond to phototype II or III, according to Fitzpatrick skin type classification scale [294]: for this reason the background color, not affected by injury, varies from white to creamy white. These images have been classified as common nevi, atypical nevi or melanomas by expert dermatologists on the basis of the following parameters:

- manual segmentation of the skin lesion;
- clinical and histological diagnosis;
- dermoscopic criteria (asymmetry, colors, pigment network, particular structures).

In the experiments presented in [280], we have considered 40 images of melanomas (5.6-a) and 80 images of common nevi (5.6-b). None of the above listed parameters, resulting from manual analysis, has been used as features in our automatic classification process. The only criterion adopted is at the image level, considering positive the images related to melanomas and negative those ones related to common nevi.

Moreover the images we have used have not been pre-processed, i.e., they have not been cleaned up from the presence of noises, such as possible hairs or the halo left by the dermoscopic gel used to allow better lighting of the nevus.

In applying the segmentation process, to avoid a large number of instances, using image scaling techniques (see 4.3.2), we have first cut the outer edges of the original image obtaining an image of 512×512 pixels and proceeding subsequently to a reduction of the image resolution to 128×128 pixels.

The segmentation has been performed as in [177] by means of the standard image processing Matlab routines *imread* and *ind2rgb*. We have proceeded by grouping the pixels in square subregions (blobs) of dimension 32×32 pixels.

For each blob we have computed the following quantities: the average and the variance of the RGB (red, green, blue) intensities of the blob and the differences between these values and the same corresponding quantities computed for the adjacent blobs (up, down, left, right). Since we have not considered the blobs along the frame of each image, we have come out with a data set where each image is a bag characterized by four instances (the blobs) and 30 features for each instance.

The MIL algorithm has been implemented in Matlab and the corresponding code, named MIL-RL, has been run on a Windows 10 system characterized by a 2.21 GHz processor. For our experiments we have considered the following three configurations of the data set (see Figure 5.6):

1. 80 images: all the 40 melanomas images and the first 40 common nevi images;
2. 120 images: the entire data set (all the 40 melanomas images and all the 80 common nevi images);

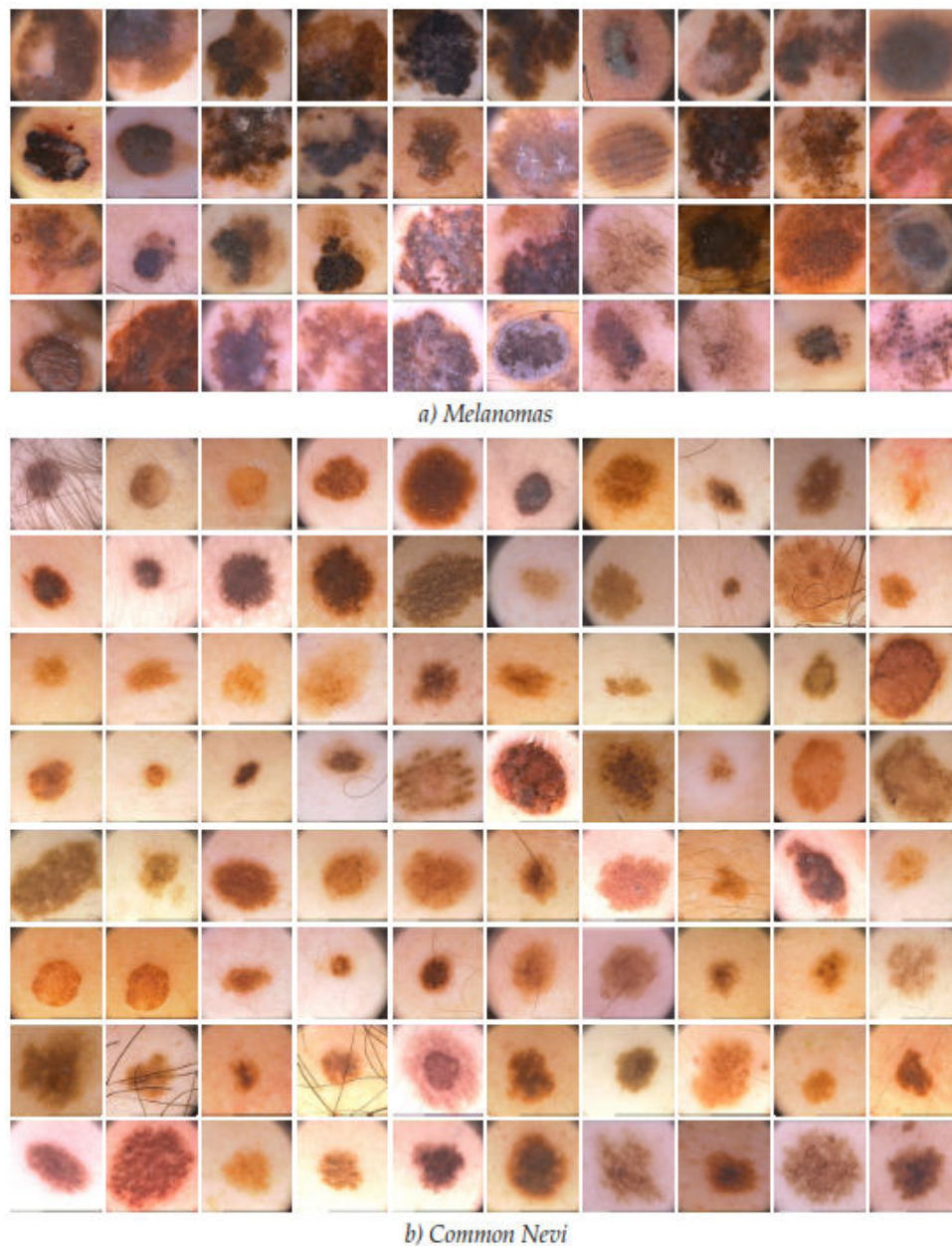


FIGURE 5.6: Melanomas and common nevi images

3. 160 images: to have a balanced data set, we have duplicated all the melanomas images, adding to the repeated ones a zero-mean Gaussian noise with variance equal to 0.0001, as in [295].

In this way we have obtained a data set configuration with 80 images of melanomas and 80 of common nevi. For each data set configuration, we have performed three types of experiments, using a five-fold and ten-fold cross validation (CV) in the first two cases, and a leave-one-out validation in the last case [296]. To compute the optimal value of the weighting parameter C we have adopted a be-level approach as in [297], [298].

The respective results are listed in Tables 5.2, 5.3, and 5.4, where we report the average of the following standard quantities: testing correctness, sensitivity, specificity, F-score and CPU time [299], [300]. Please refer to Appendix A for further details regarding the definition

of these performance measures.

Our MIL optimization model P is of the SVM type: for this reason, to compare the MIL classification paradigm with a classical one, we report in the columns named "SVM" and "SVM-RBF" the results obtained using a standard SVM approach [34] with linear and RBF kernels, respectively.

For this purpose we have used the SVM subroutines provided the Statistics and Machine Learning toolbox of the Matlab package, considering each image as a point characterized by six features: the average and the variance of the RGB intensities of the entire image. For each data set configuration and for each experiment (5-CV, 10-CV and leave-one-out), the best results in Tables 5.2, 5.3, and 5.4 have been underlined.

	5-CV			10-CV			Leave-One-Out		
	MIL-RL	SVM	SVM-RBF	MIL-RL	SVM	SVM-RBF	MIL-RL	SVM	SVM-RBF
Correctness (%)	<u>86.25</u>	<u>86.25</u>	76.25	<u>90.00</u>	87.50	80.00	<u>91.25</u>	87.50	81.25
Sensitivity (%)	<u>94.92</u>	82.26	79.68	<u>95.50</u>	90.17	82.67	<u>97.50</u>	90.00	85.00
Specificity (%)	77.54	<u>90.20</u>	72.46	84.17	<u>89.17</u>	80.00	<u>85.00</u>	<u>85.00</u>	77.50
F-score (%)	<u>87.57</u>	84.55	76.72	<u>89.40</u>	<u>87.06</u>	78.26	<u>91.76</u>	87.80	81.93
CPU time (secs)	0.19	1.21	<u>0.05</u>	0.41	1.02	<u>0.01</u>	0.41	1.15	<u>0.01</u>

TABLE 5.2: Data set constituted by 40 melanomas and 40 common nevi: average testing values

	5-CV			10-CV			Leave-One-Out		
	MIL-RL	SVM	SVM-RBF	MIL-RL	SVM	SVM-RBF	MIL-RL	SVM	SVM-RBF
Correctness (%)	<u>90.00</u>	<u>90.00</u>	86.67	90.00	<u>92.50</u>	87.50	89.17	<u>90.00</u>	88.33
Sensitivity (%)	<u>84.48</u>	82.70	75.40	<u>92.14</u>	87.74	75.24	<u>90.00</u>	80.00	75.00
Specificity (%)	91.89	<u>94.86</u>	75.81	89.10	<u>96.42</u>	95.40	88.75	<u>95.00</u>	95.00
F-score (%)	82.97	<u>84.37</u>	77.28	85.53	<u>86.98</u>	76.50	<u>84.71</u>	<u>84.21</u>	81.08
CPU time (secs)	0.82	1.56	<u>0.01</u>	1.03	1.87	<u>0.01</u>	2.04	1.67	<u>0.01</u>

TABLE 5.3: Data set constituted by 40 melanomas and 80 common nevi: average testing values

	5-CV			10-CV			Leave-One-Out		
	MIL-RL	SVM	SVM-RBF	MIL-RL	SVM	SVM-RBF	MIL-RL	SVM	SVM-RBF
Correctness (%)	<u>93.13</u>	88.13	89.38	<u>90.63</u>	85.63	89.38	<u>92.50</u>	90.00	91.88
Sensitivity (%)	<u>97.89</u>	84.52	90.69	<u>95.24</u>	79.03	90.36	<u>97.50</u>	85.00	92.50
Specificity (%)	88.66	<u>91.74</u>	88.90	84.53	<u>90.50</u>	87.85	87.50	<u>95.00</u>	91.25
F-score (%)	<u>93.65</u>	87.03	89.63	<u>91.73</u>	83.37	89.65	<u>92.86</u>	89.47	91.93
CPU time (secs)	1.52	1.55	<u>0.01</u>	1.90	1.81	<u>0.01</u>	2.68	2.08	<u>0.01</u>

TABLE 5.4: Data set constituted by 80 melanomas and 80 common nevi: average testing values

We observe that, in general, our approach outperforms the SVM technique (with both linear and RBF kernels) in terms of accuracy and specificity.

It is worth noting that whenever the accuracy is not equal to 100%, low values of specificity are generally a consequence of high values of sensitivity. Moreover, the sensitivity plays a more important role than the specificity since it is a measure of the capability to identify non-healthy patients. This is taken into account by the F-score parameter, whose values show the good performance of the MIL approach in melanoma classification with respect to the classical SVM technique. Although the SVM approach with RBF kernel is very fast, its F-score values are worse than those provided by the linear SVM, except for the third

configuration of the data set. On the other hand, we observe that the CPU times of our algorithm are comparable with those provided by the linear SVM.

In [283] we have presented the results obtained first by applying MIL-RL on a subset of photos extracted from PH^2 , and then on the same subset once the images affected by the presence of hair were pre-processed.

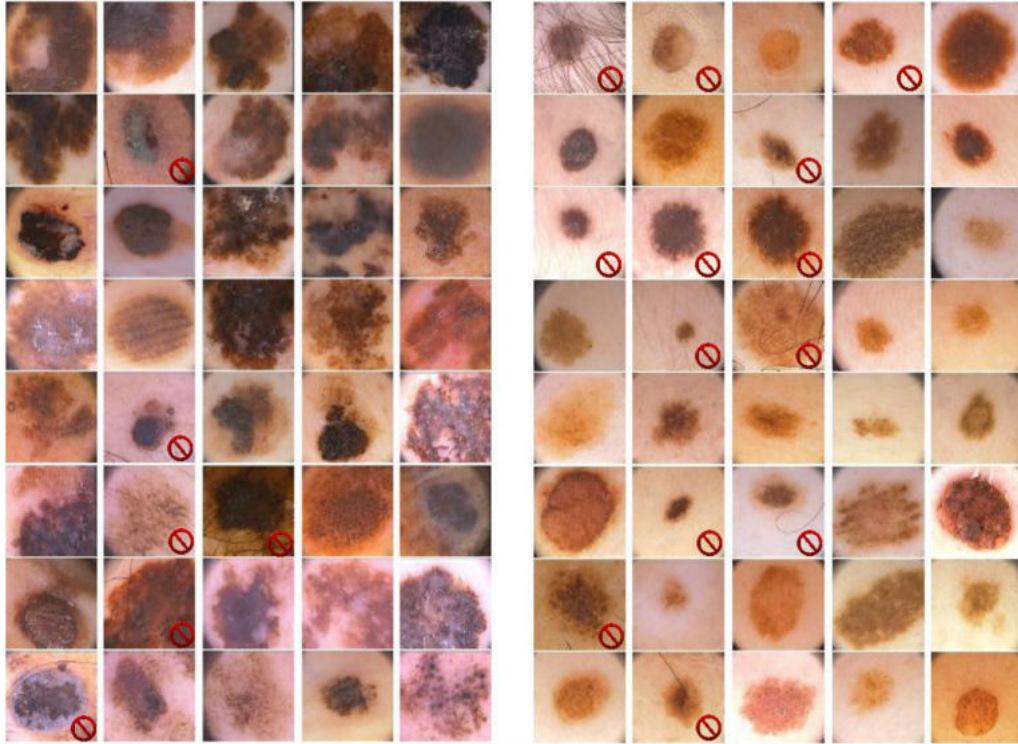


FIGURE 5.7: Selected images of melanomas and common nevi for pre-processing step

For the pre-processing step we use DullRazor [205], a dedicated software tool that removes hair from images by identifying the positions of dark hair, verifying that the dark pixel is a thin structure and finally smoothing the pixels replaced with an adaptive median filter (see 4.3.2). Although the pre-processing step was carried out only for hair removal and through a not recent tool, the classification performances were slightly better than those obtained on the non preprocessed data set.

In Figure 5.7 we report the details of the images we used. The red symbol highlights the photos that have been pre-processed and replaced in the original data set. Tables 5.5 and 5.6, report the results obtained through 10-cross-fold and 5-fold-cross validation respectively, on the data set containing pre-processed images. The last two lines of each tables highlight the small differences recorded considering the pre-processed data set (Average with PP) compared to the non-preprocessed one (Average No-PP).

Application on Plain Photos

In [301] we have presented a preliminary analysis using color and texture features on a data set constituted by plain photos, to which no pre-processing technique has been applied. This is motivated by the necessity to open new horizons in creating self-diagnosis systems for accessible skin lesions due also to a huge innovation of cameras, smartphones technology and wearable devices.

Fold number	Training correctness	Testing correctness	CPU Time
1	94.44	100.00	0.47
2	94.44	100.00	0.38
3	97.22	87.50	0.46
4	94.44	87.50	0.35
5	91.67	87.50	0.18
6	95.83	62.50	0.23
7	93.06	100.00	0.33
8	93.06	75.00	0.26
9	94.44	62.50	0.23
10	90.28	87.50	0.39
Average with PP	93.89	85.00	0.32
Average No-PP	92.64	83.75	0.37

TABLE 5.5: Test on pre-processed Melanoma DB: 10-fold cross-validation

Fold number	Training correctness	Testing correctness	CPU Time
1	95.31	87.50	0.52
2	92.19	100.00	0.32
3	95.31	75.00	0.40
4	92.19	93.75	0.32
5	93.75	87.50	0.31
Average with PP	93.75	88.75	0.38
Average No-PP	93.13	88.75	0.29

TABLE 5.6: Test on pre-processed Melanoma DB: 5-fold cross-validation

In particular, this study was inspired by the good results obtained in [280], where we applied Multiple Instance Learning techniques for classifying some medical dermoscopic images drawn from the PH^2 database [3], by using only color features. Our numerical experimentation showed that using only color features is not at all satisfactory when the classification process is performed on a data base constituted by common plain photographs. On the other hand using in addition texture features, provides reasonable classification results, especially if we consider that no pre-processing techniques were used. A numerical study is performed for classifying a set of 200 plain photographs (100 melanomas and 100 common nevi), using color and color/texture features.

This study starts from some considerations related to the works [4], [295]. In [295] C. Barata et al. analyzed the role played by the color and the texture features, showing empirically that using only the color features outperforms the use of the texture features: very good results were obtained by means of different type of classifiers on an image data set drawn from the PH^2 database [3], containing 200 melanocytic lesions images (80 common nevi, 80 atypical nevi and 40 melanomas). All the images of the database were selected on the basis of their quality, resolution and dermoscopic features. As we have mentioned before, good results were obtained also in [280] on the same data set PH^2 , by applying Multiple Instance Learning techniques and using only color features. On the other hand, the objective in [4] was to select the most important image features usable in melanoma detection.

Differently from [280] where a high quality data set was adopted, Mustafa and Kimura, performed their experimentation on a data set constituted by plain photographs publicly available from two online databases <https://www.dermquest.com> and <http://www.dermins.com>.

`net1`, obtaining very good results in terms of sensitivity (96.77%) and, consequently, in terms of F1-score (90.91%). The key point of this work consisted in an accurate pre-processing and segmentation steps.

On the contrary, the objective of our experimentation consists in simply evaluating the classification results, obtainable on 200 images (100 melanomas and 100 common nevi) drawn from the same database used in [4], but without performing any pre-processing step aimed at cleaning up the images from possible noises. This choice is motivated by the necessity to investigate the possibility to create fast self-diagnosis systems for accessible skin lesions. In particular, our experiments were performed by using three configurations of the data set, corresponding respectively to the 5-fold cross validation, the 10-fold cross-validation and the leave one-out validation. For each configuration, initially only the color (RGB) features were used, adding, only successively, those ones related to the texture by means of the co-occurrence matrix. The following classification methods, each of them implemented in Matlab, were adopted:

- Support Vector Machine with linear kernel (SVM) [34];
- Support Vector Machine with RBF kernel (SVM-RBF) [34];
- MIL-RL [96]

Regarding the MIL-LR code, the same Matlab implementation adopted in [280] were used, while, for the SVM approaches with both linear and RBF kernels, the *fitcsvm* and *predict* subroutines for training the classifier and for computing the testing correctness, respectively, were utilized. Such subroutines are included in the Matlab optimization package. The results of our experimentation are reported in Tables 5.7, 5.8 and 5.9, in terms of accuracy (testing correctness), sensitivity, specificity, F-score and CPU time. For each of these parameters and for each table, the best result was underlined.

It is worth noting that the sensitivity is in general more important than the specificity because it measures the capability to identify sick patients. This is in fact taken into account in the F-score value [299], [300]. Looking at the tables, it is possible to observe that using both color and texture features improve the results, in contrast with using only color features. Moreover better results are generally obtained with the MIL-RL algorithm, whose F-score values are however comparable with those obtained by using the SVM with RBF kernel. On the other hand using the SVM-RBF is really faster than using the SVM with linear kernel or the MIL-RL approach.

	Color			Color and texture		
	MIL-RL	SVM	SVM-RBF	MIL-RL	SVM	SVM-RBF
Correctness (%)	<u>71.00</u>	52.50	57.50	<u>74.00</u>	60.50	69.50
Sensitivity (%)	59.21	39.28	<u>60.37</u>	70.59	63.52	<u>71.07</u>
Specificity (%)	<u>83.46</u>	67.29	56.09	<u>77.21</u>	57.81	68.67
F-score (%)	<u>65.99</u>	45.74	58.53	<u>72.10</u>	61.11	69.54
CPU time (secs)	2.52	3.00	<u>0.08</u>	2.98	2.67	<u>0.42</u>

TABLE 5.7: 5-fold cross-validation

In [301] some numerical experiments were performed to classify a data set constituted by plain photos of different sizes, some of which are low resolution images and others are blurred or with hairs. Although in [4] it has been shown that using some pre-processing

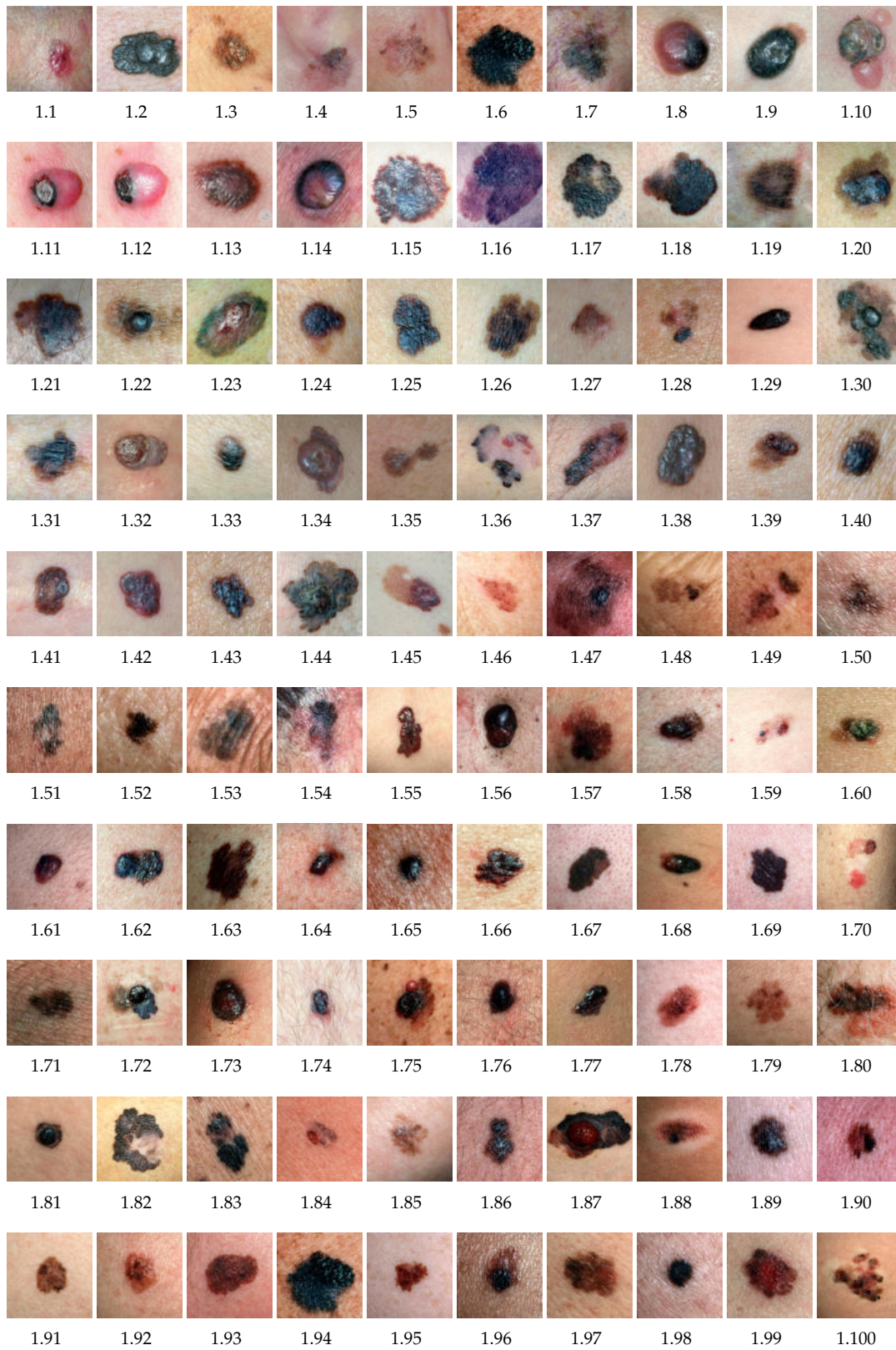


FIGURE 5.8: Plain photos of melanoma



FIGURE 5.9: Plain photos of common nevi

	Color			Color and texture		
	MIL-RL	SVM	SVM-RBF	MIL-RL	SVM	SVM-RBF
Correctness (%)	<u>71.00</u>	47.50	56.50	<u>72.00</u>	54.50	68.00
Sensitivity (%)	59.63	37.19	<u>61.55</u>	67.30	59.91	<u>70.98</u>
Specificity (%)	<u>83.97</u>	61.75	54.23	<u>78.16</u>	49.62	65.30
F-score (%)	<u>65.14</u>	41.83	57.93	<u>69.54</u>	56.27	68.22
CPU time (secs)	3.26	2.93	<u>0.02</u>	3.82	3.31	<u>0.01</u>

TABLE 5.8: 10-fold cross-validation

	Color			Color and texture		
	MIL-RL	SVM	SVM-RBF	MIL-RL	SVM	SVM-RBF
Correctness (%)	<u>71.50</u>	48.50	57.50	<u>69.00</u>	55.00	67.50
Sensitivity (%)	57.00	33.00	<u>64.00</u>	63.00	59.00	<u>69.00</u>
Specificity (%)	<u>86.00</u>	64.00	51.00	<u>75.00</u>	51.00	66.00
F-score (%)	<u>66.67</u>	39.05	60.09	67.02	56.73	<u>67.98</u>
CPU time (secs)	4.44	4.19	<u>0.02</u>	5.50	3.49	<u>0.01</u>

TABLE 5.9: Leave-One-Out validation

methodologies on the images improves the classification performances, in our experiments we preferred not using such techniques. We intended to operate in the worst conditions in order to compare the classification performances obtainable by using, on one hand, only color features (as proposed in [176], [177], [281]) and, on the other hand, color and texture features. The fact that we used a data set of plain photographs, instead of high-quality dermatoscopic images such as those ones constituting the PH^2 database, justifies the obtained classification performance.

5.3.4 Classification tasks involving dysplastic nevi

Malignant melanoma is responsible for the highest number of deaths related to skin lesions. However, early diagnosis may allow positive treatment of this terrible form of cancer. The similarities of melanoma with other skin lesions such as *dysplastic nevi*, however, constitute a pitfall for early diagnosis. The research community is committed to proposing software solutions that favor the computerized analysis of lesions for melanoma detection. The proposed algorithms and methods have had as main focus the dichotomous distinction of melanoma from benign lesions and they rarely focused on the case of melanomas against dysplastic nevi. This challenge is much more difficult due to the similarity of the injuries.

Currently, there is a debate about *dysplastic nevi syndrome*, also referred to as *atypical mole syndrome*, concerning the number of moles present on the human body as potential melanoma risk factor.

In this section, we consider the challenging task of applying a multi-instance learning algorithm for the differentiation of melanomas from dysplastic nevi and outline an even more complex challenge related to the classification of dysplastic nevi as opposed to common nevi. Since the results appear promising, we conclude that a MIL technique could become the basis of more sophisticated tools useful to skin lesion detection.

Some studies have shown that specific ethnic groups have a greater number of common and dysplastic nevi present on the surface of their bodies. For example, 8% of the Caucasian population has been reported to have dysplastic nevi or unusual lesions that may resemble melanoma [302]. Individuals with *dysplastic nevi syndrome* or dysplastic nevi with family history of melanoma face a greater risk of developing melanoma, and generally People with

10 or more atypical moles have $12\times$ the risk of melanoma [303]. However, only a small number of dysplastic nevi could really degenerate into a melanoma [304].

These premises justify the consideration that the automatic diagnosis of skin lesions must consider, besides the distinction between melanomas and common nevi, also that between melanomas and dysplastic nevi. In particular, the discrimination of melanomas from dysplastic nevi is more difficult due to the similarities of the two type of lesions [305]. In Figure 5.11, macroscopic images of two melanocytic lesions are reported. The characteristics are superimposed by the ABCD rule (asymmetry, irregular borders, varied coloration, diameter greater than 6 mm).

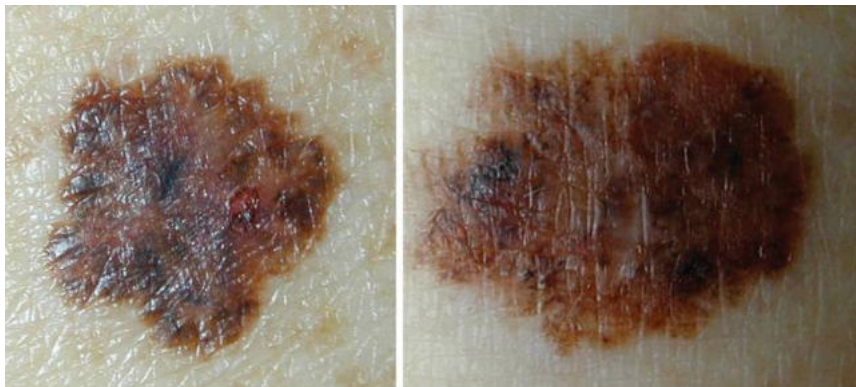


FIGURE 5.10: Left - dysplastic nevus; Right – cutaneous melanoma.

In the next subsection we focus on the role of dysplastic nevi and common nevi in terms of risk of melanoma onset.

Dysplastic nevi The term “dysplastic nevus” (DN) indicates that this nevus exists and presents different histological and genetic characteristics compared to common nevus. The term “dysplastic nevus” (DN) derives from the Greek *dis-* (bad or malfunction) and *-plasia* (development of growth or change) [306] and indicates a potentially dangerous lesion for his host. The scientific community has not yet specifically clarified how a dysplastic nevus can be a risk factor. Several studies have attempted to correlate the degree of dysplasia of melanocytes with the risk of melanoma [307], [308].

Syndrome of dysplastic nevus (DNS) refers to subjects who have a high number of benign moles and also have dysplastic nevi. A small percentage of these individuals are predisposed to melanoma formation [308]. The inherited syndrome of dysplastic nevus is an autosomal dominant condition. Dysplastic nevi are more likely to undergo malignant transformation when they occur among members of melanoma families.

In [309], the authors indicate a cumulative lifetime risk of almost 100% in individuals who have dysplastic nevi and are related to melanoma; about 30% of melanomas occur within atypical moles. In 40-50% of cases, there is a genetic predisposition for the formation of melanoma. The onset of this skin cancer has been associated with germline mutations in the CDKN2A gene, which encodes p16 (a regulator of cell division).

In [310], some studies based on histological analysis have correlated the presence of dysplastic nevi with melanomas. Caution should be observed in correspondence with a diagnosis of a severe DNS, as it could represent a miss-diagnosed in situ melanoma [311]. Therefore, a DN classified as severely dysplastic may reflect the dermatopathological uncertainty

related to a wrong diagnosis. Currently there is a debate in the scientific community, to define more precisely the correlation between the presence of dysplastic and common nevi and the possibility of melanoma occurrence [302]. There are two objective criteria that have been shown to be related to the risk of melanoma:

- A high number of nevi is related to an increased risk of melanoma [312]. In [313], people with a number of nevi greater than 100 had a 7 times greater risk of melanoma than those with a count of less than 15.
- The presence of large nevi increases the relative risk of melanoma. A histological study of nevi has shown that the greater their extension the higher the risk of dysplastic nevi turning into melanoma. If these nevi have a diameter less than 2.4 mm they have a relative risk of 1, while the relative risk progressively increases up to 5 if the lesion has a diameter greater than 4.4 mm [314].

Simultaneously with the definition of the exact cause-effect correlations, various solutions have been proposed over time for the automatic identification of skin lesions.

The MIL algorithm we use for the classification task in this chapter has been proposed in [96], and has been tested for the classification of both dermoscopic images [280], [281] taken from the PH^2 database [3], and for the classification of photographs data sets publicly available from two online databases <https://www.dermquest.com> and <http://www.dermins.net> [301].

For the classification experiments we considered the 40 images of melanomas the 80 of dysplastic nevi and the 80 of common nevi (see Figure 5.11), without taking into account the indications resulting from the manual analysis carried out by the specialists. The only criterion adopted is at the image level, considering:

- A) Melanomas (positive images) vs Dysplastic nevi (negative images);
- B) Dysplastic nevi (positive images) vs Common nevi (negative images).
- C) Melanomas (positive images) vs Dysplastic nevi and Common nevi (negative images)

Although the pre-processing phases of the images allow for better performance (see [283], [315]), the images we used were not pre-processed, i.e. they were not cleaned of the presence of noise, such as possible hair or halo left by the dermoscopic gel used to allow better illumination of the lesions. To avoid the problems related to the use of data sets with unbalanced classes, we have duplicated all the images of melanomas, adding to the ones repeated a Gaussian noise with zero mean with variance equal to 0.0001, as in the [295]. In this way we obtained a balanced data set containing three classes of data, Melanomas (M), Dysplastic Nevi (DN) and Common Nevi (N) each with 80 images. For our experiments we have considered the following two data set configurations:

- 160 images: 80 Melanomas vs 80 Dysplastic Nevi;
- 160 images: 80 Dysplastic Nevi vs 80 Common Nevi;
- 240 images: 80 Melanomas vs (80 Dysplastic Nevi + 80 Common Nevi).

For each data set configuration, we performed two types of experiments using a five fold and a ten fold cross-validation. The respective results are listed in Tables 5.10 and 5.11, where we report the average of the following standard quantities: Correctness, Sensitivity, Specificity, F score and CPU time.

The proposed MIL optimization model P is of the SVM type; in order to appreciate the MIL classification paradigm, we report in the columns "SVM" and "SVM-RBF" the results obtained using a standard SVM approach [34] with linear and RBF kernels, respectively. For each data set configuration and for each experiment (5-CV and 10-CV), the best results in Tables 5.10 and 5.11 have been underlined.

	5-CV			10-CV		
	MIL-RL	SVM	SVM-RBF	MIL-RL	SVM	SVM-RBF
Correctness (%)	<u>87.50</u>	72.50	85.63	<u>86.25</u>	69.38	<u>86.25</u>
Sensitivity (%)	<u>92.56</u>	77.21	87.06	<u>91.08</u>	69.65	87.88
Specificity (%)	81.50	67.51	<u>85.51</u>	<u>82.12</u>	69.87	<u>85.95</u>
F-score (%)	<u>88.31</u>	74.96	85.84	87.01	68.68	<u>87.52</u>
CPU time (secs)	0.90	1.84	<u>0.04</u>	1.20	2.05	<u>0.03</u>

TABLE 5.10: Data set constituted by 80 melanomas and 80 dysplastic nevi: average testing values

	5-CV			10-CV		
	MIL-RL	SVM	SVM-RBF	MIL-RL	SVM	SVM-RBF
Correctness (%)	<u>60.63</u>	60.00	49.38	<u>59.38</u>	58.13	51.88
Sensitivity (%)	35.07	<u>54.91</u>	53.58	31.77	43.67	<u>58.92</u>
Specificity (%)	<u>84.86</u>	67.58	46.97	<u>87.06</u>	73.48	46.47
F-score (%)	44.76	<u>56.79</u>	50.09	42.77	48.57	<u>53.74</u>
CPU time (secs)	1.38	1.95	<u>0.01</u>	1.71	2.13	<u>0.03</u>

TABLE 5.11: Data set constituted by 80 dysplastic nevi and 80 common nevi: average testing values

	5-CV			10-CV		
	MIL-RL	SVM	SVM-RBF	MIL-RL	SVM	SVM-RBF
Correctness (%)	<u>90.42</u>	73.75	88.33	<u>90.83</u>	78.33	87.50
Sensitivity (%)	<u>90.92</u>	63.48	75.74	<u>90.43</u>	66.86	78.70
Specificity (%)	90.61	80.01	<u>94.89</u>	92.22	85.33	<u>92.42</u>
F-score (%)	<u>86.50</u>	61.05	81.18	<u>87.13</u>	66.42	81.16
CPU time (secs)	2.94	2.81	<u>0.01</u>	4.06	3.17	<u>0.03</u>

TABLE 5.12: Data set constituted by 80 melanomas against 80 dysplastic nevi and 80 common nevi: average testing values

Melanomas vs Dysplastic Nevi

From numerical experiments it emerges that, in general, MIL-RL overcomes the SVM technique (with both linear and RBF kernels) in terms of correctness and sensitivity.

In medical fields, sensitivity plays a more important role than specificity since it is a measure of the ability to identify sick patients. Looking at the results obtained with the 5-fold-cross validation, F-score values show the good performance of the proposed MIL approach

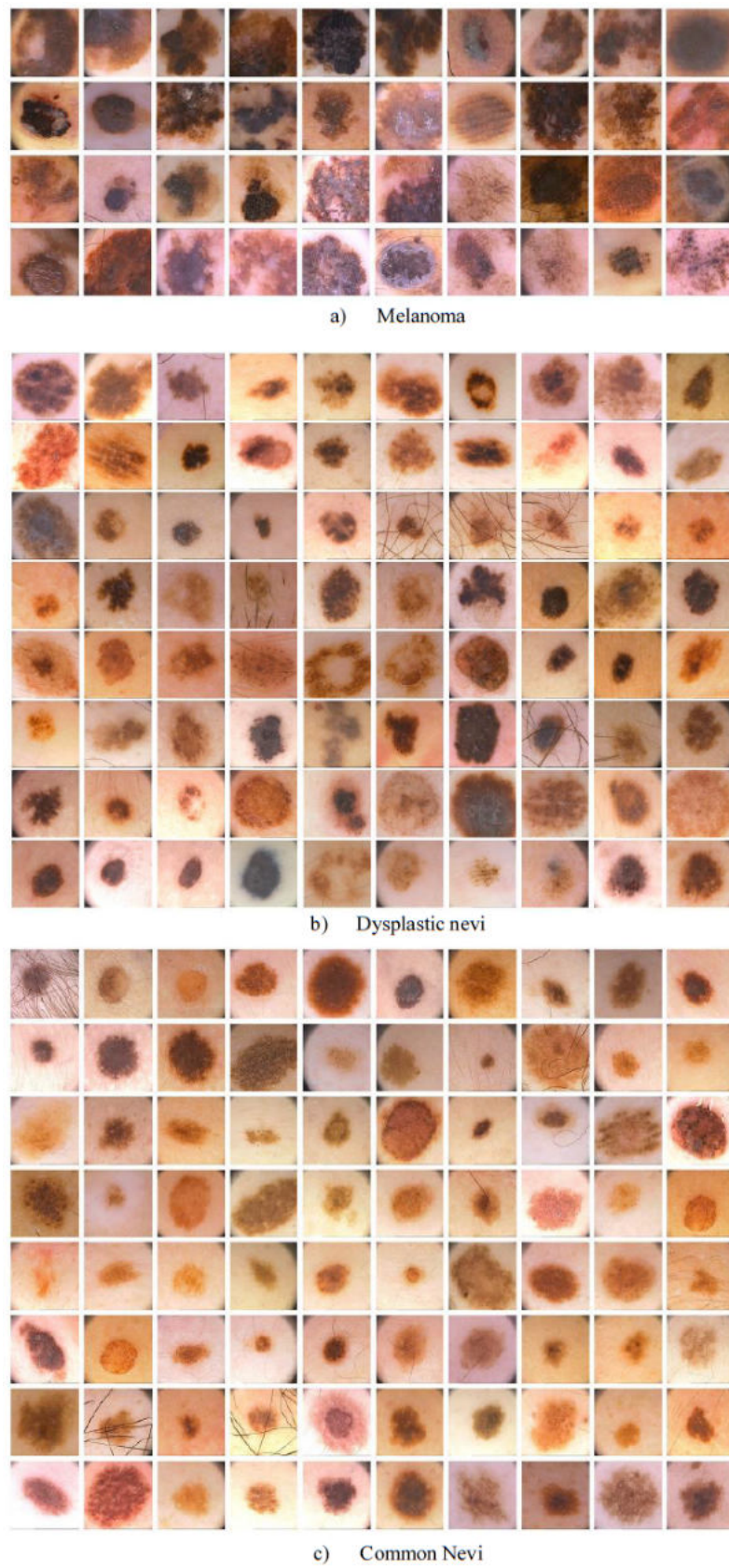


FIGURE 5.11: Dermoscopic images of Melanomas (a), Dysplastic nevi (b) and Common nevi (c)

in classifying melanomas from dysplastic nevi against the classic SVM technique. Although the SVM approach with the RBF kernel is very fast, its sensitivity values are slightly worse than those provided by MIL-RL. On the other hand, we observe that the CPU times of our algorithm are better than those recorded by linear SVM.

Dysplastic Nevi vs Common Nevi

With regard to the experimental section on the classification of dysplastic nevi against common nevi, the performances of all three methods appear unsatisfactory. The MIL-RL algorithm records the best values of accuracy and specificity, but in general it is not effective in solving the proposed task.

Melanomas vs (Dysplastic Nevi and Common Nevi)

With regard to the experimental section on the classification of melanomas against dysplastic and common nevi, MIL-RL overcomes the SVM technique (with both linear and RBF kernels) in terms of correctness, sensitivity and F-score. Although the SVM approach with the RBF kernel is very fast, its correctness, sensitivity and F-score values are worse than those provided by MIL-RL. On the other hand, we observe that the CPU times of our algorithm are comparable to the ones of linear SVM.

We have presented an application of a multiple instance learning approach for the detection of melanomas against dysplastic nevi, of dysplastic nevi against common ones and of melanomas against dysplastic and common nevi. These three issues have received little attention in the literature despite pathologies such as the dysplastic nevi syndrome can imply numerous death.

The implementation of frameworks to support the diagnosis of specialists such as Computer Aided Diagnosis Systems, and of mobile applications useful for promoting self diagnosis could be crucial to increase life expectancy.

The results obtained show that in the first and in the third case the MIL approach is very promising, even in the conditions in which we performed the experiments, i.e. with only color features and without using pre-processing steps.

In the second case, the MIL approach as well as the SVM in the linear and kernel RBF version, do not give satisfactory results. The excessive similarity of the lesions is not properly discriminated with approaches aimed at identifying linear separation surfaces [316].

One way we decided to take includes the application of MIL approaches that use spherical separation surfaces.

In particular, the algorithm [145] seems to be an interesting proposal for applications in contexts where positive and negative elements have similar characteristics. In fact, the MIL paradigm adapts very well to the classification of images in these contexts because it is able to detect global information (bags) working locally (instance level).

We observe that some images are characterized by a lot of hairs. As demonstrated in [283], [315], we underline that better results could be obtained using images pre-processing aimed at eliminating the presence of possible noises.

Even the adoption of further useful features extracted from blob is a possibility that would allow to improve the classification performances [251], [252].

5.4 DC-SMIL for automated Melanoma Detection

In this section we briefly discuss the results obtained applying DC-SMIL to the same classification tasks faced by using the MIL-RL code. In particular, we have repeated all experiments presented in the Section 5.3.3 on dermatoscopic and non-dermatoscopic images, using DC-SMIL instead of MIL-RL. The intent was to verify whether a spherical classification approach could provide better results than those obtained with MIL-RL.

We have noted that in discriminating melanomas from common nevi, melanomas from dysplastic nevi and melanomas from dysplastic nevi and common ones, MIL-RL overcomes DC-SMIL in terms of correctness and F-score. The results obtained by DC-SMIL are comparable to the classical SVM approach, using both the linear and the RBF kernels: for this reason we do not report the corresponding results.

On the other hand, some interesting considerations can be made regarding the classification between dysplastic and common nevi, which is a complex task due to the extreme similarity of the lesions to be discriminated. The corresponding results, obtained by using the 5-fold and 10-fold cross validation, are reported in Tables 5.13 and 5.14, respectively.

	5-CV			
	DC-SMIL	MIL-RL	SVM	SVM-RBF
Correctness (%)	50.50	60.63	60.00	49.38
Sensitivity (%)	47.83	35.07	54.91	53.58
Specificity (%)	60.36	84.86	67.58	46.97
F-score (%)	53.60	44.76	56.79	50.09
CPU time (secs)	0.58	1.38	1.95	0.01

TABLE 5.13: Data set constituted by 80 dysplastic nevi and 80 common nevi: 5-CV average testing values

	10-CV			
	DC-SMIL	MIL-RL	SVM	SVM-RBF
Correctness (%)	59.38	59.38	58.13	51.88
Sensitivity (%)	59.63	31.77	43.67	58.92
Specificity (%)	59.88	87.06	73.48	46.47
F-score (%)	59.81	42.87	48.75	53.74
CPU time (secs)	0.58	1.71	2.13	0.03

TABLE 5.14: Data set constituted by 80 dysplastic nevi and 80 common nevi: 10-CV average testing values

We can note that the sensitivity, the F-score and the CPU time provided by DC-SMIL are better than those obtained by MIL-RL. As we have discussed in Chapter 3, DC-SMIL offers performances of interest as regards classification tasks in which the classes to be discriminated contain similar elements. Only a few color features have been taken into consideration and no pre-processing has been carried out on the images: this suggests that possible improvements could be obtained using other features and applying pre-processing steps on images. Another possibility relates to the choice of the starting point of DC-SMIL and to a different management of the bundle.

In view of possible use in Computer Aided Diagnosis system dedicated to automated skin lesion detection, the creation of some framework that adopts multi-classifiers in order

to obtain performances of interest even in the distinction between dysplastic and common nevi would be most useful. The sensitivity parameter assumes particular importance in the medical field, effectively representing the number of correctly diagnosed patients. The good CPU time values open up the possibility of being able to evaluate multi-classification steps even for self-diagnosis applications.

“Success is never definitive, failure is never fatal; it is the courage to continue that counts.

– Winston Churchill

5.5 Results obtained with MIL approach

As reported in [2], it is not easy to compare the various methods. In fact, these methods use different machine learning approaches to distinguish melanoma lesions, and they have been applied on different data sets. However, SVM and ANN seem to be the most used methods. Another aspect to be emphasized is related to the features to be adopted in view of best classification performances. Beyond the quantity and type of features (color, texture, shape form or whatever), the features can be extracted in a global or local way. Global features are extracted taking the lesion as a whole, while local features are extracted from portions of the image. A local approach allows to increase the size of the feature vector, but also the complexity of the feature space.

We close the section by pictorially summarizing the results previously presented of MIL-RL in comparison to the results of the literature on melanoma detection using dermatoscopic images (see also Figure 5.1). MIL-RL algorithm has been used with images coming from the dermatoscopic data set PH^2 .

In Figure 5.12, where such results are reported, we have used the same notation as in Figure 5.1. We can appreciate how the obtained results are of interest, especially considering the restrictive hypotheses in which we have operated. We refer in particular to the absence of a pre-processing phase and to the use of only color features, i.e. mean and variance of RGB color channel. We have also experimented how effectively both the adoption of further features, as well as an adequate pre-processing phase allow us to improve the classification performance.

We observe here that DC-SMIL is a rather fast algorithm thus providing a promising tool for dealing with data sets made up of a large number of images.

5.6 Discussion

The spread of melanoma, both in terms of diagnosed cases and deaths, as well as for the growing availability of databases of dermoscopic images has increased the interest on tools for automatic classification of skin lesions. The choice and the design of classification models requires full awareness in the use of appropriate learning techniques and algorithms, and in their statistical validation. This task is urgent due to the low quality of the training data available as image annotation phase is very demanding. Multiple Instance Learning-type approaches seem fairly appropriate, as they are able to solve this aspect by allowing to manage images with only one global label. The imbalance between the classes of training

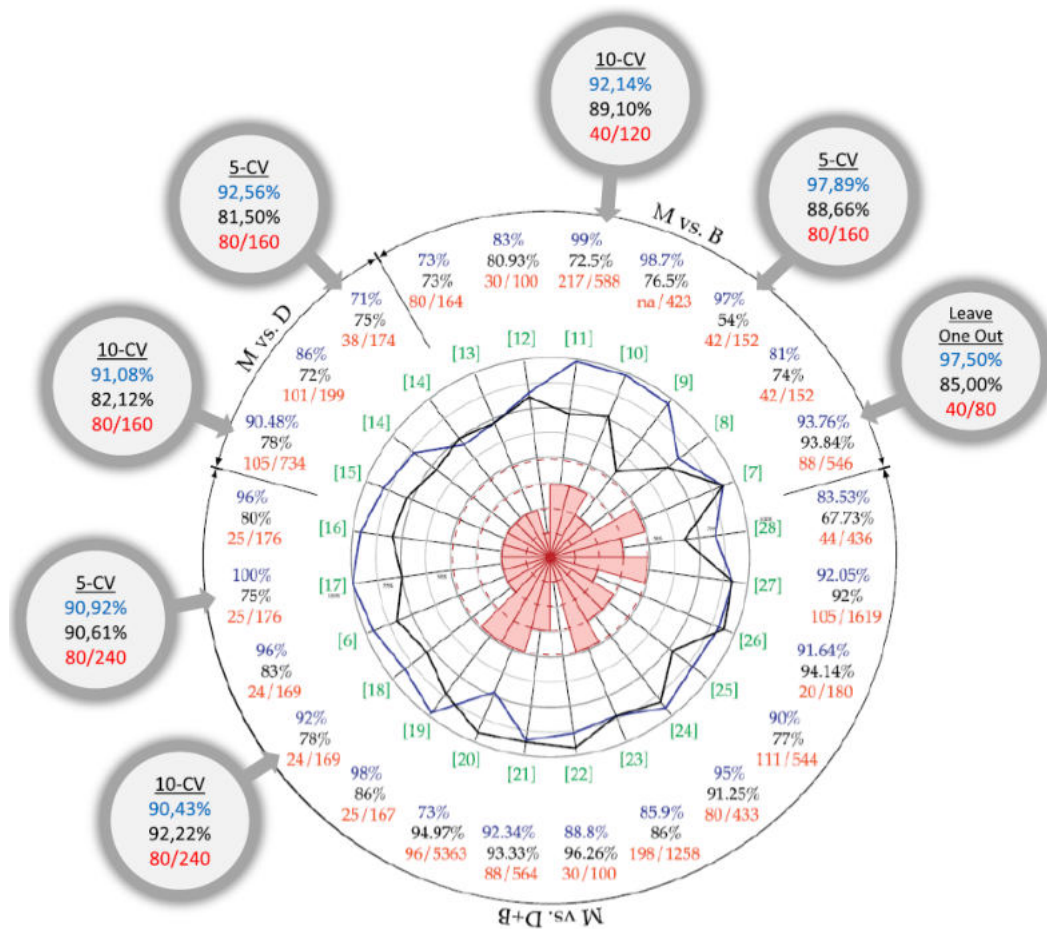


FIGURE 5.12: Comparison of obtained performances with the literature results

data sets should not be underestimated. The risk consists in undermining the classification performance of the models, which can manifest over-fitting, thus losing in generalization.

The most appropriate classification approach for a specific study is not easily identifiable. Each classification method has its own merits. Subject to solving the issues related to image acquisition and pre-processing steps, it is possible to obtain different classification results depending on a series of factors. The choice of the classifier, the difference in the composition of the various classes, and precisely the number of melanomas images in relation to the dysplastic and common nevi images, as well as features extraction and selection, are the most important factors that influence model's classification performance [145], [176], [177].

In recent years we have witnessed the rapid growth of deep learning (DL)-based systems in image processing which year after year sets new standards in terms of achieved performance. Melanoma detection, and more generally, medical image processing, can benefit from recent advances in deep learning-based methods that improve melanoma detection from the early-stage.

Feature selection is a very important aspect. The growing interest in DL architectures also lies in the fact that the choice of features can be made through the model. One of the inherent risks to the DL approach is the loss of sensitivity of the model. DL "believes" in the data submitted to it and does not check for inconsistencies and implausibility, errors in the

input material will be reflected in the outgoing results. The DL has an intrinsic "bias" that leads to consider what appears to be very frequent in data to be true. This is an open debate, especially in the medical field.

Few researchers have provided comparisons of different classification algorithms using the same set of images (see[215], [224], [246], [280]). From these comparative studies it emerges that some models such as multi layer perceptron (MLP) offer better performance than Bayesian and kNN classifiers, while SVM with MIL approaches outperformed the linear and RBF kernel versions.

This has given rise to solutions that include a combination of classifiers. In [317]–[320], some proposals for combination schemes of classifiers have been devised for dermoscopic images, also demonstrating that some of them constantly exceed the performance of a single best classifier. Even in this case which multi-classifier architecture is better, it depends on the specific application context.

The interaction of the specialist with a framework capable of providing a diagnostic support, together with a cultural model oriented to greater population proactivity through mobile self-diagnostic tools, is an emerging recipe for achieving a significant reduction in mortality rate of melanoma. The development of automated tools which support the clinician in detecting melanoma from its early-stage, tracking its evolution in time, and which could even be remotely used, represents an unprecedented opportunity to improve the way to detect this aggressive form of skin cancer.

The analysis of melanoma statistics has shown that this pathology is of particular importance both as regards new cases and deaths as well as for the growing trend in world areas characterized by high population density. The same statistics have confirmed that if skin cancers, and in particular melanoma, are quickly diagnosed, excision of the lesion tends to be decisive, which justifies the 5-year survival rates close to 100% (see Figure 4.3). The possibility of creating diagnostic tools capable of supporting specialists by providing increasingly accurate diagnoses becomes crucial.

The lack of adequate cultural models suggesting correct sun exposure practices, and the great diffusion of smartphones equipped with advanced cameras and fast network connection, open the possibility of creating applications that can support skin self-diagnosis, addressing people to specialist visits. Through the studies carried out and the obtained experimental results we have verified how instance space optimization models for Multiple Instance Learning constitute a valuable tool for the classification of medical images.

In order to have a comparison with the results from the literature, we have verified how using only some color features and without pre-processing steps, the adopted models guarantee interesting performances for classification of dermoscopic images. In particular, from the literature analysis, we have taken inspiration to present a new MIL model that uses spherical separation and that appears to be suitable for classifying data sets characterized by a high similarity and high dimension (see for example the case of dysplastic nevi versus common ones). The proposed model is therefore of interest in medical fields, where two images, although minimally different, may be indicative the one of a pathological state, the other of a healthy organ or tissue.

5.7 Future Work

The continuation of the research activities includes the development of new mathematical models to support medical image classification. Some of these models are being defined and they concern classification heuristics characterized by small CPU times. In addition to the classification measures such as those adopted in the presented experimental phases, the measures related to the processing speed are important, especially for the prototyping of applications designed for mobile use. One way to obtain better classification performances involves suitable pre-processing steps of the images, the evaluation of additional features for each blob, as well as the definition of more sophisticated segmentation strategies.

Another research direction concerns the application of MIL techniques in other medical contexts. In [321], we presented a study on the features that must be taken into consideration for the automatic classification of tongue images. The facility to disseminate and share information leads to the globalization of medical protocols which were previously used only in some world areas. This is the case, for example, of tongue inspection, widely used in Traditional Chinese Medicine (TCM) to perform a diagnosis about internal organs by simply observing the color and the consistency of patient's tongue. The current interest in tongue's image analysis is also motivated by the possibility of performing a first self-analysis on a possible disease suggesting further medical investigation.

This thesis work prefigures a scenario of possible applications for both medical and mobile use. This idea is the basis of the realization of a spin-off that will involve teachers from the Department of Computer Science, Modelling, Electronic and System Engineering (DIMES) of the University of Calabria and from the Bioinformatics Laboratory Surgical and Medical Science Department (DMSC) of the University Magna Graecia of Catanzaro in addition to the software house E-way Enterprise Business Solutions.

In [322], SIMPATICO 3D (Sistema Informativo Medico PATologIe COMplesse), was presented. SIMPATICO 3D is a system supporting scientists and physicians by providing facilities for the management, organization, analysis and distribution of medical data. A future work could be to add in SIMPATICO 3D a new module referring preliminary diagnosis on skin lesion, such as melanomas.

The studies carried out and the potential outlets on the market are the basis of a business idea, which revolves around some keywords: artificial intelligence, knowledge, things and humans. Starting from these keywords, the acronym for the Spin-off was also obtained: Artificial Intelligence for knowLedge, thIngs and humaNS (in short **AI-WINS**).

The main objective is to use artificial intelligence techniques and innovative technologies, such as the Internet of Things, for data analysis in particularly relevant sectors such as eHealth. The goal is the conception, design, development and marketing of innovative high-tech tools in the field of image analysis. Our aim is that AI-WINS will invest in solutions ready for the market, linked to image analysis for the diagnosis of melanoma. As we have shown, this type of diagnosis is far from simple because of the similarity of melanoma with other skin lesions such as dysplastic nevi.

The need to have tools to allow early diagnosis of melanoma able to precisely distinguish melanoma from benign lesions becomes a crucial challenge.

These tools will be used to support the specialist for the diagnosis of melanomas, and self-diagnosis through mobile applications. The reference area of the proposal is that of

Life Sciences, in order to satisfy the technological trajectories aimed at "Biomedical devices, biomechanics, systems and new medical and diagnostic applications".

The mission is located in the eHealth field and embraces different aspects by exploiting and promoting innovative solutions. Specifically, it provides a contribution to image processing, ranging from three-dimensional display problems to quantitative analysis problems for the automatic or semi-automatic extraction of diagnostic indices.

5.7.1 The market and competition

In recent decades, melanoma has become one of the most aggressive tumors; it is spreading rapidly in many areas of the world and data shows that, unfortunately, its incidence rate is characterized by a positive trend. Populations living in Europe, North America and Australia in particular are heavily affected by this type of skin cancer. Despite the ever increasing spread and its well-known aggression, if melanoma is identified in its initial stage (i.e. through early diagnosis) and is removed it can be treated without major complications for the individual, as evidenced by the survival rate at 5 years reported by the World Health Organization (WHO) [1]. The main problem therefore is that in the initial stages melanoma appears to be very similar to other benign lesions and it is difficult, even for experts in the sector, to identify it. In light of the scenario described above, it should therefore be noted that the referred market is the international one which requires stable and efficient diagnosis systems. Furthermore, the research that we conducted has faced, for the first time, the issue of classification of dysplastic from common nevi. Summing up, the results of our research have shown interesting performances than the ones present in literature.

5.7.2 Market value

Relatively to the support tool for the specialist of the diagnosis of melanomas, the specific reference sector is the *public and private health market*. In Europe, the total expense on eHealth technologies and digital services for health and healthcare/hospital care is expected to increase from the current \$ 3.39 billion to \$ 7.1 billion in 2024, according to Market Data Forecast estimates [323]. Finally, Octopus Ventures [324] has reported an increase in the use of digital technologies even in the treatment of mental illnesses and neurosciences, as evidenced by the financial resources interested in this sector, which went from 120 million pounds in 2014 to 580 million pounds in the 2019. For what concerns the dermatological instrument (app) for self-diagnosis, the specific reference sector is that of *dermatological devices* for the diagnosis of skin cancer which includes dermatoscopes, microscopes, and imaging instruments. The global market for dermatological devices has an annual growth rate of 11.50% and is estimated to reach \$ 14.17 billion by 2021. In addition, in 2016 the sector of devices for the diagnosis of skin cancer represented the largest market share.

The project idea could access funding sources linked to public instruments through participation in regional, national and European funding notices in the ICT field with particular concern in issues related to eHealth. In the short term, we aim to implement a prototype solution to be placed on the market, relying on the acquisition of additional skills in the eHealth field.

Appendix A: Evaluation of Classification Performance

To assess the quality of a classification model, both discrimination and calibration must be taken into account. Classification takes into account the way in which the two classes in the data set are separate while calibration measures how much the forecasts of a given model are close to the real underlying probability on the basis of specialized knowledge.

The formal definition of the measurements' sensitivity, specificity and accuracy derives from the quantification of the number of true positives (TP), true negatives (TN), false negatives (FN) and false positives (FP).

If a disease is actually present in a patient and the diagnostic test detects the disease, the test result is considered true positive (TP). Similarly, if a patient is healthy and the diagnostic test does not detect the disease, the test result is true negative (TN). However, if the diagnostic test indicates the presence of a disease in a healthy patient, the test result is false positive (FP). Similarly, if the diagnostic test result does not detect the disease in a sick patient, the test result is false negative (FN).

The measures we used to evaluate the methods considered are:

$$Correctness = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (6.1)$$

$$Sensitivity = \frac{TP}{TP + FN} \times 100\% \quad (6.2)$$

$$Specificity = \frac{TN}{TN + FP} \times 100\% \quad (6.3)$$

$$F_{Score} = 2 \times \frac{Sensitivity \times Specificity}{Sensitivity + Specificity} \quad (6.4)$$

$$CPU_{time} = \frac{CPU \text{ clock cycles}}{clock \text{ frequency}} \quad (6.5)$$

Correctness (6.1), also referred as "accuracy", specifies if a model is properly trained and how it can work in general. The problem with using correctness as the main performance metric is when the classes are significantly not balanced.

Sensitivity and specificity are the most commonly used performance evaluation parameters in the literature, at least in medical field. *Sensitivity* (6.2), also referred to as "precision", is a measure of correctly identifying non-healthy patients. *Specificity* (6.3) or "recall", is a measure of the capability to identify healthy patients (i.e., to avoid false positive patients).

To optimize the evaluation metric, *F-score* (6.4) is frequently used, which is a combined measure of sensitivity and specificity. A good F-score means we have low false positives and

low false negatives, so you are carefully identifying real threats and not being bothered by false alarms. F-score is considered perfect when it is 1, while the model fails when it is 0.

CPU time (6.5), or "process time", is the amount of time for which a central processing unit (CPU) is used for processing instructions of a computer program. The CPU time is measured in clock ticks or seconds. CPU time may decrease both by increasing the frequency of the clock or by decreasing the clock cycles needed to run the program.

In [317], Sboner et al., introduced d_{class} , which is a suitable measure to compare the performance of different classifiers. Using this parameter instead of the correctness, the comparison between classifiers can be done accurately, avoiding the unbalanced class problem.

A correct estimate of the discrimination and calibration of a model must take into account the effects of unbalanced classes and the relationship between training and testing sets. In fact, numerous studies have shown that the degradation of correctness is more evident on unbalanced data sets or when the classes of data tend to overlap [325]–[327]: these conditions typically occur for classification of skin lesions.

Many studies present numerical sections in which classifiers focus on the learning of more numerous classes (dysplastic and common nevi) than on smaller classes (melanoma). This results in a poor accuracy of the classification for small classes invalidating the diagnostic indication.

The relationship between training and testing sets is another important factor that influences classification results. As the size of the training set increases, the results improve [328]. The effect of this ratio on classification accuracy is studied in [319]: excessive training can lead the classifier to fit too much into training data (overfitting).

There are many possibilities for choosing the training and the testing data, taking into account that a test on separate data provides an unbiased estimate of the generalization error. One of them is the well known k -fold cross validation technique.

Regarding the classification performance of skin lesions, 5 and 10 fold cross validation are almost always adopted. If the original data set is too small for this approach, the recommended strategy is to use cross-validation [329] or bootstrap [330] in order to best use the available data. When k coincides with the cardinality of the entire dataset, the k -fold cross validation is known as *leave-one-out cross validation*: in such case, each time the testing set is constituted just by one element. Bootstrap is a flexible and powerful statistical tool that can be used to quantify the uncertainty associated with a given estimator or statistical learning method. It works by sampling with replacement from the original data, and take the "not chosen" data points as test cases. We can make this several times and calculate the average score as estimation of our model performance.

Bootstrap is rarely used in the literature for skin lesions, but has been shown to be superior to cross-validation on many other data sets [331].

Bibliography

- [1] WHO, *World health organization*, [Online; last consultation 18-november-2019], 2019. [Online]. Available: <http://gco.iarc.fr/today/explore>.
- [2] M. Rastgoo, R. Garcia, O. Morel, and F. Marzani, "Automatic differentiation of melanoma from dysplastic nevi", *Computerized Medical Imaging and Graphics*, vol. 43, pp. 44–52, 2015.
- [3] T. Mendonça, P. M. Ferreira, J. S. Marques, A. R. Marcal, and J. Rozeira, "Ph 2-a dermoscopic image database for research and benchmarking", in *2013 35th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*, IEEE, 2013, pp. 5437–5440.
- [4] S. Mustafa and A. Kimura, "A svm-based diagnosis of melanoma using only useful image features", in *2018 International Workshop on Advanced Image Technology (IWAIT)*, IEEE, 2018, pp. 1–4.
- [5] T. Mitchell, "Machine learning, mcgraw-hill higher education", *New York*, 1997.
- [6] L. S. Gottfredson, *Mainstream science on intelligence: An editorial with 52 signatories, history, and bibliography*, 1997.
- [7] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited, 2016.
- [8] A. Geitgey, "Machine learning is fun! part 4: Modern face recognition with deep learning", *Medium. com*, vol. 24, 2016.
- [9] S. Tufféry, *Data mining and statistics for decision making*. John Wiley & Sons, 2011.
- [10] E. Vocaturo and P. Veltri, "On the use of networks in biomedicine", *Procedia Computer Science*, vol. 110, pp. 498–503, 2017.
- [11] D. J. Hand, "Data mining", *Encyclopedia of Environmetrics*, vol. 2, 2006.
- [12] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.
- [13] P. M. Domingos, "A few useful things to know about machine learning.", *Commun. acm*, vol. 55, no. 10, pp. 78–87, 2012.
- [14] X. J. Zhu, "Semi-supervised learning literature survey", University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 2005.
- [15] P. Harrington, *Machine learning in action*. Manning Publications Co., 2012.
- [16] T. Hastie, R. Tibshirani, and J. Friedman, "Unsupervised learning", in *The elements of statistical learning*, Springer, 2009, pp. 485–585.
- [17] D. M. Hawkins, *Identification of outliers*. Springer, 1980, vol. 11.
- [18] A. K. Jain, R. C. Dubes, et al., *Algorithms for clustering data*. Prentice hall Englewood Cliffs, NJ, 1988, vol. 6.

- [19] V. Estivill-Castro, "Why so many clustering algorithms: A position paper.", *SIGKDD explorations*, vol. 4, no. 1, pp. 65–75, 2002.
- [20] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey", *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.
- [21] M. van Otterlo and M. Wiering, "Reinforcement learning and markov decision processes", in *Reinforcement Learning*, Springer, 2012, pp. 3–42.
- [22] D. P. Bertsekas, *Dynamic programming and optimal control*, 2. Athena scientific Belmont, MA, 1995, vol. 1.
- [23] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [24] A. Gosavi, "Simulation optimisation: Parametric optimisation techniques and reinforcement learning", *Norvell: Kluwer Academic Publishers*, 2003.
- [25] O. Chapelle, B. Scholkopf, A. Zien, *et al.*, "Semi-supervised learning, vol. 2", *Cambridge: MIT Press*. Cortes, C., & Mohri, M.(2014). *Domain adaptation and sample bias correction theory and algorithm for regression*. *Theoretical Computer Science*, vol. 519, p. 103 126, 2006.
- [26] H. Scudder, "Probability of error of some adaptive pattern-recognition machines", *IEEE Transactions on Information Theory*, vol. 11, no. 3, pp. 363–371, 1965.
- [27] A. Astorino and A. Fuduli, "Nonsmooth optimization techniques for semisupervised classification", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2135–2142, 2007.
- [28] J. Schurmann, *Pattern classification: a unified view of statistical and neural approaches*. Wiley New York, 1996.
- [29] P. E. Hart, *Pattern classification and scene analysis*. 1973.
- [30] B. D. Ripley and N. Hjort, *Pattern recognition and neural networks*. Cambridge university press, 1996.
- [31] R. P. Lippmann, "Pattern classification using neural networks", *IEEE communications magazine*, vol. 27, no. 11, pp. 47–50, 1989.
- [32] J. Lampinen, J. Laaksonen, and E. Oja, *Neural network systems, techniques and applications in pattern recognition*. Citeseer, 1997.
- [33] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers", *Machine learning*, vol. 29, no. 2-3, pp. 131–163, 1997.
- [34] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.
- [35] C. J. Burges, "A tutorial on support vector machines for pattern recognition", *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.
- [36] N. Cristianini, J. Shawe-Taylor, *et al.*, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.
- [37] J. Shawe-Taylor, N. Cristianini, *et al.*, *Kernel methods for pattern analysis*. Cambridge university press, 2004.
- [38] B. Scholkopf, C. J. Burges, A. J. Smola, *et al.*, *Advances in kernel methods: support vector learning*. MIT press, 1999.

- [39] M. G. Genton, "Classes of kernels for machine learning: A statistics perspective", *Journal of machine learning research*, vol. 2, no. Dec, pp. 299–312, 2001.
- [40] Y. Bengio *et al.*, "Learning deep architectures for ai", *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.
- [41] D. F. Specht, "Probabilistic neural networks", *Neural networks*, vol. 3, no. 1, pp. 109–118, 1990.
- [42] S. S. Haykin *et al.*, *Neural networks and learning machines/Simon Haykin*. New York: Prentice Hall, 2009.
- [43] S. S. Young, P. D. Scott, and N. M. Nasrabadi, "Object recognition using multilayer hopfield neural network", *IEEE Transactions on Image Processing*, vol. 6, no. 3, pp. 357–372, 1997.
- [44] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks", *IEEE transactions on acoustics, speech, and signal processing*, vol. 37, no. 3, pp. 328–339, 1989.
- [45] L. Grippo, "A class of unconstrained minimization methods for neural network training", *Optimization Methods and Software*, vol. 4, no. 2, pp. 135–150, 1994. DOI: [10.1080/10556789408805583](https://doi.org/10.1080/10556789408805583).
- [46] C. Charalambous, "Conjugate gradient algorithm for efficient training of artificial neural networks", *IEE Proceedings G (Circuits, Devices and Systems)*, vol. 139, no. 3, pp. 301–310, 1992.
- [47] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. N. Hultender, "Learning to rank using gradient descent", in *Proceedings of the 22nd International Conference on Machine learning (ICML-05)*, 2005, pp. 89–96.
- [48] V. Kumar, J. K. Chhabra, and D. Kumar, "Performance evaluation of distance metrics in the clustering algorithms", *INFOCOMP Journal of Computer Science*, vol. 13, no. 1, pp. 38–52, 2014.
- [49] R. Xu and D. Wunsch, *Clustering*. John Wiley & Sons, 2008, vol. 10.
- [50] R. Sibson, "Slink: An optimally efficient algorithm for the single-link cluster method", *The computer journal*, vol. 16, no. 1, pp. 30–34, 1973.
- [51] D. Defays, "An efficient algorithm for a complete link method", *The Computer Journal*, vol. 20, no. 4, pp. 364–366, 1977.
- [52] H.-P. Kriegel, P. Kroger, J. Sander, and A. Zimek, "Density-based clustering", *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 231–240, 2011.
- [53] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise.", in *Kdd*, vol. 96, 1996, pp. 226–231.
- [54] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: Ordering points to identify the clustering structure", in *ACM Sigmod record*, ACM, vol. 28, 1999, pp. 49–60.

- [55] E. Achtert, C. Bohm, and P. Kroger, "Deli-clu: Boosting robustness, completeness, usability, and efficiency of hierarchical clustering by a closest pair ranking", in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, 2006, pp. 119–128.
- [56] P. D. Tao *et al.*, "Minimum sum-of-squares clustering by dc programming and dca", in *International Conference on Intelligent Computing*, Springer, 2009, pp. 327–340.
- [57] W. Khalaf, A. Astorino, P d'Alessandro, and M. Gaudioso, "A dc optimization-based clustering technique for edge detection", *Optimization Letters*, vol. 11, no. 3, pp. 627–640, 2017.
- [58] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm", *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.
- [59] R. Sundberg, "Maximum likelihood theory for incomplete data from an exponential family", *Scandinavian Journal of Statistics*, pp. 49–58, 1974.
- [60] R. J. Rossi, *Mathematical Statistics: An Introduction to Likelihood Based Inference*. John Wiley & Sons, 2018.
- [61] I. J. Myung, "Tutorial on maximum likelihood estimation", *Journal of mathematical Psychology*, vol. 47, no. 1, pp. 90–100, 2003.
- [62] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- [63] C. J. Wu *et al.*, "On the convergence properties of the em algorithm", *The Annals of statistics*, vol. 11, no. 1, pp. 95–103, 1983.
- [64] L. Rabiner, "First hand: The hidden markov model", *IEEE Global History*, 2013.
- [65] B. Kamiński, M. Jakubczyk, and P. Szufel, "A framework for sensitivity analysis of decision trees", *Central European journal of operations research*, vol. 26, no. 1, pp. 135–159, 2018.
- [66] J. Rissanen, "A universal prior for integers and estimation by minimum description length", *The Annals of statistics*, pp. 416–431, 1983.
- [67] E. Hunt and J. S. Martin, *P.(1966), experiments in induction*.
- [68] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and regression trees. belmont, ca: Wadsworth", *International Group*, vol. 432, pp. 151–166, 1984.
- [69] S. K. Murthy, "Automatic construction of decision trees from data: A multi-disciplinary survey", *Data mining and knowledge discovery*, vol. 2, no. 4, pp. 345–389, 1998.
- [70] S. B. Kotsiantis, I Zaharakis, and P Pintelas, "Supervised machine learning: A review of classification techniques", *Emerging artificial intelligence applications in computer engineering*, vol. 160, pp. 3–24, 2007.
- [71] L. A. Breslow and D. W. Aha, "Simplifying decision trees: A survey", *The Knowledge Engineering Review*, vol. 12, no. 1, pp. 1–40, 1997.
- [72] I. Bruha, "From machine learning to knowledge discovery: Survey of preprocessing and postprocessing", *Intelligent Data Analysis*, vol. 4, no. 3-4, pp. 363–374, 2000.
- [73] T. Elomaa and J. Rousu, "General and efficient multisplitting of numerical attributes", *Machine learning*, vol. 36, no. 3, pp. 201–244, 1999.

- [74] T. G. Dietterich, "Ensemble methods in machine learning", in *International workshop on multiple classifier systems*, Springer, 2000, pp. 1–15.
- [75] L. Breiman, "Random forests, machine learning 45", *Journal of Clinical Microbiology*, vol. 2, pp. 199–228, 2001.
- [76] J. R. Quinlan, "Discovering rules by induction from large collections of examples", *Expert systems in the micro electronics age*, 1979.
- [77] J. Quinlan, "Program for machine learning", *C4. 5*, 1993.
- [78] S. Ruggieri, "Efficient c4. 5 [classification algorithm]", *IEEE transactions on knowledge and data engineering*, vol. 14, no. 2, pp. 438–444, 2002.
- [79] J. Gehrke, R. Ramakrishnan, and V. Ganti, "Rainforest—a framework for fast decision tree construction of large datasets", *Data Mining and Knowledge Discovery*, vol. 4, no. 2-3, pp. 127–162, 2000.
- [80] Q. Li and R. M. Nishikawa, *Computer-aided detection and diagnosis in medical imaging*. Taylor & Francis, 2015.
- [81] J. Cabestany, I. Rojas, and G. Joya, *Advances in Computational Intelligence: 11th International Work-Conference on Artificial Neural Networks, IWANN 2011, Torremolinos-Málaga, Spain, June 8-10, 2011, Proceedings*. Springer, 2011, vol. 6692.
- [82] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles", *Artificial intelligence*, vol. 89, no. 1-2, pp. 31–71, 1997.
- [83] J. Amores, "Multiple instance classification: Review, taxonomy and comparative study", *Artificial intelligence*, vol. 201, pp. 81–105, 2013.
- [84] M.-A. Carbonneau, V. Cheplygina, E. Granger, and G. Gagnon, "Multiple instance learning: A survey of problem characteristics and applications", *Pattern Recognition*, vol. 77, pp. 329–353, 2018.
- [85] J. Hoffman, D. Pathak, T. Darrell, and K. Saenko, "Detector discovery in the wild: Joint multiple instance and representation learning", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2883–2891.
- [86] J. Wu, Y. Yu, C. Huang, and K. Yu, "Deep multiple instance learning for image classification and auto-annotation", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3460–3469.
- [87] P. Sudharshan, C. Petitjean, F. Spanhol, L. E. Oliveira, L. Heutte, and P. Honeine, "Multiple instance learning for histopathological breast cancer image classification", *Expert Systems with Applications*, vol. 117, pp. 103–111, 2019.
- [88] G. Quellec, G. Cazuguel, B. Cochener, and M. Lamard, "Multiple-instance learning for medical image and video analysis", *IEEE reviews in biomedical engineering*, vol. 10, pp. 213–234, 2017.
- [89] W. Li, L. Duan, D. Xu, and I. W.-H. Tsang, "Text-based image retrieval using progressive multi-instance learning", in *2011 International Conference on Computer Vision, IEEE*, 2011, pp. 2049–2055.
- [90] J. Salamon, B. McFee, P. Li, and J. P. Bello, "Dcase 2017 submission: Multiple instance learning for sound event detection", *DCASE2017 Challenge, Tech. Rep*, 2017.

- [91] V. Cheplygina, D. M. Tax, and M. Loog, "On classification with bags, groups and sets", *Pattern recognition letters*, vol. 59, pp. 11–17, 2015.
- [92] G. Vanwinckelen, D. Fierens, H. Blockeel, *et al.*, "Instance-level accuracy versus bag-level accuracy in multi-instance learning", *Data mining and knowledge discovery*, vol. 30, no. 2, pp. 313–341, 2016.
- [93] E. Alpaydın, V. Cheplygina, M. Loog, and D. M. Tax, "Single-vs. multiple-instance classification", *Pattern recognition*, vol. 48, no. 9, pp. 2831–2838, 2015.
- [94] J. Foulds and E. Frank, "A review of multi-instance learning assumptions", *The Knowledge Engineering Review*, vol. 25, no. 1, pp. 1–25, 2010.
- [95] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning", in *Advances in neural information processing systems*, 2003, pp. 577–584.
- [96] A. Astorino, A. Fuduli, and M. Gaudioso, "A lagrangian relaxation approach for binary multiple instance classification", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 9, pp. 2662–2671, 2019. DOI: [10.1109/TNNLS.2018.2885852](https://doi.org/10.1109/TNNLS.2018.2885852).
- [97] Y. Xiao, B. Liu, and Z. Hao, "A sphere-description-based approach for multiple-instance learning", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 2, pp. 242–257, 2017.
- [98] M.-A. Carbonneau, E. Granger, A. J. Raymond, and G. Gagnon, "Robust multiple-instance learning ensembles using random subspace instance selection", *Pattern recognition*, vol. 58, pp. 83–99, 2016.
- [99] G. Doran and S. Ray, "Multiple-instance learning from distributions", *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 4384–4433, 2016.
- [100] S. Ray and M. Craven, "Supervised versus multiple instance learning: An empirical comparison", in *Proceedings of the 22nd international conference on Machine learning*, ACM, 2005, pp. 697–704.
- [101] C. Zhang, J. C. Platt, and P. A. Viola, "Multiple instance boosting for object detection", in *Advances in neural information processing systems*, 2006, pp. 1417–1424.
- [102] Y. Jia and C. Zhang, "Instance-level semisupervised multiple instance learning.", in *AAAI*, 2008, pp. 640–645.
- [103] M. Carbonneau, E Granger, and G Gagnon, "Decision threshold adjustment strategies for increased accuracy in multiple instance learning", in *Proceedings of the international conference on image processing theory, tools and application*, 2016.
- [104] C. Yang, M. Dong, and J. Hua, "Region-based image annotation using asymmetrical support vector machine-based multiple-instance learning", in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 06)*, IEEE, vol. 2, 2006, pp. 2057–2063.
- [105] W. Li and D. Y. yeung, "Mild: Multiple-instance learning via disambiguation", *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 1, pp. 76–89, 2010.
- [106] A. Blum and A. Kalai, "A note on learning from multiple-instance examples", *Machine Learning*, vol. 30, no. 1, pp. 23–29, 1998.

- [107] O. Maron and T. Lozano-Pérez, "A framework for multiple-instance learning", in *Advances in neural information processing systems*, 1998, pp. 570–576.
- [108] X.-S. Wei, J. Wu, and Z.-H. Zhou, "Scalable multi-instance learning", in *2014 IEEE International Conference on Data Mining*, IEEE, 2014, pp. 1037–1042.
- [109] R. C. Bunescu and R. J. Mooney, "Multiple instance learning for sparse positive bags", in *Proceedings of the 24th international conference on Machine learning*, ACM, 2007, pp. 105–112.
- [110] Z.-H. Zhou, Y.-Y. Sun, and Y.-F. Li, "Multi-instance learning by treating instances as non-iid samples", in *Proceedings of the 26th annual international conference on machine learning*, ACM, 2009, pp. 1249–1256.
- [111] D. Zhang, Y. Liu, L. Si, J. Zhang, and R. D. Lawrence, "Multiple instance learning on structured data", in *Advances in Neural Information Processing Systems*, 2011, pp. 145–153.
- [112] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies", 2008.
- [113] M. S. Ryoo and J. K. Aggarwal, "Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities.", in *ICCV*, Citeseer, vol. 1, 2009, p. 2.
- [114] S. Yan, X. Zhu, G. Liu, and J. Wu, "Sparse multiple instance learning as document classification", *Multimedia tools and applications*, vol. 76, no. 3, pp. 4553–4570, 2017.
- [115] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories", in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 06)*, IEEE, vol. 2, 2006, pp. 2169–2178.
- [116] J. Amores, "Vocabulary-based approaches for multiple-instance data: A comparative study", in *2010 20th International Conference on Pattern Recognition*, IEEE, 2010, pp. 4246–4250.
- [117] Q. Wang, L. Si, and D. Zhang, "A discriminative data-dependent mixture-model approach for multiple instance learning in image classification", in *European Conference on Computer Vision*, Springer, 2012, pp. 660–673.
- [118] Q. Zhang and S. A. Goldman, "Em-dd: An improved multiple-instance learning technique", in *Advances in neural information processing systems*, 2002, pp. 1073–1080.
- [119] D. M. Tax and R. P. Duin, "Learning curves for the analysis of multiple instance classifiers", in *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, Springer, 2008, pp. 724–733.
- [120] J. Wang and J.-D. Zucker, "Solving multiple-instance problem: A lazy learning approach", 2000.
- [121] Z. Wang, Z. Zhao, and C. Zhang, "Learning with only multiple instance positive bags", in *2016 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2016, pp. 334–341.

- [122] R. Venkatesan, P. Chandakkar, and B. Li, "Simpler non-parametric methods provide as good or better results to multiple-instance learning", in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2605–2613.
- [123] W. Li and N. Vasconcelos, "Multiple instance learning for soft bags via top instances", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4277–4285.
- [124] O. L. Mangasarian and E. W. Wild, "Multiple instance classification via successive linear programming", *Journal of Optimization Theory and Applications*, vol. 137, no. 3, pp. 555–568, 2008.
- [125] Z.-H. Zhou, "Multi-instance learning: A survey", *Department of Computer Science & Technology, Nanjing University, Tech. Rep*, 2004.
- [126] R. Rahmani and S. A. Goldman, "Missl: Multiple-instance semi-supervised learning", in *Proceedings of the 23rd international conference on Machine learning*, ACM, 2006, pp. 705–712.
- [127] B. Babenko, "Multiple instance learning: Algorithms and applications", *View Article PubMed/NCBI Google Scholar*, pp. 1–19, 2008.
- [128] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning", in *2009 IEEE Conference on computer vision and Pattern Recognition*, IEEE, 2009, pp. 983–990.
- [129] —, "Robust object tracking with online multiple instance learning", *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 8, pp. 1619–1632, 2010.
- [130] C. Bergeron, G. Moore, J. Zaretzki, C. M. Breneman, and K. P. Bennett, "Fast bundle algorithm for multiple-instance learning", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 6, pp. 1068–1079, 2011.
- [131] S. Sabato and N. Tishby, "Multi-instance learning with any hypothesis class", *Journal of Machine Learning Research*, vol. 13, no. Oct, pp. 2999–3039, 2012.
- [132] G. Doran and S. Ray, "A theoretical and empirical analysis of support vector machine methods for multiple-instance classification", *Machine Learning*, vol. 97, no. 1-2, pp. 79–102, 2014.
- [133] Y. Xu, T. Mo, Q. Feng, P. Zhong, M. Lai, and E. I. Chang, "Deep learning of feature representation with multiple instance learning for medical image analysis", in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 1626–1630. DOI: [10.1109/ICASSP.2014.6853873](https://doi.org/10.1109/ICASSP.2014.6853873).
- [134] V. Cheplygina and D. M. Tax, "Characterizing multiple instance datasets", in *International Workshop on Similarity-Based Pattern Recognition*, Springer, 2015, pp. 15–27.
- [135] M. Kandemir and F. A. Hamprecht, "Computer-aided diagnosis from weak supervision: A benchmarking study", *Computerized medical imaging and graphics*, vol. 42, pp. 44–50, 2015.
- [136] F. Herrera, S. Ventura, R. Bello, C. Cornelis, A. Zafra, D. Sanchez Tarrago, and S. Vluymans, *Multiple Instance Learning. Foundations and Algorithms*. Oct. 2016, ISBN: 978-3-319-47758-9. DOI: [10.1007/978-3-319-47759-6](https://doi.org/10.1007/978-3-319-47759-6).

- [137] X.-S. Wei and Z.-H. Zhou, "An empirical study on image bag generators for multi-instance learning", *Machine Learning*, 2016. [Online]. Available: <https://doi.org/10.1007/s10994-016-5560-1>.
- [138] D. Komura and S. Ishikawa, "Machine learning methods for histopathological image analysis", *Computational and structural biotechnology journal*, vol. 16, pp. 34–42, 2018.
- [139] V. Cheplygina, M. de Bruijne, and J. P. Pluim, "Not-so-supervised: A survey of semi-supervised, multi-instance, and transfer learning in medical image analysis", *Medical image analysis*, vol. 54, pp. 280–296, 2019.
- [140] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, and J. Dean, "A guide to deep learning in healthcare", *Nature medicine*, vol. 25, no. 1, p. 24, 2019.
- [141] D. M. Tax and R. P. Duin, "Data domain description using support vectors.", in *ESANN*, vol. 99, 1999, pp. 251–256.
- [142] A. Astorino and M. Gaudioso, "A fixed-center spherical separation algorithm with kernel transformations for classification problems", *Computational Management Science*, vol. 6, no. 3, pp. 357–372, 2009.
- [143] F. Plastria, E. Carrizosa, and J. Gordillo, "Multi-instance classification through spherical separation and vns", *Computers & Operations Research*, vol. 52, pp. 326–333, 2014.
- [144] A. Astorino, A. Fuduli, and M. Gaudioso, "Margin maximization in spherical separation", *Computational Optimization and Applications*, vol. 53, no. 2, pp. 301–322, 2012.
- [145] M. Gaudioso, G. Giallombardo, G. Miglionico, and E. Vocaturo, "Classification in the multiple instance learning framework via spherical separation", *Soft Computing*, pp. 1–7, 2019.
- [146] W. de Oliveira, "Proximal bundle methods for nonsmooth dc programming", *Journal of Global Optimization*, pp. 1–41, 2019.
- [147] M. Gaudioso, G. Giallombardo, and G. Miglionico, "Minimizing piecewise-concave functions over polyhedra", *Mathematics of Operations Research*, vol. 43, no. 2, pp. 580–597, 2017.
- [148] M. Gaudioso, G. Giallombardo, G. Miglionico, and A. M. Bagirov, "Minimizing non-smooth dc functions via successive dc piecewise-affine approximations", *Journal of Global Optimization*, vol. 71, no. 1, pp. 37–55, 2018.
- [149] K. Joki, A. M. Bagirov, N. Karmita, and M. M. Makela, "A proximal bundle method for nonsmooth dc optimization utilizing nonconvex cutting planes", *Journal of Global Optimization*, vol. 68, no. 3, pp. 501–535, 2017.
- [150] K. Joki, A. M. Bagirov, N. Karmita, M. M. Makela, and S. Taheri, "Double bundle method for finding clarke stationary points in nonsmooth dc programming", *SIAM Journal on Optimization*, vol. 28, no. 2, pp. 1892–1919, 2018.
- [151] A. Strekalovsky, "On global optimality conditions for dc programming problems", *Technical Paper, Irkutsk State University*, 1997.
- [152] J.-B. Hiriart-Urruty, "Generalized differentiability/duality and optimization for problems dealing with differences of convex functions", in *Convexity and duality in optimization*, Springer, 1985, pp. 37–70.

- [153] —, “From convex optimization to nonconvex optimization. necessary and sufficient conditions for global optimality”, in *Nonsmooth optimization and related topics*, Springer, 1989, pp. 219–239.
- [154] A. Astorino and G. Miglionico, “Optimizing sensor cover energy via dc programming”, *Optimization Letters*, vol. 10, no. 2, pp. 355–368, 2016.
- [155] P. D. Tao *et al.*, “The dc (difference of convex functions) programming and dca revisited with dc models of real world nonconvex optimization problems”, *Annals of operations research*, vol. 133, no. 1-4, pp. 23–46, 2005.
- [156] J. E. Kelley Jr, “The cutting-plane method for solving convex programs”, *Journal of the society for Industrial and Applied Mathematics*, vol. 8, no. 4, pp. 703–712, 1960.
- [157] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex analysis and minimization algorithms I: Fundamentals*. Springer science & business media, 2013, vol. 305.
- [158] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik, “Blobworld: A system for region-based image indexing and retrieval”, in *International conference on advances in visual information systems*, Springer, 1999, pp. 509–517.
- [159] S. Robertson and D. A. Hull, “The trec-9 filtering track final report”, in *TREC*, Cite-seer, 2000, pp. 25–40.
- [160] *Ibm ilog cplex optimizer*, <http://www.cplex.com>.
- [161] A. Astorino, A. Fuduli, G. Giallombardo, and G. Miglionico, “Svm-based multiple instance classification via dc optimization”, *Algorithms*, vol. 12, no. 12, p. 249, 2019.
- [162] W. de Oliveira and M. P. Tcheou, “An inertial algorithm for dc programming”, *Set-Valued and Variational Analysis*, pp. 1–25, 2018.
- [163] O. Maron and A. L. Ratan, “Multiple-instance learning for natural scene classification.”, in *ICML*, vol. 98, 1998, pp. 341–349.
- [164] R. L. Siegel, K. D. Miller, and A. Jemal, “Cancer statistics, 2019”, *CA: a cancer journal for clinicians*, vol. 69, no. 1, pp. 7–34, 2019.
- [165] R. Sanghera and P. S. Grewal, “Dermatological symptom assessment”, in *Patient Assessment in Clinical Pharmacy*, Springer, 2019, pp. 133–154.
- [166] G. Argenziano, G. Fabbrocini, P. Carli, V. De Giorgi, E. Sammarco, and M. Delfino, “Epiluminescence microscopy for the diagnosis of doubtful melanocytic skin lesions: Comparison of the abcd rule of dermatoscopy and a new 7-point checklist based on pattern analysis”, *Archives of dermatology*, vol. 134, no. 12, pp. 1563–1570, 1998.
- [167] G. Anastasi, S. Capitani, M. Carnazza, S Cinti, O Cremona, R De Caro, R. Donato, V. Ferrario, L Fonzi, A. Franzi, *et al.*, “Trattato di anatomia umana”, 2010.
- [168] A. R. Sadri, M. Zekri, S. Sadri, N. Gheissari, M. Mokhtari, and F. Kolahdouzan, “Segmentation of dermoscopy images using wavelet networks”, *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 4, pp. 1134–1141, 2012.
- [169] WHO, *American academy of dermatology*, [Online; last consultation 18-november-2019], 2019. [Online]. Available: <https://www.aad.org/skin-cancer-melanoma>.
- [170] S. Menzies, C Ingvar, and W. McCarthy, “A sensitivity and specificity analysis of the surface microscopy features of invasive melanoma.”, *Melanoma research*, vol. 6, no. 1, pp. 55–62, 1996.

- [171] R. J. Pariser and D. M. Pariser, "Primary care physicians' errors in handling cutaneous disorders: A prospective survey", *Journal of the American Academy of Dermatology*, vol. 17, no. 2, pp. 239–245, 1987.
- [172] C. Pleiss, J. H. Risse, H.-J. Biersack, and H. Bender, "Role of fdg-pet in the assessment of survival prognosis in melanoma", *Cancer biotherapy & radiopharmaceuticals*, vol. 22, no. 6, pp. 740–747, 2007.
- [173] P. Aberg, I. Nicander, J. Hansson, P. Geladi, U. Holmgren, and S. Ollmar, "Skin cancer identification using multifrequency electrical impedance—a potential screening tool", *IEEE transactions on biomedical engineering*, vol. 51, no. 12, pp. 2097–2102, 2004.
- [174] S. Sigurdsson, P. A. Philipsen, L. K. Hansen, J. Larsen, M. Gniadecka, and H.-C. Wulf, "Detection of skin cancer by classification of raman spectra", *IEEE transactions on biomedical engineering*, vol. 51, no. 10, pp. 1784–1793, 2004.
- [175] T. Maier, D. Kulichova, K. Schotten, R. Astrid, T. Ruzicka, C. Berking, and A. Udrea, "Accuracy of a smartphone application using fractal image analysis of pigmented moles compared to clinical diagnosis and histological result", *Journal of the European Academy of Dermatology and Venereology*, vol. 29, no. 4, pp. 663–667, 2015.
- [176] A. Astorino, A. Fuduli, P. Veltri, and E. Vocaturo, "On a recent algorithm for multiple instance learning. preliminary applications in image classification", in *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, IEEE, 2017, pp. 1615–1619.
- [177] A. Astorino, A. Fuduli, M. Gaudio, and E. Vocaturo, "A multiple instance learning algorithm for color images classification", in *Proceedings of the 22nd International Database Engineering & Applications Symposium*, ACM, 2018, pp. 262–266.
- [178] C. Demir and B. Yener, "Automated cancer diagnosis based on histopathological images: A systematic survey", *Rensselaer Polytechnic Institute, Tech. Rep*, 2005.
- [179] S. S. Agaian, K. Panetta, and A. M. Grigoryan, "A new measure of image enhancement", in *IASTED International Conference on Signal Processing & Communication*, 2000, pp. 19–22.
- [180] H. T. Lau and A. Al-Jumaily, "Automatically early detection of skin cancer: Study based on neural network classification", in *2009 International Conference of Soft Computing and Pattern Recognition*, IEEE, 2009, pp. 375–380.
- [181] L. Zhang and X. Wu, "An edge-guided image interpolation algorithm via directional filtering and data fusion", *IEEE transactions on Image Processing*, vol. 15, no. 8, pp. 2226–2238, 2006.
- [182] N. N. Sultana and N. B. Puhan, "Recent deep learning methods for melanoma detection: A review", in *International Conference on Mathematics and Computing*, Springer, 2018, pp. 118–132.
- [183] R. Garnavi, M. Aldeen, M. E. Celebi, A. Bhuiyan, C. Dolianitis, and G. Varigos, "Automatic segmentation of dermoscopy images using histogram thresholding on optimal color channels", *International Journal of Medicine and Medical Sciences*, vol. 1, no. 2, pp. 126–134, 2010.

- [184] P Barbini, G Cevenini, P Rubegni, M. Massai, M. Flori, P Carli, and L Andreassi, "Instrumental measurement of skin colour and skin type as risk factors for melanoma: A statistical classification procedure.", *Melanoma research*, vol. 8, no. 5, pp. 439–447, 1998.
- [185] M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta, "A standard default color space for the internet-srgb", *Microsoft and Hewlett-Packard Joint Report*, 1996.
- [186] S. S. Al-amri, N. Kalyankar, and S. Khamitkar, "Linear and non-linear contrast enhancement image", *International Journal of Computer Science and Network Security*, vol. 10, no. 2, pp. 139–143, 2010.
- [187] R. D. Fiete, *Modeling the imaging chain of digital cameras*. SPIE press Bellingham, 2010.
- [188] J. Lu, D. M. Healy, and J. B. Weaver, "Contrast enhancement of medical images using multiscale edge representation", *Optical engineering*, vol. 33, no. 7, pp. 2151–2162, 1994.
- [189] D. D. Gómez, C. Butakoff, B. K. Ersboll, and W. Stoecker, "Independent histogram pursuit for segmentation of skin lesions", *IEEE transactions on biomedical engineering*, vol. 55, no. 1, pp. 157–161, 2007.
- [190] A. Sultana, M. Ciuc, T. Radulescu, L. Wanyu, and D. Petrache, "Preliminary work on dermatoscopic lesion segmentation", in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, IEEE, 2012, pp. 2273–2277.
- [191] R. L. Lagendijk and J. Biemond, "Basic methods for image restoration and identification", in *The essential guide to image processing*, Elsevier, 2009, pp. 323–348.
- [192] N Radhika and T. Antony, "Image denoising techniques preserving edge", *ACEEE Int. J. on Information Technology*, vol. 1, no. 02, 2011.
- [193] P. Patidar, M. Gupta, S. Srivastava, and A. K. Nagawat, "Image de-noising by various filters for different noise", *International journal of computer applications*, vol. 9, no. 4, pp. 45–50, 2010.
- [194] P. F. Manfredi, P. Maranesi, and T. Tacchi, *L'amplificatore operazionale*. Bollati Boringhieri, 1993.
- [195] M. C. Motwani, M. C. Gadiya, R. C. Motwani, and F. C. Harris, "Survey of image denoising techniques", in *Proceedings of GSPX*, 2004, pp. 27–30.
- [196] D. Rao and P. P. Panduranga, "A survey on image enhancement techniques: Classical spatial filter, neural network, cellular neural network, and fuzzy filter", in *2006 IEEE International Conference on Industrial Technology*, IEEE, 2006, pp. 2821–2826.
- [197] B. Shinde, D. Mhaske, M. Patare, A. Dani, and A. Dani, "Apply different filtering techniques to remove the speckle noise using medical images", *International Journal of Engineering Research and Applications*, vol. 2, no. 1, pp. 1071–1079, 2012.
- [198] R. Garnavi, M. Aldeen, and J. Bailey, "Classification of melanoma lesions using wavelet-based texture analysis", in *2010 International Conference on Digital Image Computing: Techniques and Applications*, IEEE, 2010, pp. 75–81.
- [199] K Madhankumar and P Kumar, "Characterization of skin lesions", in *International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012)*, IEEE, 2012, pp. 302–306.

- [200] A. Kaur and V. Chopra, "Blind image deconvolution technique for image restoration using ant colony optimization", *Int. J. Comput. Appl. Inf. Technol.*, vol. 1, no. 2, pp. 55–59, 2012.
- [201] Z. Zhao and R. E. Blahut, "Blind and nonblind nonnegative impulse response isi channel demodulation using the richardson-lucy algorithm", in *Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology, 2005.*, IEEE, 2005, pp. 445–450.
- [202] C. Khare and K. K. Nagwanshi, "Implementation and analysis of image restoration techniques", *International Journal of Computer Trends and Technology-May to June*, no. 2011, 2011.
- [203] J. Benesty, J. Chen, Y. A. Huang, and S. Doclo, "Study of the wiener filter for noise reduction", in *Speech Enhancement*, Springer, 2005, pp. 9–41.
- [204] Q. Abbas, M. E. Celebi, and I. F. García, "Hair removal methods: A comparative study for dermoscopy images", *Biomedical Signal Processing and Control*, vol. 6, no. 4, pp. 395–404, 2011.
- [205] T. Lee, V. Ng, R. Gallagher, A. Coldman, and D. McLean, "Dullrazor®: A software approach to hair removal from images", *Computers in biology and medicine*, vol. 27, no. 6, pp. 533–543, 1997.
- [206] H. Mirzaalian, T. K. Lee, and G. Hamarneh, "Learning features for streak detection in dermoscopic color images using localized radial flux of principal intensity curvature", in *2012 IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, IEEE, 2012, pp. 97–101.
- [207] M. G. Fleming, C. Steger, J. Zhang, J. Gao, A. B. Cognetta, C. R. Dyer, *et al.*, "Techniques for a structural analysis of dermatoscopic imagery", *Computerized medical imaging and graphics*, vol. 22, no. 5, pp. 375–389, 1998.
- [208] A. Khan, K. Gupta, R. J. Stanley, W. V. Stoecker, R. H. Moss, G. Argenziano, H. P. Soyer, H. S. Rabinovitz, and A. B. Cognetta, "Fuzzy logic techniques for blotch feature evaluation in dermoscopy images", *Computerized Medical Imaging and Graphics*, vol. 33, no. 1, pp. 50–57, 2009.
- [209] M. E. Celebi, H. Iyatomi, W. V. Stoecker, R. H. Moss, H. S. Rabinovitz, G. Argenziano, and H. P. Soyer, "Automatic detection of blue-white veil and related structures in dermoscopy images", *Computerized Medical Imaging and Graphics*, vol. 32, no. 8, pp. 670–677, 2008.
- [210] W. V. Stoecker, M. Wronkiewicz, R. Chowdhury, R. J. Stanley, J. Xu, A. Bangert, B. Shrestha, D. A. Calcara, H. S. Rabinovitz, M. Oliviero, *et al.*, "Detection of granularity in dermoscopy images of malignant melanoma using color and texture features", *Computerized Medical Imaging and Graphics*, vol. 35, no. 2, pp. 144–147, 2011.
- [211] I. Maglogiannis, S. Pavlopoulos, and D. Koutsouris, "An integrated computer supported acquisition, handling, and characterization system for pigmented skin lesions in dermatological images", *IEEE Transactions on Information Technology in Biomedicine*, vol. 9, no. 1, pp. 86–98, 2005.

- [212] T. Tanaka, S. Torii, I. Kabuta, K. Shimizu, and M. Tanaka, "Pattern classification of nevus with texture analysis", *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 3, no. 1, pp. 143–150, 2008.
- [213] N. R. Abbasi, H. M. Shaw, D. S. Rigel, R. J. Friedman, W. H. McCarthy, I. Osman, A. W. Kopf, and D. Polsky, "Early diagnosis of cutaneous melanoma: Revisiting the abcd criteria", *Jama*, vol. 292, no. 22, pp. 2771–2776, 2004.
- [214] H. Zhou, M. Chen, and J. M. Rehg, "Dermoscopic interest point detector and descriptor", in *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, IEEE, 2009, pp. 1318–1321.
- [215] G. Surowka and K. Grzesiak-Kopec, "Different learning paradigms for the classification of melanoid skin lesions using wavelets", in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE, 2007, pp. 3136–3139.
- [216] C. Lee and D. A. Landgrebe, "Decision boundary feature extraction for neural networks", *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 75–83, 1997.
- [217] I. Maglogiannis, E. Zafiroopoulos, and C. Kyranoudis, "Intelligent segmentation and classification of pigmented skin lesions in dermatological images", in *Hellenic Conference on Artificial Intelligence*, Springer, 2006, pp. 214–223.
- [218] M. Healsmith, J. Bourke, J. Osborne, and R. Graham-Brown, "An evaluation of the revised seven-point checklist for the early diagnosis of cutaneous malignant melanoma", *British Journal of Dermatology*, vol. 130, no. 1, pp. 48–50, 1994.
- [219] H. P. Soyer, G. Argenziano, I. Zalaudek, R. Corona, F. Sera, R. Talamini, F. Barbato, A. Baroni, L. Cicale, A. Di Stefani, *et al.*, "Three-point checklist of dermoscopy", *Dermatology*, vol. 208, no. 1, pp. 27–31, 2004.
- [220] H. Pehamberger, A. Steiner, and K. Wolff, "In vivo epiluminescence microscopy of pigmented skin lesions. i. pattern analysis of pigmented skin lesions", *Journal of the American Academy of Dermatology*, vol. 17, no. 4, pp. 571–583, 1987.
- [221] S. W. Menzies, *An atlas of surface microscopy of pigmented skin lesions: dermoscopy*. McGraw Hill Professional, 2003.
- [222] R. H. Johr, "Dermoscopy: Alternative melanocytic algorithms—the abcd rule of dermoscopy, menzies scoring method, and 7-point checklist", *Clinics in dermatology*, vol. 20, no. 3, pp. 240–247, 2002.
- [223] C. Dolianitis, J. Kelly, R. Wolfe, and P. Simpson, "Comparative performance of 4 dermoscopic algorithms by nonexperts for the diagnosis of melanocytic lesions", *Archives of dermatology*, vol. 141, no. 8, pp. 1008–1014, 2005.
- [224] I. Maglogiannis and C. N. Doukas, "Overview of advanced computer vision systems for skin lesions characterization", *IEEE transactions on information technology in biomedicine*, vol. 13, no. 5, pp. 721–733, 2009.
- [225] W. V. Stoecker, K. Gupta, R. J. Stanley, R. H. Moss, and B. Shrestha, "Detection of asymmetric blotches (asymmetric structureless areas) in dermoscopy images of malignant melanoma using relative color", *Skin Research and Technology*, vol. 11, no. 3, pp. 179–184, 2005.

- [226] M. E. Celebi, H. A. Kingravi, B. Uddin, H. Iyatomi, Y. A. Aslandogan, W. V. Stoecker, and R. H. Moss, "A methodological approach to the classification of dermoscopy images", *Computerized Medical imaging and graphics*, vol. 31, no. 6, pp. 362–373, 2007.
- [227] D. S. Rigel, J. Russak, and R. Friedman, "The evolution of melanoma diagnosis: 25 years beyond the abcds", *CA: a cancer journal for clinicians*, vol. 60, no. 5, pp. 301–316, 2010.
- [228] E. L. Psaty and A. C. Halpern, "Current and emerging technologies in melanoma diagnosis: The state of the art", *Clinics in dermatology*, vol. 27, no. 1, pp. 35–45, 2009.
- [229] S. Nakariyakul and D. P. Casasent, "Improved forward floating selection algorithm for feature subset selection", in *2008 International Conference on Wavelet Analysis and Pattern Recognition*, IEEE, vol. 2, 2008, pp. 793–798.
- [230] G. Argenziano, H. Soyer, V De Giorgi, D. Piccolo, P. Carli, and M. Delfino, "Interactive atlas of dermoscopy (book and cd-rom)", 2000.
- [231] T. Schindewolf, R. Schiffner, W. Stolz, R. Albert, W. Abmayr, and H. Harms, "Evaluation of different image acquisition techniques for a computer vision system in the diagnosis of malignant melanoma", *Journal of the American Academy of Dermatology*, vol. 31, no. 1, pp. 33–41, 1994.
- [232] H. E. Exner and H. Hougardy, "Quantitative image analysis of microstructures", *Oberursel, DGM Informationsgesellschaft mbH*, 1988.
- [233] G. Edgar, *Measure, topology, and fractal geometry*. Springer Science & Business Media, 2007.
- [234] C. Grana, G. Pellacani, R. Cucchiara, and S. Seidenari, "A new algorithm for border description of polarized light surface microscopic images of pigmented skin lesions", *IEEE Transactions on Medical Imaging*, vol. 22, no. 8, pp. 959–964, 2003.
- [235] S. E. Umbaugh, R. H. Moss, and W. V. Stoecker, "Applying artificial intelligence to the identification of variegated coloring in skin tumors", *IEEE engineering in medicine and biology magazine*, vol. 10, no. 4, pp. 57–62, 1991.
- [236] SOTC, *Skin oncology teaching center*, 2007. [Online]. Available: <http://www.dermoncology.com/>.
- [237] R. J. Stanley, W. V. Stoecker, and R. H. Moss, "A relative color approach to color discrimination for malignant melanoma detection in dermoscopy images", *Skin Research and Technology*, vol. 13, no. 1, pp. 62–72, 2007.
- [238] F. Nachbar, W. Stolz, T. Merkle, A. B. Cognetta, T. Vogt, M. Landthaler, P. Bilek, O. Braun-Falco, and G. Plewig, "The abcd rule of dermatoscopy: High prospective value in the diagnosis of doubtful melanocytic skin lesions", *Journal of the American Academy of Dermatology*, vol. 30, no. 4, pp. 551–559, 1994.
- [239] G. Betta, G. Di Leo, G. Fabbrocini, A. Paolillo, and M. Scalvenzi, "Automated application of the "7-point checklist" diagnosis method for skin lesions: Estimation of chromatic and shape parameters.", in *2005 IEEE Instrumentation and Measurement Technology Conference Proceedings*, IEEE, vol. 3, 2005, pp. 1818–1822.
- [240] M. Anantha, R. H. Moss, and W. V. Stoecker, "Detection of pigment network in dermoscopy images using texture analysis", *Computerized Medical Imaging and Graphics*, vol. 28, no. 5, pp. 225–234, 2004.

- [241] E. Vocaturo, D. Perna, and E. Zumpano, "Machine learning techniques for automated melanoma detection", in *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, IEEE, 2019, pp. 2310–2317.
- [242] T. Cover and P. Hart, "Nearest neighbor pattern classification", *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [243] M. Burrioni, R. Corona, G. Dell'Eva, F. Sera, R. Bono, P. Puddu, R. Perotti, F. Nobile, L. Andreassi, and P. Rubegni, "Melanoma computer-aided diagnosis: Reliability and feasibility study", *Clinical cancer research*, vol. 10, no. 6, pp. 1881–1886, 2004.
- [244] B. V. Dasarathy, "Nearest neighbor (nn) norms: Nn pattern classification techniques", *IEEE Computer Society Tutorial*, 1991.
- [245] H. Ganster, P. Pinz, R. Rohrer, E. Wildling, M. Binder, and H. Kittler, "Automated melanoma recognition", *IEEE transactions on medical imaging*, vol. 20, no. 3, pp. 233–239, 2001.
- [246] D. Ruiz, V. Berenguer, A. Soriano, and B. SáNchez, "A decision support system for the diagnosis of melanoma: A comparative approach", *Expert Systems with Applications*, vol. 38, no. 12, pp. 15 217–15 223, 2011.
- [247] S. Dreiseitl, L. Ohno-Machado, H. Kittler, S. Vinterbo, H. Billhardt, and M. Binder, "A comparison of machine learning methods for the diagnosis of pigmented skin lesions", *Journal of biomedical informatics*, vol. 34, no. 1, pp. 28–36, 2001.
- [248] D. Steinberg and P. Colla, "Cart: Classification and regression trees", *The top ten algorithms in data mining*, vol. 9, p. 179, 2009.
- [249] T. Oates and D. D. Jensen, "Large datasets lead to overly complex models: An explanation and a solution.", in *KDD*, 1998, pp. 294–298.
- [250] V. Pugazhenthii, S. Naik, A. Joshi, S. Manerkar, V. Nagvekar, K. Naik, C. Palekar, and K. Sagar, "Skin disease detection and classification", 2019.
- [251] E. Vocaturo, E. Zumpano, and P. Veltri, "Features for melanoma lesions characterization in computer vision systems", in *2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA)*, IEEE, 2018, pp. 1–8.
- [252] L. Caroprese, E. Vocaturo, and E. Zumpano, "Features for melanoma lesions: Extraction and classification", in *IEEE/WIC/ACM International Conference on Web Intelligence-Volume 24800*, ACM, 2019, pp. 238–243.
- [253] A. Kamboj *et al.*, "A color-based approach for melanoma skin cancer detection", in *2018 First International Conference on Secure Cyber Computing and Communication (IC-SCCC)*, IEEE, 2018, pp. 508–513.
- [254] M. A. Arasi, E.-S. M. El-Horbaty, A. El-Sayed, *et al.*, "Classification of dermoscopy images using naïve bayesian and decision tree techniques", in *2018 1st Annual International Conference on Information and Sciences (AiCIS)*, IEEE, 2018, pp. 7–12.
- [255] Y.-T. Chen, R. Dubrow, T. R. Holford, T. Zheng, R. L. Barnhill, and M. Berwick, "Malignant melanoma risk factors by anatomic site: A case-control study and polychotomous logistic regression analysis", *International journal of cancer*, vol. 67, no. 5, pp. 636–643, 1996.

- [256] W. H. Clark Jr, D. E. Elder, D. Guerry IV, L. E. Braitman, B. J. Trock, D. Schultz, M. Synnestvedt, and A. C. Halpern, "Model predicting survival in stage i melanoma based on tumor progression", *JNCI: Journal of the National Cancer Institute*, vol. 81, no. 24, pp. 1893–1904, 1989.
- [257] R. Nie, S.-Q. Yuan, Y.-B. Chen, Y. Wang, S. Chen, S.-M. Li, J. Zhou, G.-M. Chen, T.-Q. Luo, Y.-F. Li, *et al.*, "Robust immunoscore model to predict the response to anti-pd1 therapy in melanoma", *Available at SSRN 3453329*, 2019.
- [258] B. D. Ripley, *Pattern recognition and neural networks*. Cambridge university press, 2007.
- [259] G. S. Vennila, L. P. Suresh, and K. Shunmuganathan, "Dermoscopic image segmentation and classification using machine learning algorithms", in *2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET)*, IEEE, 2012, pp. 1122–1127.
- [260] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118.
- [261] C. Bishop, "Neural networks for pattern recognition: Oxford university press", *New York*, 1996.
- [262] L. Fausett, *Fundamentals of neural networks: architectures, algorithms, and applications*. Prentice-Hall, Inc., 1994.
- [263] M. Wati, N. Puspitasari, E. Budiman, R. Rahim, *et al.*, "First-order feature extraction methods for image texture and melanoma skin cancer detection", in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1230, 2019, p. 012 013.
- [264] M. Hasan, S. D. Barman, S. Islam, and A. W. Reza, "Skin cancer detection using convolutional neural network", in *Proceedings of the 2019 5th International Conference on Computing and Artificial Intelligence*, ACM, 2019, pp. 254–258.
- [265] W. Zhang *et al.*, "Shift-invariant pattern recognition neural network and its optical architecture", in *Proceedings of annual conference of the Japan Society of Applied Physics*, 1988.
- [266] Q. Li, W. Cai, X. Wang, Y. Zhou, D. D. Feng, and M. Chen, "Medical image classification with convolutional neural network", in *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, IEEE, 2014, pp. 844–848.
- [267] M. D. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks", *arXiv preprint arXiv:1301.3557*, 2013.
- [268] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, *et al.*, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [269] H. Liao, "A deep learning approach to universal skin disease classification", *University of Rochester Department of Computer Science, CSC*, 2016.
- [270] F. Ercal, A. Chawla, W. V. Stoecker, H.-C. Lee, and R. H. Moss, "Neural network diagnosis of malignant melanoma from color images", *IEEE Transactions on biomedical engineering*, vol. 41, no. 9, pp. 837–845, 1994.
- [271] F. Erçal, H.-C. Lee, W. V. Stoecker, and R. H. Moss, "Skin cancer classification using hierarchical neural networks and fuzzy systems", 1999.

- [272] A. Menegola, M. Fornaciali, R. Pires, F. V. Bittencourt, S. Avila, and E. Valle, "Knowledge transfer for melanoma screening with deep learning", in *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, IEEE, 2017, pp. 297–300.
- [273] A. R. Lopez, X. Giro-i Nieto, J. Burdick, and O. Marques, "Skin lesion classification from dermoscopic images using deep learning techniques", in *2017 13th IASTED International Conference on Biomedical Engineering (BioMed)*, IEEE, 2017, pp. 49–54.
- [274] J. Kawahara, A. BenTaieb, and G. Hamarneh, "Deep features to classify skin lesions", in *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, IEEE, 2016, pp. 1397–1400.
- [275] E. Ayan and H. M. Ünver, "Data augmentation importance for classification of skin lesions via deep learning", in *2018 Electric Electronics, Computer Science, Biomedical Engineerings Meeting (EBBT)*, IEEE, 2018, pp. 1–4.
- [276] R. Fatima, M. Z. A. Khan, K. Dhruve, *et al.*, "Computer aided multi-parameter extraction system to aid early detection of skin cancer melanoma", *International Journal of Computer Science and Network Security*, vol. 12, no. 10, pp. 74–86, 2012.
- [277] J. Platt *et al.*, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods", *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.
- [278] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, "Support vector clustering", *Journal of machine learning research*, vol. 2, no. Dec, pp. 125–137, 2001.
- [279] M. Q. Khan, A. Hussain, S. U. Rehman, U. Khan, M. Maqsood, K. Mehmood, and M. A. Khan, "Classification of melanoma and nevus in digital images for diagnosis of skin cancer", *IEEE Access*, vol. 7, pp. 90 132–90 144, 2019.
- [280] A. Astorino, A. Fuduli, P. Veltri, and E. Vocaturo, "Melanoma detection by means of multiple instance learning", *Interdisciplinary Sciences: Computational Life Sciences*, vol. 12, no. 1, pp. 24–31, 2020.
- [281] A. Astorino, A. Fuduli, M. Gaudioso, and E. Vocaturo, "Multiple instance learning algorithm for medical image classification", in *Proceedings of the 27th Italian Symposium on Advanced Database Systems, Castiglione della Pescaia (Grosseto), Italy*, 2019.
- [282] G. Zhang, X. Shu, Z. Liang, Y. Liang, S. Chen, and J. Yin, "Multi-instance learning for skin biopsy image features recognition", in *2012 IEEE International Conference on Bioinformatics and Biomedicine*, IEEE, 2012, pp. 1–6.
- [283] E. Vocaturo, E. Zumpano, and P. Veltri, "On the usefulness of pre-processing step in melanoma detection using multiple instance learning", in *International Conference on Flexible Query Answering Systems*, Springer, 2019, pp. 374–382.
- [284] M. Guignard, "Lagrangian relaxation", *Top*, vol. 11, no. 2, pp. 151–200, 2003.
- [285] N. Z. Shor, "Minimization methods for non-differentiable functions", 1985.
- [286] M Gaudioso and M. Monaco, "Variants to the cutting plane approach for convex nondifferentiable optimization", *Optimization*, vol. 25, no. 1, pp. 65–75, 1992.
- [287] A Fuduli and M Gaudioso, "Tuning strategy for the proximity parameter in convex minimization", *Journal of optimization theory and applications*, vol. 130, no. 1, pp. 95–112, 2006.

- [288] A. V. Demyanov, A. Fuduli, and G. Miglionico, "A bundle modification strategy for convex minimization", *European journal of operational research*, vol. 180, no. 1, pp. 38–47, 2007.
- [289] A. Astorino, A. Frangioni, M. Gaudioso, and E. Gorgone, "Piecewise-quadratic approximations in convex numerical optimization", *SIAM Journal on Optimization*, vol. 21, no. 4, pp. 1418–1438, 2011.
- [290] A. Astorino, A. Frangioni, A. Fuduli, and E. Gorgone, "A nonmonotone proximal bundle method with (potentially) continuous step decisions", *SIAM Journal on Optimization*, vol. 23, no. 3, pp. 1784–1809, 2013.
- [291] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization", *Journal of optimization theory and applications*, vol. 109, no. 3, pp. 475–494, 2001.
- [292] A. Astorino and M. Gaudioso, "Polyhedral separability through successive lp", *Journal of Optimization theory and applications*, vol. 112, no. 2, pp. 265–293, 2002.
- [293] E. Claridge, S. Cotton, P. Hall, and M. Moncrieff, "From colour to tissue histology: Physics-based interpretation of images of pigmented skin lesions", *Medical Image Analysis*, vol. 7, no. 4, pp. 489–502, 2003.
- [294] W. E. Roberts, "Skin type classification systems old and new", *Dermatologic clinics*, vol. 27, no. 4, pp. 529–533, 2009.
- [295] C. Barata, M. Ruela, M. Francisco, T. Mendonça, and J. S. Marques, "Two systems for the detection of melanomas in dermoscopy images using texture and color features", *IEEE Systems Journal*, vol. 8, no. 3, pp. 965–979, 2013.
- [296] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection", in *Ijcai*, Montreal, Canada, vol. 14, 1995, pp. 1137–1145.
- [297] A. Astorino, I. Bomze, A. Fuduli, and M. Gaudioso, "Robust spherical separation", *Optimization*, vol. 66, no. 6, pp. 925–938, 2017.
- [298] A. Astorino and A. Fuduli, "The proximal trajectory algorithm in svm cross validation", *IEEE transactions on neural networks and learning systems*, vol. 27, no. 5, pp. 966–977, 2015.
- [299] W. Zhu, N. Zeng, N. Wang, *et al.*, "Sensitivity, specificity, accuracy, associated confidence interval and roc analysis with practical sas implementations", *NESUG proceedings: health care and life sciences*, Baltimore, Maryland, vol. 19, p. 67, 2010.
- [300] D. M. Powers, "Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation", 2011.
- [301] A. Fuduli, P. Veltri, E. Vocaturo, and E. Zumpano, "Melanoma detection using color and texture features in computer vision systems", *Advances in Science, Technology and Engineering Systems Journal*, vol. 4, no. 5, pp. 16–22, 2019. DOI: [10.25046/aj040502](https://doi.org/10.25046/aj040502).
- [302] J. H. A. Silva, B. C. S. de Sá, A. L. R. d. Ávila, G. Landman, and J. A. P. Duprat Neto, "Atypical mole syndrome and dysplastic nevi: identification of populations at risk for developing melanoma - review article", in *Clinics*, vol. 66, pp. 493–499, 2011, ISSN: 1807-5932.
- [303] *Skin cancer foundation*, <https://www.skincancer.org/risk-factors/atypical-moles/>.

- [304] *Melanoma skin cancer*, <http://www.cancer.org/acs/groups/cid/documents/webcontent/003120-pdf.pdf>.
- [305] M. Burrioni, P. Sbano, G. Cevenini, M. Risulo, G. Dell'Eva, P. Barbini, C. Miracco, M. Fimiani, L. Andreassi, and P. Rubegni, "Dysplastic naevus vs. in situ melanoma: Digital dermoscopy analysis", *British Journal of Dermatology*, vol. 152, no. 4, pp. 679–684, 2005. DOI: [10.1111/j.1365-2133.2005.06481.x](https://doi.org/10.1111/j.1365-2133.2005.06481.x). eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1365-2133.2005.06481.x>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2133.2005.06481.x>.
- [306] K. Duffy and D. Grossman, "The dysplastic nevus: From historical perspective to management in the modern era: Part i. historical, histologic, and clinical aspects", *Journal of the American Academy of Dermatology*, vol. 67, no. 1, 1–e1, 2012.
- [307] D. E. Elder, L. I. Goldman, S. C. Goldman, M. H. Greene, and W. H. Clark Jr, "Dysplastic nevus syndrome: A phenotypic association of sporadic cutaneous melanoma", *Cancer*, vol. 46, no. 8, pp. 1787–1794, 1980.
- [308] S. Save, "Dysplastic nevi", *Dermoscopy: Text and Atlas*, p. 447, 2019.
- [309] R. Pampana, A. Kyrgidis, A. Lallas, E. Moscarella, G. Argenziano, and C. Longo, "A meta-analysis of nevus-associated melanoma: Prevalence and practical implications", *Journal of the American Academy of Dermatology*, vol. 77, no. 5, pp. 938–945, 2017.
- [310] M. Arumi-Uria, N. S. McNutt, and B. Finnerty, "Grading of atypia in nevi: Correlation with melanoma risk", *Modern pathology*, vol. 16, no. 8, p. 764, 2003.
- [311] K. K. Reddy, M. J. Farber, J. Bhawan, R. G. Geronemus, and G. S. Rogers, "Atypical (dysplastic) nevi: Outcomes of surgical excision and association with melanoma", *JAMA dermatology*, vol. 149, no. 8, pp. 928–934, 2013.
- [312] E. Rieger, H. P. Soyer, C. Garbe, P. Büttner, R. Kofler, J. Weiss, U. Stocker, S. Krüger, M. Roser, J. Weckbecker, *et al.*, "Overall and site-specific risk of malignant melanoma associated with nevus counts at different body sites: A multicenter case-control study of the german central malignant-melanoma registry", *International journal of cancer*, vol. 62, no. 4, pp. 393–397, 1995.
- [313] S. Gandini, F. Sera, M. S. Cattaruzza, P. Pasquini, O. Picconi, P. Boyle, and C. F. Melchi, "Meta-analysis of risk factors for cutaneous melanoma: Ii. sun exposure", *European journal of cancer*, vol. 41, no. 1, pp. 45–60, 2005.
- [314] M. Xiong, M. Rabkin, M. Piepkorn, R. Barnhill, Z. Argenyi, L. Erickson, J. Guitart, L. Lowe, C. Shea, M. Trotter, *et al.*, "Diameter of dysplastic nevi is a more robust biomarker of increased melanoma risk than degree of histologic dysplasia: A case-control study", *Journal of the American Academy of Dermatology*, vol. 71, no. 6, 2014.
- [315] E. Vocaturo, E. Zumpano, and P. Veltri, "Image pre-processing in computer vision systems for melanoma detection", in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, IEEE, 2018, pp. 2117–2124.
- [316] E. Vocaturo and E. Zumpano, "Dangerousness of dysplastic nevi: A multiple instance learning solution for early diagnosis", in *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, IEEE, 2019, pp. 2318–2323.

- [317] A. Sboner, C. Eccher, E. Blanzieri, P. Bauer, M. Cristofolini, G. Zumiani, and S. Forti, "A multiple classifier system for early melanoma diagnosis", *Artificial intelligence in medicine*, vol. 27, no. 1, pp. 29–44, 2003.
- [318] M. M. Rahman, P. Bhattacharya, and B. C. Desai, "A multiple expert-based melanoma recognition system for dermoscopic images of pigmented skin lesions", in *2008 8th IEEE International Conference on BioInformatics and BioEngineering*, IEEE, 2008, pp. 1–6.
- [319] K. Przystalski, L. Nowak, M. Ogorzałek, and G. Surówka, "Decision support system for skin cancer diagnosis", 2010.
- [320] M. Takruri, M. W. Rashad, and H. Attia, "Multi-classifier decision fusion for enhancing melanoma recognition accuracy", in *2016 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA)*, IEEE, 2016, pp. 1–5.
- [321] E. Vocaturo, E. Zumpano, and P. Veltri, "On discovering relevant features for tongue colored image analysis", in *Proceedings of the 23rd International Database Applications & Engineering Symposium*, ACM, 2019, p. 12.
- [322] E. Zumpano, P. Iaquinta, L. Caroprese, G. L. Cascini, F. Dattola, P. Franco, M. Iusi, P. Veltri, and E. Vocaturo, "SIMPATICO 3d: A medical information system for diagnostic procedures", in *IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2018, Madrid, Spain, December 3-6, 2018*, 2018, pp. 2125–2128.
- [323] *Market data forecast*, [Online; last consultation 26- Aprile-2020], 2020. [Online]. Available: <https://www.marketdataforecast.com>.
- [324] *Octopus ventures*, [Online; last consultation 26- Aprile-2020], 2020. [Online]. Available: <https://octopusventures.com>.
- [325] A Bharathi and A. Natarajan, "Cancer classification using modified extreme learning machine based on anova features", *European Journal of Scientific Research*, vol. 58, no. 2, pp. 156–165, 2011.
- [326] N. Japkowicz *et al.*, "Learning from imbalanced data sets: A comparison of various strategies", in *AAAI workshop on learning from imbalanced data sets*, Menlo Park, CA, vol. 68, 2000, pp. 10–15.
- [327] R. C. Prati, G. E. Batista, and M. C. Monard, "Class imbalances versus class overlapping: An analysis of a learning system behavior", in *Mexican international conference on artificial intelligence*, Springer, 2004, pp. 312–321.
- [328] D. Nie, "Classification of melanoma and clark nevus skin lesions based on medical image processing techniques", in *2011 3rd International Conference on Computer Research and Development*, IEEE, vol. 3, 2011, pp. 31–34.
- [329] M. Schumacher, N. Holländer, and W. Sauerbrei, "Resampling and cross-validation techniques: A tool to reduce bias caused by model building?", *Statistics in medicine*, vol. 16, no. 24, pp. 2813–2827, 1997.
- [330] R. W. Johnson, "An introduction to the bootstrap", *Teaching Statistics*, vol. 23, no. 2, pp. 49–54, 2001.
- [331] B. Efron, "The estimation of prediction error: Covariance penalties and cross validation", *Journal of the American Statistical Association*, vol. 99, no. 467, pp. 619–632, 2004.