Università degli Studi di Bologna
Facoltà di Scienze Matematiche, Fisiche e Naturali
Dottorato di Ricerca in Fisica
XVII Ciclo

*Ph.D. Thesis*

# Optimal Image Representations For Mass Detection In Digital Mammography

*by*

**Matteo Masotti**

Advisor:
Prof. Renato Campanini

Coordinator:
Prof. Roberto Soldati

*March, 2005*

Università degli Studi di Bologna
Facoltà di Scienze Matematiche, Fisiche e Naturali
Dottorato di Ricerca in Fisica
XVII Ciclo

*Ph.D. Thesis*

# Optimal Image Representations For Mass Detection In Digital Mammography

*by*

**Matteo Masotti**

**Keywords:** *Ranklets, Wavelets, Steerable Filters, Support Vector Machine, Recursive Feature Elimination, Image Processing, Pattern Recognition, Computer–Aided Detection, Digital Mammography*

Advisor:
Prof. Renato Campanini

Coordinator:
Prof. Roberto Soldati

*March, 2005*

*One step closer to knowing…*

# Acknowledgments

First of all, I would like to express my deepest gratitude to Prof. Renato Campanini for providing valuable comments and guidance over the years. He gave me the possibility to conduct research with the maximal freedom.

I would also like to express my sincerest appreciation to Prof. Enzo Gandolfi for introducing me to the world of wavelets and multi–resolution analysis with many extended discussions during the development of my M.Sc. thesis. Without him, I probably would not even have known this exciting world.

Many thanks also to everyone in the Medical Imaging Group for providing such a stimulating research atmosphere during the day and friendly hours nighttime at the pub.

Finally, many thanks to my family.

# Abstract

This work addresses a two–class classification problem related to one of the leading cause of death among women worldwide, namely breast cancer. The two classes to separate are tumoral masses and normal breast tissue.

The proposed approach does not rely on any feature extraction step aimed at finding few measurable quantities characterizing masses. On the contrary, the mammographic regions of interest are passed to the classifier—a *Support Vector Machine* (SVM)—in their raw form, for instance as vectors of gray–level values. In this sense, the approach adopted is a *featureless* approach, since no feature is extracted from the region of interest, but its image representation embodies itself all the features to classify.

In order to find the *optimal image representation*, several ones are evaluated by means of Receiver Operating Characteristic (ROC) curve analysis. Image representations explored include *pixel–based*, *wavelet–based*, *steer–based* and *ranklet–based* ones. In particular, results demonstrate that the best classification performances are achieved by the ranklet–based image representation. Due to its good results, its performances are further explored by applying SVM *Recursive Feature Elimination* (SVM–RFE), namely recursively eliminating some of the less discriminant ranklets coefficients according to the cost function of SVM. Experiments show good classification performances even after a significant reduction of the number of ranklet coefficients.

Finally, the ranklet–based and wavelet–based image representations are practically applied to a real–time working *Computer–Aided Detection* (CAD) system developed by our group for tumoral mass detection. The classification performances achieved by the proposed algorithm are interesting, with a false–positive rate of 0.5 marks per–image and 77% of cancers marked per–case.

# Contents

# List of Figures

# List of Tables

# Introduction

## Overview And Motivation

Suppose you are willing to find a bunch of influent features—or properties—which are well suited to separate—let say—a group of men from a group of women. It is evident that the height and weight of those people should prove more discriminant than—for example—the length and thickness of their hairs. The evidence with which this can be affirmed is mainly due the the simplicity of the problem. However, finding features which characterize well the classes under study is not always as straightforward and effective, but could sometimes require much deeper investigations. This is the case of the two–class classification problem faced here.

In this work, the demanding problem to address is concerned with one of the major cause of death among women, namely breast cancer. The two classes to separate are tumoral masses—thickenings of the breast tissue with size ranging from 3 mm to 30 mm—and normal breast tissue. To this aim, each X–ray image of the breast under study—namely mammogram—is scanned at all the possible locations with the passage of a window. A corresponding crop of the mammographic image is therefore extracted. Successively, each extracted crop is classified by means of a Support Vector Machine (SVM) classifier as belonging to the class of tumoral masses or to the class of normal breast tissue. Differently from the most part of the algorithms dealing with this problem, the novel approach adopted herein does not rely on any feature extraction step aimed at finding few measurable quantities which characterize tumoral masses and differentiate them from normal breast tissue. On the contrary, it is rather a *featureless* approach in which crops are classified as they are, namely as vectors of gray–level values.

In order to explore the possibility of improving the classification performances, several image representations of the crops are evaluated by means of ROC curve analysis. Starting from the simplest one—namely the *pixel–based* image representation—other image representations are tested, such as the *wavelet–based* and the *steer–based* one, respectively based on the transformation of the crops by means of the multi–resolution wavelet transform and the multi–resolution steerable pyramid. In these last two cases, evidently, the features classified by SVM are represented by the coefficients obtained applying those transforms to the mammographic crops under study.

A further and novel image representation is then developed and optimized specifically for this problem. This image representation—referred to as *ranklet–based*—is based on the application of a new rank–based technique introduced for the first time in 2002 for face detection problems. In particular, this very promising technique—never been applied to imaging problems different from face detection, such as for instance medical imaging—is known as ranklet transform. The performances of this ranklet–based image representation are also explored by means of a technique known as SVM Recursive Feature Elimination (SVM–RFE), namely recursively eliminating some of the less discriminant ranklets coefficients according to the cost function of SVM.

Experiments show that the ranklet–based image representation achieves the best classification results. Good results are achieved as well by the pixel–based and wavelet–based image representations, the latter used in its overcomplete—or redundant—version. The tests performed by using SVM–RFE on the ranklet–based image representation show that—with this method—the number of ranklet coefficients presented to SVM can be sensibly reduced without affecting the classification performances. Furthermore, an accurate analysis of the most discriminant ranklet coefficients gives interesting suggestions about which coefficients are important for classification purposes.

Finally, the ranklet–based and the wavelet–based image representations previously tested are implemented into a real–time working Computer–Aided Detection (CAD) system, specifically designed for mass detection. The motivation for choosing those image representations is twofold. First, they prove to obtain excellent classification performances. Second, their computational times are definitely acceptable. Here, the two image representations are used as independent detectors whose opinions on each crop—mass or normal breast tissue—are combined by performing a logical AND. This strategy results particularly effective and allows the proposed CAD system to achieve good performances.

Notice, finally, that the experimental part of this work—namely Section 4 and Section 5—together with some of the theoretical details discussed in Section 3.4 concerning the ranklet transform, walk mainly along the road traced by some recent work developed by our group. In particular, the evaluation of the pixel–based and wavelet–based image representations is mainly discussed in (Angelini *et al.*, 2004), some theoretical aspects about the ranklet transform together with its evaluation are discussed in (Masotti, 2004), whereas the application of SVM–RFE to the ranklet coefficients is described in (Masotti, 2005). Furthermore, an earlier version of the real–time working CAD system—where the ranklet–based and wavelet–based were not yet implemented—is described in (Campanini *et al.*, 2002, 2004c,a).

## Thesis Contributions

The contributions of this work are mainly related to the featureless approach adopted, to the evaluation of several image representations, to the classifier used, to the development and optimization of a novel image representation based on the ranklet transform, to the application of SVM–RFE to the ranklet coefficients and, finally, to the implementation of the best image representations into a real–time working CAD system. In the following they will be discussed in just a little more detail.

**Featureless approach** Due to the great variety of tumoral masses, it is extremely difficult to get a common set of few measurable quantities effective for every kind of masses. In order to virtually detect all kind of masses—thus avoiding to concentrate only to a restricted family for which those quantities are easily individuated—the featureless approach previously described is adopted. Except for some of our past works (Campanini *et al.*, 2002, 2004c,a), this approach has never been used in mammographic mass detection applications. However, a similar technique has been proposed for face, people and car detection in some works developed at the MIT Artificial Intelligence Laboratory, namely (Papageorgiou, 1997; Oren *et al.*, 1997; Papageorgiou *et al.*, 1998a,b; Papageorgiou & Poggio, 1999a,b).

**Image representations evaluation** In order to find the optimal image representation—namely that achieving the best classification performances—several tests are performed. In particular—as previously discussed—the pixel–based, the wavelet–based, the steer–based and, finally, the ranklet–based image representation are evaluated. Except for the steer–based image representation—which has been evaluated similarly in (Sajda *et al.*, 2002)—all these image representations represent a novel approach to mammographic mass detection.

**Classifier** Traditional classification techniques—such as the multilayer perceptrons—use empirical risk minimization and only guarantee minimum error over the training set. These techniques can result in overfitting of the training data and therefore poor generalization performance. In this work, a relatively new pattern classification technique is used—SVM—that has recently received a great deal of attention in the literature. The number of applications of SVM is still quite small, thus the presentation of SVM as the core learning machine represents a significant advancement of the technique in the context of practical applications.

**Ranklet–based image representation and SVM–RFE** With the fundamental aim of improving the classification performances, a novel image representation—namely ranklet–based image representation—is evaluated. Moreover—due to its good classification results—its performances are explored by means of SVM–RFE, namely recursively eliminating some of the less discriminant ranklet coefficients according to the cost function of an SVM. Except for some applications mainly focused on faced detection—namely (Smeraldi, 2002, 2003a; Smeraldi & Rob, 2003; Smeraldi, 2003b)—ranklets have never been applied to image classification. For this reason, their application to medical imaging—and in particular to mammographic mass detection—together with their submission to SVM–RFE, represents a significant contribution of this work, probably the most innovative.

**Real–time working CAD system** This work presents a practical application of the best image representations found—namely the ranklet–based and the wavelet–based—into a real–time working CAD system developed by our group and specifically suited for mass detection. This system is currently deployed at three hospitals worldwide in its prototype version.

# Outline

The rest of this work is organized as follows.

**Chapter 1** In Chapter 1, an overview of digital mammography will be given, together with some introductory elements concerning the most common breast abnormalities, such as tumoral masses and micro–calcifications. Some details about screening mammography and CAD systems will be also outlined.

**Chapter 2** In Chapter 2, some basic knowledge concerning machine learning will be provided, with particular emphasis on the classification techniques used in this work, namely SVM and SVM–RFE.

**Chapter 3** Chapter 3 is intended to provide the reader with a clearer picture of the imaging techniques that will be adopted in this work. To this aim, it will discuss in great detail the pixel–based image representation of an image, together with the multi–resolution wavelet transform, the multi–resolution steerable pyramid and the multi–resolution ranklet transform.

**Chapter 4** The approaches adopted in order to evaluate the different image representations and their classification results are discussed and presented in Chapter 4. Here—in particular—the performances achieved by the pixel–based, wavelet–based, steer–based and ranklet–based image representations will be discussed in detail. Furthermore, the application of SVM–RFE to the ranklet coefficients will be described and some discussion about the most influent ranklet coefficients for classification purposes will be given.

**Chapter 5** In Chapter 5, a practical implementation into a real–time working CAD system of the ranklet–based and wavelet–based image representations evaluated is described in detail. Tests are performed and the very good classification performances achieved are presented and discussed.

Chapter 1

# Digital Mammography

*In this Chapter, an introduction to digital mammography will be given. Several aspects will be treated, starting from some simple considerations concerning the breast anatomy, passing by a description of the most common breast abnormalities and finishing with some considerations about the screening techniques adopted to face breast cancer incidence, for instance Computer–Aided Detection (CAD). In particular, in Section 1.1 breast cancer will be examined and a detailed discussion of its—tremendous—incidence statistics expected for year 2005 will be given. Section 1.2 will present a very simple description of the gross anatomy of the breast, with particular attention to the regions where usually breast cancer forms, namely the so–called Terminal Duct Lobular Units (TDLU). In Section 1.3, the most important breast abnormalities will be analyzed. In particular, tumoral masses will be considered, together with calcifications and other less common abnormalities, such as dilated lactiferous ducts, areas of asymmetry or architectural distortion and, finally, thickening or retraction of the skin. Notice, specifically, that in this work the attention will be mainly devoted to tumoral masses, namely thickenings of the breast tissue with size ranging from 3 mm to 30 mm. The importance of screening mammography for an early detection of breast cancer will be discussed in Section 1.4. Here, the correspondence between an earlier detection of breast cancer and higher likelihood that treatments will be successful—namely higher chances of survival for the patient—will be also stressed. Finally, in Section 1.5 CAD systems for the automatic detection of lesions in X–ray breast images will be introduced.*

## 1.1 Breast Cancer

*Cancer* develops when cells in a specific part of the body begin to grow out of control. Typically, normal body cells grow, divide and die in an orderly fashion. In particular, during the early years of an individual's life, normal cells divide more rapidly until the individual becomes an adult. After that, cells in most parts of the body divide only to replace worn–out or dying cells and to repair injuries. Differently, cancer cells grow and divide indefinitely. Instead of dying, in fact, they outlive normal cells and continue to form new abnormal cells.

Cancer usually forms as a *tumor*, namely as an abnormal growth of tissue. The most dangerous form of tumor is the *malignant tumor* which is comprised of cancer cells that invade neighboring tissues, where they begin to grow and replace normal tissue. This process is called *metastasis*. Regardless of where a cancer may spread, it is always named for the place it began. For instance, breast cancer that spreads to the liver is still called breast cancer, not liver cancer. Fortunately, not all tumors are malignant, or cancerous. *Benign tumors* do not invade neighboring tissues and do not seed metastases, but may locally grow to great size. They usually do not return after surgical removal and—with very rare exceptions—are not life threatening.

*Breast cancer* is a malignant tumor that has developed from cells of the breast. It represents the second leading cause of cancer death in women, exceeded only by lung cancer. Although lung cancer has a lower incidence than breast cancer, in fact, more women die each year of the former. Nevertheless, the majority of deaths from lung cancer can be attributed to smoking, thus breast cancer continues to be the leading cause of *non–preventable* cancer death. According to the World Health Organization, more than 1.3 million people will be diagnosed with breast cancer in 2005 worldwide. In (American Cancer Society, 2005) the American Cancer Society estimates that in 2005 approximately 211240 women in the United States will be diagnosed invasive breast cancer. Another 58490 women will diagnosed with in situ breast cancer, a very early form of the disease. Though much less common, breast cancer also occurs in men. An estimated 1690 cases will be diagnosed in men in 2005. It is also estimated that 40410 women and 460 men will die from breast cancer in the United States this year. Although those estimations are tremendously high, it must be noticed that mortality rates declined by 2.3% per year from 1990 to 2001 in all women, with larger decreases in younger—namely less than 50 years old—women. Medical experts attribute this decline to increased awareness, earlier detection through screening and more effective treatments.

As regards the geographical distribution of breast cancer, this disease has a higher incidence in Europe—especially Western Europe—and North America. In the Far East and parts of Africa, the mortality rate due to breast cancer is much lower, with an incidence about 5 times smaller than in the West. This is mainly due to the lack of information on this subject from under–developed and developing countries. Nevertheless, there has been a substantial increase in the number of new cases. During the last few years, Japan has witnessed a growth of 10 times in the number of breast cancers. In the Western world, recent results show that breast cancer accounts for a high percentage of the overall cancer incidence in women, approximately 32% of all cancer cases. According to (Ferlay *et al.*, 2004), around the world there are approximately 1151000 new cases of breast cancer every year, of which the more developed regions account for approximately 636000 and the European Community for approximately 261000. Amongst the developed countries, Italy is rated as one of the regions with the highest incidence in breast cancer, since approximately 36000 new cases occur and 11000 women die each year.

Researchers have tried to trace both environmental and genetic causes that lead to developing the disease. Still, there is so far insufficient evidence to support theories that attribute unhealthy food, alcohol, genetic mutations, pollution—and others—as major factors in the expansion of the disease. Many attribute that 70% of cancers have their origins in the foods eaten. As discussed in (American Cancer Society, 2005), obesity has been found to be a breast cancer risk in all studies, especially for women after menopause. Although ovaries produce most of estrogen, fat tissue produces a small amount of estrogen. Having more fat tissue can increase the estrogen levels, thus increase the likelihood of developing breast cancer. Most studies found also that breast cancer is less common in countries where the typical diet is low in total fat, low in polyunsaturated fat and low in saturated fat. Alcohol is linked as well to a slightly increased risk of developing breast cancer. Compared with nondrinkers, women who consume one alcoholic drink a day have a very small increase in risk, whereas those who have 2 to 5 drinks daily have about one and a half times the risk of women who drink no alcohol. Alcohol is also known to increase the risk of developing cancers of the mouth, throat and esophagus. On the other hand, recent studies have shown that about 5% to 10% of breast cancer cases are hereditary as a result of gene mutations. The most common gene changes are those of the BRCA1 and BRCA2 genes. Normally, these genes help to prevent cancer by making proteins that keep cells from growing abnormally. Finally, other well–established factors refer to pollution, family history, ethnic background, absence of childbirth and others.

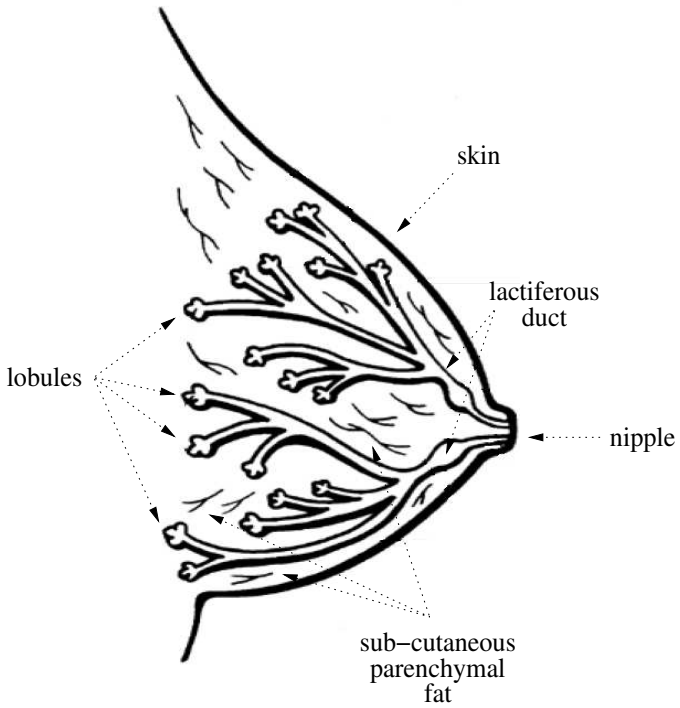



**Figure 1.1:** Gross anatomy of normal breast. Shown are the lobules, the lactiferous ducts, the nipple, the sub–cutaneous and inter–parenchymal fat and the skin.

## 1.2 Brief Anatomy Of The Breast

For a better understanding of the subject, an overview of the breast anatomy becomes necessary. The *breast*—or mammary gland—is a modified sweat gland with the specific function of milk production. The development of the breast begins in the embryo in the $5^{th}$ week with the formation of the primitive milk streak from the axilla to the groin. Once adult, the breast lies on the pectoralis major muscle, which crosses the chest obliquely. It is composed of three basic structures, namely the *skin*, the *sub–cutaneous and parenchymal fat* and, finally, the inner breast tissue including the *lobules*, see Fig. 1.1. In particular, this inner part is contained by superficial and deep fascial layers drained by *lactiferous ducts* which converge beneath the *nipple* and empty as five to eight collecting ducts on its surface.

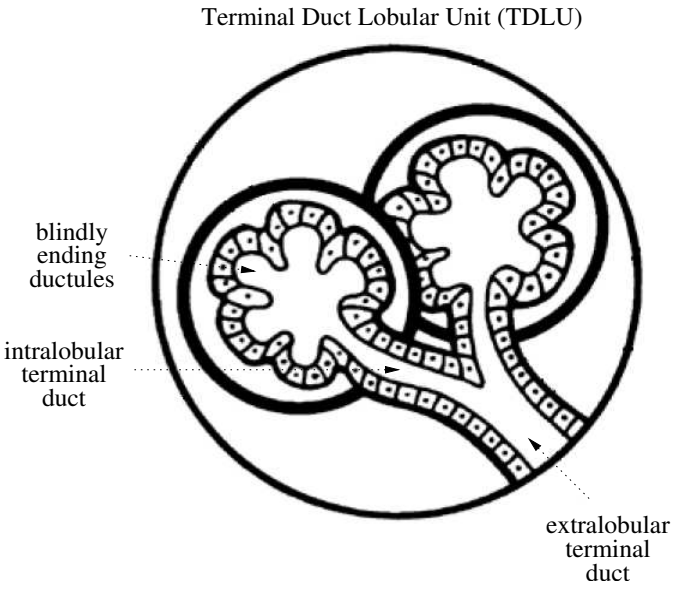



**Figure 1.2:** Basic histopathologic unit of the breast, namely Terminal Duct Lobular Unit (TDLU). Shown are the inner branches of the lactiferous ducts, namely the extralobular terminal duct, the intralobular terminal duct and the blindly ending ductules.

The micro–anatomy of the breast has been mainly described by Wellings in (Wellings *et al.*, 1975). He identified the basic histopathologic unit of the breast as the *Terminal Duct Lobular Unit* (TDLU). This basic unit is composed of the inner branches of the lactiferous ducts, namely the *extralobular terminal duct*, the *intralobular terminal duct* and the *blindly ending ductules*, see Fig. 1.2. The TDLU is important physiologically because it is the site of milk production. It is also the site of development of most benign and malignant breast lesions. An understanding of this anatomic structure is thus important in the correlation of mammographic and pathologic findings. The majority of ductal carcinomas are thought to develop in the terminal duct branches and the calcifications associated with these lesions tend to have a linear, branching orientation or configuration, corresponding to the duct lumen. Lobular processes often are benign, and include many forms of fibrocystic changes, namely adenosis, sclerosing adenosis, cystic hyperplasia. Associated calcifications are more smoothly marginated and rounded, conforming to the configuration of the ductules.

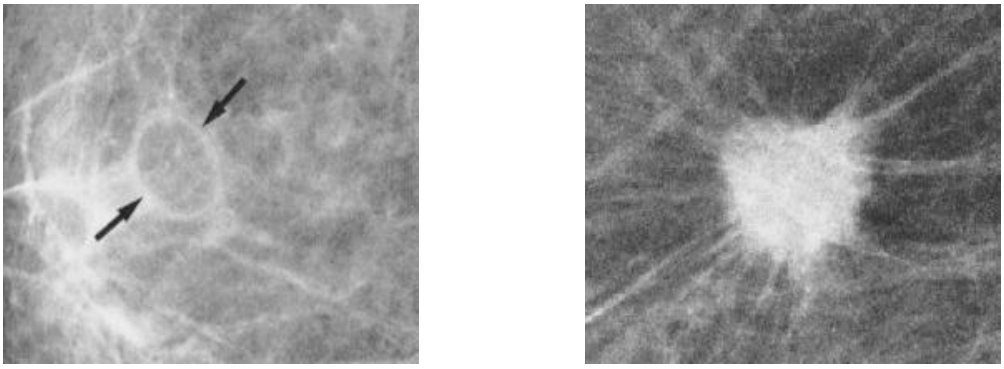


**Figure 1.3:** Tumoral masses. Well—circumscribed mass (left). Spiculated mass (right).

## 1.3 Types Of Breast Abnormalities

The lesions which may be detected on X–ray breast images are related to the local manifestations of an ever–increasing number of neoplastic cells. Abnormalities that may be seen include masses, calcifications, areas with asymmetric densities or architectural distortions, prominent lactiferous ducts, skin or nipple thickening and/or retraction, see (Shaw de Paredes, 1993).

The assessment of these abnormalities is critical in tailoring further evaluation of the lesion and its management. The X–ray images of the two breasts—namely *mammograms*—are placed as mirror images to allow for careful comparison of the tissue patterns and to assess for symmetry. Correlation with the patient's history and findings at clinical examination are important in determining the role of other procedures and in further defining the feasible causes of the particular abnormality identified. In order to stress the importance of recognizing these abnormalities, some of the most common will be discussed in the following.

### 1.3.1 Masses

Evaluating a tumoral *mass*—namely a thickening of the breast tissue with size ranging from 3 to 30 mm—involves assessment of the shape and margination of the lesion. It involves also assessment of the presence of a fatty halo together with the presence of other associated findings. In particular—as described in (Sickles, 1989)—a mass is for instance classified with respect to its margination, namely according to its being relatively well *circumscribed* or *spiculated*, see Fig. 1.3.

Benign lesions tend to be well circumscribed with a fatty halo surrounding the margin and are of medium to low density. Well–circumscribed masses may be, in fact, classified into four groups, namely radiolucent, mixed fat and soft tissue density, medium density and high density. Although a good indication of benignancy, the halo sign is not infallible. In fact, even though well–circumscribed masses are most likely benign, some cancers also may be well defined. In particular, as discussed in (Marsteller & Shaw de Paredes, 1989), the majority of breast cancers that appear as relatively well–defined lesions are *infiltrating ductal carcinomas*.

A poorly defined or spiculated lesion is more likely to be malignant than is a well–circumscribed mass. If no appropriate history suggests that a spiculated lesion may be benign, biopsy may be necessary to confirm or exclude malignancy. The classic appearance of a primary infiltrating breast carcinoma is, in fact, that of a spiculated mass. *Infiltrating ductal carcinomas*—which account for $70-80\%$ of breast malignancies—tend to present spiculated lesions with or without associated micro–calcifications. The center of the lesion is of medium to high density in comparison with the surrounding tissue and fine tendrils surround the tumor mass. This appearance is produced by the infiltration of the tumor into the breast and by the desmoplastic reaction associated with these lesions. Occasionally, the tumor will diffusely infiltrate the breast without producing a central tumor mass and the mammographic findings are much more subtle. On the other hand, *infiltrating lobular carcinoma* accounts for $3-4\%$ of breast malignancies and is characterized at pathologic examination by a linear arrangement of tumor cells throughout the tissue. Bi–laterality and multi–centricity are more frequently associated with infiltrating lobular rather than infiltrating ductal carcinoma. There is a tendency for infiltrating lobular carcinoma to grow in a diffuse manner, manifesting itself as a poorly defined opacity or an architectural distortion.

### 1.3.2 Calcifications

*Calcifications* of some type are identified on the majority of mammograms. Many of those calcifications are clearly benign and need no further evaluation. Their analysis is based on assessment from prior mammograms of morphologic features, size, distribution, location, variability and stability. Features of breast calcifications suggesting a benign cause are their being macro–calcifications, their having smooth margination or their being diffusely scattered micro–calcifications in both breasts. Features suggesting a malignant process are focal clustered micro–calcifications of variable size and shape with irregular margination, see Fig. 1.4.

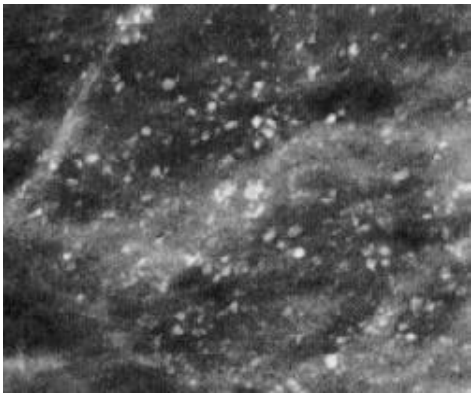


**Figure 1.4:** Micro–calcifications.

Since they can represent a potential malignant process, an understanding of the anatomy and histopathologic features is important in correlating mammographic findings of micro–calcifications and what they may potentially represent. For instance, they may be divided into two types based on their site of origin. *Lobular calcifications* occur in the terminal ductules within the lobule and have a smooth rounded configuration conforming to the internal contour of the ductule. These calcifications tend to be similar in size, shape and density. Lobular–type micro–calcifications scattered in both breasts usually are benign and can be followed up with mammography. In particular, they tend to be focal and may be located at pathologic examination within the lobules containing lobular carcinoma in situ or in adjacent benign lobules. On the other hand, *ductal micro–calcifications* often form in the terminal ducts and are considered a more suspect mammographic finding than are lobular micro–calcifications. The mammographic appearance of ductal calcifications is of small irregular mixed–morphologic calcifications that may be oriented in a linear arrangement. They can occur in benign conditions, including ductal hyperplasia and atypical ductal hyperplasia. However, these calcifications are generally not highly irregular or branching. When ductal micro–calcifications are present, intraductal carcinoma must be considered. Malignant ductal calcifications usually are small, namely $0.1 - 0.3$ mm in diameter. Combinations of forms, including rod–shaped, punctate, comma–shaped, branching and lacy calcifications, may occur together and are highly suspicious for carcinoma. The greater the number of calcifications within an area, the more suspicion (Wolfe, 1977). Malignant calcifications tend to occur in tight clusters of 1 cm diameter or less.

### 1.3.3 Others

Other less common mammographic signs of carcinoma include dilated lactiferous ducts, focal areas of asymmetry or architectural distortion and, finally, thickening or retraction of the skin. As with the signs already discussed, history and clinical examination play important roles in the decision of how to manage these abnormalities. The lactiferous ducts may be dilated, however, if the dilated ducts are quite asymmetric or if a solitary dilated duct is present, ductal malignancy must be considered. Focally asymmetric glandular tissue commonly is seen at mammography and is benign. However, if asymmetry is present and has a similar shape on two views, is highly opaque, is associated with other findings, or is palpable, biopsy must be considered. Architectural distortion usually is seen as a disturbance of the normal orientation of tissue to the nipple. A secondary sign of breast cancer is focal skin thickening or retraction. The tumor may extend along the Cooper ligaments, retracting them and therefore also retracting the skin. The tumor may also extend directly to the skin, causing skin thickening or ulceration.

## 1.4 Screening Mammography

As already anticipated, X–ray imaging is the technique commonly used for breast cancer investigation. In particular, *screening mammography* is the periodical low–dose X–ray examination of the breast that is performed on women with no complaints or symptoms of breast cancer—in other words—asymptomatic. The main objective is to detect breast cancer when it is still too small to be palpated by a physician or by the patient herself by means of self–breast examinations. If a radiographic image of the breast presents any features that seem suspicious to the radiologist, the patient will be asked to attend an assessment clinic where more investigations are performed by means of medical imaging and consulting.

Early detection is crucial in screening mammography. In fact, breast cancers that are detected due to their symptoms tend to be larger and are more likely to have spread beyond the breast. In contrast, breast cancers found during screening examinations are more likely to be small and still confined to the breast. This is the reason why finding a breast cancer as early as possible improves the likelihood that treatments will be successful. At the same time, this represents the reason why the size of a breast cancer—and how far it has spread—are the most important factors in predicting the prognosis—namely the outlook for chances of survival—of a woman with this disease.

It is worth noticing that detecting breast cancer in its earlier stage is not as straightforward, even though screening programmes possess several advanced techniques available. The main reason is that the signs of breast cancer that appear in X–ray mammograms present a significant challenge to radiologists and they are generally difficult to distinguish in the highly textured breast anatomy. Nevertheless, from the first trials in USA and Canada in the sixties and its very first implementation in the seventies in Sweden, screening programmes proved to be fundamental, reducing the mortality caused by breast cancer in women by nearly 30%, as demonstrated in (Thurfjell & Lindgren, 1996). Furthermore, according to optimistic statements, screening mammography almost doubles the chances of survival in women that develop breast cancer.

## 1.5 Computer–Aided Detection

Although screening mammography is considered the most effective method for early detection of breast cancers, it is well known that radiologists may miss 15−30% of breast lesions. Missed detections may be due to the subtle nature of the radiographic findings, poor image quality, eye fatigue or oversight by the radiologists. In order to face this problem, it has been suggested that double reading—by two radiologists—may increase sensitivity, see (Thurfjell *et al.*, 1994).

The aim of *Computer–Aided Detection* (CAD) systems is to increase the efficiency and effectiveness of screening procedures by using a computer system as a second reader. The main idea is to indicate locations of suspicious abnormalities in mammograms as an aid to the radiologist, but leaving the final decision regarding the likelihood of the presence of a cancer to him. The computer output indicating the potential sites of lesions may be useful to assist the radiologists in interpreting mammograms, especially in mass screening, where the majority of cases are normal and only a small fraction are breast cancers. In particular, it has been reported that 30 − 50% of breast carcinomas detected mammographically demonstrate clustered micro–calcifications, with about 80% of breast carcinomas revealing micro–calcifications upon microscopic examination, see (Sickles, 1982; Murphy & DeSchryver-Kecskemeti, 1978). Additionally, studies indicate that 26% of non–palpable cancers present mammographically as masses while 18% present as a mass with micro–calcifications, see (Sickles, 1986). With the above statistics in mind, most computerized schemes are being developed for the detection of both mass lesions and clustered micro–calcifications.

There is a strong evidence of the potential benefit of CAD in the detection and characterization of some lesions in mammography. However, it is important to be cautious about potential pitfalls associated with the use of the computer output. Advances in science and technology can bring many benefits, but also can be harmful if not used properly. Potential pitfalls of CAD can occur for all the four possible outcomes of the automatic detection, namely, false positives and false negatives and even with true positives and true negatives. As it will be discussed in much more detail in Section 2.1.4, false positives represent non–lesions classified as lesions, false negatives represent lesions classified as non–lesions, true positives represent lesions classified as lesions and, finally, true negatives represent non–lesions classified as non–lesions. In particular, there has been a general concern that false positives determined by computer may increase the number of unnecessary biopsies in mammographic screening. However, because many computer false positives are different from radiologists false positives, it is unlikely to produce a large increase in biopsy and call–back rate. In fact, two studies of CAD in mammography—namely (Nishikawa *et al.*, 1995; Roehrig *et al.*, 1998)—produced similar outcomes. That is, there was no increase in the call–back rate for additional examinations when CAD was implemented. On the other hand, if radiologists are strongly influenced by the computer output—and/or for some other reasons in determining a threshold level on decision–making such as biopsy versus non–biopsy—there will be a danger of unnecessary biopsies. False negatives identified by computer can cause a problem as well, in particular with missing obvious and detectable lesions if the computer output is trusted excessively and if radiologists curtail their usual effort in searching for lesions. Therefore, radiologists expertise and conscientious efforts will remain critically needed. At the same time, radiologists may face difficult situations when they disagree with true positives and true negatives obtained by computer, even when the final decisions are made conscientiously with their best effort.

Computerized schemes for CAD generally include three basic components which are based on three different technologies. The first component is *image processing* for enhancement and extraction of lesions. It is important to notice that the image processing involved in CAD schemes is aimed at facilitating the computer—rather than the human observer—to pick up the initial candidates of lesions and suspicious patterns. Various image–processing techniques have been employed for different types of lesions. Some of the most commonly used techniques include filtering based on Fourier analysis, wavelet transform, morphological filtering and so forth.

The second component is the *extraction of image features* such as size, contrast and shape of the candidates selected in the first step. It is possible to define numerous features based on some mathematical formula that may not be easily understood by the human observer. However, it is generally useful to define—at least at the initial phase of CAD development—image features that have already been recognized and described subjectively by radiologists. The reason is that radiologists' knowledge is based on their observations of numerous cases over the years and their diagnostic accuracy is generally very high and reliable. In this sense, one of the most important factors in the development of CAD schemes is to find unique features that can distinguish reliably between a lesion and normal anatomic structures.

Finally, the third component is *data processing* for distinction between normal and abnormal patterns, based on the features obtained from the second step. A simple and common approach employed in this step is a rule–based method, which may be established based on the understanding of lesions and other normal patterns. Therefore, it is important to note that the rule–based method may provide useful information for improving the CAD schemes. Other techniques used include discriminant analysis, Artificial Neural Networks (ANNs) or decision–tree methods.

Chapter 2

# Pattern Classification

*An overview of pattern classification will be given in this Chapter, with particular emphasis on a specific classifier—known as Support Vector Machine (SVM)—which will be used intensively in the rest of this work. In Section 2.1, some introductory notions about learning machines, together with some remarks on how to validate and successively present their classification performance, will be given. Section 2.2 will introduce some fundamental concepts of statistical learning theory, a mathematical formulation developed by V. Vapnik which describes the statistical aspects of automated learning. Furthermore, it will discuss the mathematical details of SVM. Finally, in Section 2.3, a recent feature reduction technique—known as Recursive Feature Elimination (RFE)—will be presented and its implementation by using SVM will be explained in detail.*

## 2.1 Machine Learning

### 2.1.1 The act of learning

In humans the act of learning is namely the process of gaining knowledge or skill in something by experience. Common and apparently simple human processes as recognizing a landscape, understanding spoken words, reading handwritten characters or identifying an object by touching it, they all belie the act of learning. In fact, the condition for a landscape to be recognized, spoken words to be understood, handwritten characters to be read and objects to be identified, is that the human brain has been previously trained in order to do that, namely it has *learnt*

how to do that. This is why it is necessary to admire a landscape several times before recognizing it from a slightly different view, or to hear an unknown foreign word more than once before becoming familiar with it.

From the examples discussed above, it is evident that the act of learning plays a crucial role in all those processes requiring the solution of a pattern recognition task, thus all those processes in which the human brain is required to take an action based on the class of the data it has acquired. For example, hearing a voice and deciding whether it is a male or a female voice, reading a handwritten character and deciding whether it is an $\mathcal{A}$ or a $\mathcal{B}$, touching an object and guessing its temperature, those are typical pattern recognition problems. Notice that this kind of processes represents almost the totality of the processes a human being has to deal with. Finding them a solution has been crucial for humans to survive. For that reason, highly sophisticated neural and cognitive systems have been evolved for such tasks over the past tens of millions of years. The scheme used by the human brain to address pattern recognition tasks is based on two separate phases, namely a training phase and a test phase. In the training phase the human brain gets experienced by dealing with patterns taken from the same population, as landscapes, spoken words, handwritten characters. Then, in the test phase, it applies to patterns of the same population—but previously unseen—what it has learnt during the training phase. In this sense, admiring a known landscape several times—trying to identify its characteristics—represents the training phase, whereas recognizing it from a slightly different view represents the test phase.

As regards machines, the act of learning refers to artificial intelligences—for instance computer programs—which are able to recursively change their own internal structures in response to input patterns in such a manner that their performance in recognizing previously unseen patterns improves. In this context, machine learning is an area of artificial intelligence concerned with the development of techniques which allow machines to learn how to solve pattern recognition problems, whereas learning machines are automata which solve pattern recognition problems. In a similar way to what happens for the human brain, the solution of a pattern recognition problem initially involves the collection of a data set of training patterns. The learning machine structure is then adapted so as to create a mapping from the input patterns to its output values, such that the latter approximate the attended values as closely as possible over the whole training patterns. The recognition performance of the trained learning machine is then evaluated on a data set of test patterns, namely patterns which were not part of the training data set, but which were taken from the same population.

The success of machine learning—since 1960s up to nowadays—is twofold. First, it is evident that implementing learning processes by using machines is fundamental in order to automatically address pattern recognition problems which—due to their complexity—are almost impossible for a human brain to solve. For example, challenging pattern recognition tasks as speech recognition, fingerprint identification, optical character recognition, DNA sequence identification, video surveillance—and much more—can be easily and automatically addressed by means of learning machines. Second, by trying to give answers and explanations to the numerous questions and doubts arising when implementing such automatic learning systems, a deeper understanding of the processes governing human learning is gained. In fact, many techniques in machine learning derive from the efforts gone in order to make more precise the theories of human learning through computational models. At the same time, it seems likely also that the concepts being explored by researchers in machine learning may illuminate certain aspects of biological learning.

Before proceeding, it is well worth specifying in more detail the significance of pattern recognition problems from a more technical perspective, see (Tarassenko, 1998). As already discussed, all those problems requiring a human or an artificial intelligence to take an action based on the data acquired, are formally defined as pattern recognition problems. That family of problems can be further divided into families of sub–problems. The most common and important ones are *pattern classification* problems, *regression* problems and *time–series prediction* problems. Pattern classification problems are those in which the learner is required to learn how to separate the input patterns into two or more classes. A typical pattern classification problem could require—for example—a human brain or a learning machine to separate into two classes the handwritten $\mathcal{A}$s and $\mathcal{B}$s taken from a data set of handwritten characters. When the problem do not require to associate the class of membership to an input pattern, but rather to associate a continuous value, a regression problem is faced. A typical regression problem could require a human brain or a learning machine to associate an age to input patterns represented by pictures of human faces. Finally, time–series prediction problems, in which a learning machine is trained to predict the $(n+1)^{th}$ sample in a time series from the previous $n$ samples, is a special case of a regression problem but which assumes that the underlying data generator is stationary, namely its statistical properties are time–independent. In this work, the whole attention will be concentrated on pattern classification, which is actually the most common type of pattern recognition problem.

### 2.1.2 Learning pattern classification

Specific details about learning pattern classification—namely the way in which learning machines address pattern classification tasks—will be given in this Section. In particular, two important aspects of learning machines will be discussed. First, how they learn directly from data, thus without using any a priori assumption on the classification problem they are facing. Second, how the supervised and unsupervised learning paradigms are implemented in order to practically solve pattern classification problems.

#### Learning from data

One of the most important characteristic of learning machines is that they are not programmed by using some a priori knowledge on the probability structure of the data set considered, but—as anticipated in Section 2.1.1—they are rather trained by being repeatedly shown large numbers of examples for the problem under consideration. In a sense, they learn directly from the data how to separate the different existing classes. This approach determines some important peculiarities of learning machines. First, they are particularly suited for complex classification problems whose solution is difficult to specify a priori. Second, after being trained, they are able to classify data previously not encountered. This is often referred to as the *generalization* ability of learning machines. Finally, since they learn directly from data, then the effective classification solution can be constructed far more quickly than using traditional approaches entirely reliant on a deep knowledge and experience in the particular field to which data refer. In order to stress the importance of an approach purely based on learning from data—in particular when a dynamical model of what is happening behind the scenes does not exist or whenever the underlying dynamics is too complicated—let us mention one enlightening example borrowed from (Schölkopf, 1997):

> When a human writer decides to write a letter, for example the letter $\mathcal{A}$, the actual outcome is the result of a series of complicated processes which cannot be modeled comprehensively in their entirety. The intensity of the lines depends on chemical properties of ink and paper, their shape on the friction between pencil and paper, on the dynamics of the writer's joints and on motor programmes initiated in the brain, these in turn are based on what the writer has learnt at school. The chain could be continued ad infinitum.

It is evident that, in such a situation, it is nearly impossible to address a classification task which is required to separate different handwritten characters—such as for example $\mathcal{A}$s and $\mathcal{B}$s—by modeling the way in which they are written by hand. For this reason, an approach purely based on learning from data is probably the most appropriate solution.

Nevertheless, some approaches in which the probability structure underlying the classes of the data set is known perfectly—or at most its general form—do exist. For example, as described in (Duda *et al.*, 2000), Bayesian decision theory is a fundamental statistical approach to the problem of pattern classification. It makes the assumption that the decision problem is posed in probabilistic terms and that all the relevant probability values are known. In particular, it is based on quantifying the trade offs between various classification decisions using the probability and the costs that accompany such decisions. Unfortunately, for the most part of the applications, the probabilistic structure of the problem is unknown. At most, only some vague and general knowledge about the situation, together with a number of training data representative of the patterns to classify, do exist. The problem is then to find some way to use this information in order to design the classifier. One approach is to use the training patterns for estimating the unknown probabilities and probability densities and to use the resulting estimates as if they were the true values. Let us quote a further example borrowed from (Schölkopf, 1997):

> Suppose, for example, that some temporal sequences of detailed observations of double star systems were given and that the problem is to predict whether, eventually, one of the stars will collapse into a black hole. Given a small set of observations of different double star systems, including target values indicating the eventual outcome, an approach purely based on learning from data would probably have difficulties extracting the desired dependency. A physicist, on the other hand, would infer the star masses from the spectra's periodicity and Doppler shifts, and use the theory of general relativity to predict the eventual fate of the stars.

In this example—differently from the previous one—modeling the stars collapsing into black holes is probably more straightforward, owing to the deep knowledge of those phenomena. For that reason, here it could be more appropriate and effective to address the classification task with a modeling approach rather than with an approach purely based on learning from data.

### Supervised and unsupervised learning

The machine learning scheme can be implemented in two different ways, literally as a *supervised* or as an *unsupervised* scheme. In the former, the input patterns used to train the learning machine are labeled, in other words they are patterns whose class membership is known. In the latter, they are unlabeled, namely their class membership is unknown.

In order to implement the supervised learning scheme, a labeled data set of training patterns must be provided. To this purpose, the training patterns:

$$(\boldsymbol{x_1}, \ldots, \boldsymbol{x_l}) \qquad \text{with} \quad \boldsymbol{x_i} \in \mathbb{R}^n \quad \forall i = 1, \ldots, l \tag{2.1}$$

are needed, as well as the associated labels indicating their class membership:

$$(y_1, \ldots, y_l) \qquad \text{with} \quad y_i = \pm 1 \quad \forall i = 1, \ldots, l \tag{2.2}$$

Each training pattern $i$ is thus represented by a vector $\boldsymbol{x_i}$ of $n$ *features*, namely the individual measurable heuristic properties of the phenomena being observed. Furthermore, it is associated to a specific value of the label $y_i$, which takes values $+1$ or $-1$ according to its class membership. For example, in a pattern classification problem in which digital images representing the handwritten characters $\mathcal{A}$ and $\mathcal{B}$ are required to be separated, the pixel values of each image could be used as classification features. In alternative, some specific measurements on each image—as luminosity, gradient, and so on—could be used as well. As regards the labels, all the patterns belonging to the class of $\mathcal{A}$s could be associated to the label $+1$, whereas those belonging to the class of $\mathcal{B}$s could be associated to the label $-1$.

During the training phase of the supervised learning scheme, the learning machine adjust its internal parameters by being shown the features $\boldsymbol{x_i}$ of patterns taken from class $+1$ and those of patterns taken from class $-1$. Once the training phase is terminated, then the learning machine is supposed to have learnt how to recognize features belonging to the class $+1$ and features belonging to the class $-1$. In particular, it is supposed to have learnt how to correctly separate the two different classes in the $n$–dimensional feature space, as shown in Fig. 2.1. At the end, its generalization performance is tested on a new set of labeled data which was not part of the training set. Typical learning machines which implement the supervised learning scheme in order to solve pattern classification problems are perceptrons and neural networks, whose structures will be outlined in Section 2.1.5, and SVM, which will be discussed in detail in Section 2.2.

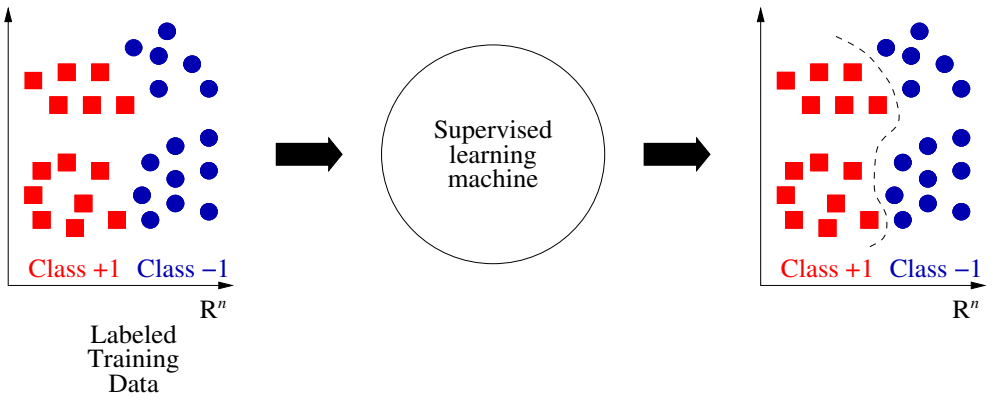



**Figure 2.1:** Supervised learning scheme.

As regards the unsupervised learning scheme, there are three main reasons for being interested in that procedure. First, for many real–world problems the data set does not have labels. This means that only the training patterns:

$$(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_l) \qquad \text{with} \quad \boldsymbol{x}_i \in \mathbb{R}^n \quad \forall i = 1, \ldots, l \tag{2.3}$$

are given and that no class membership is associated to them. The collection and labeling of a large number of patterns is—in fact—a costly and time–consuming exercise. Often, only a small proportion of the total number of training patterns has been assigned a class membership, whereas the great majority has not. For example, speech recording is quite an easy task, but accurately labeling the speech—namely marking what word or phoneme is being uttered at each instant—can be very expensive and time consuming. In this sense, if a classifier can be crudely designed by using a restricted set of labeled patterns—and then tuned up by allowing it to run without supervision on a large unlabeled data set—much time and trouble can be saved. The second reason for being interested in unsupervised learning is that, in the early stages of an investigation, it may be valuable to gain some insight into the nature or structure of the data set without making any a priori assumption. The discovery of distinct sub–classes or major departures from expected characteristics may in fact significantly alter the approach to designing the classifier. Lastly, unsupervised methods can be used to find features that will then be useful for categorization. There are in fact unsupervised methods that represent a form of actual data–dependent preprocessing or feature extraction.

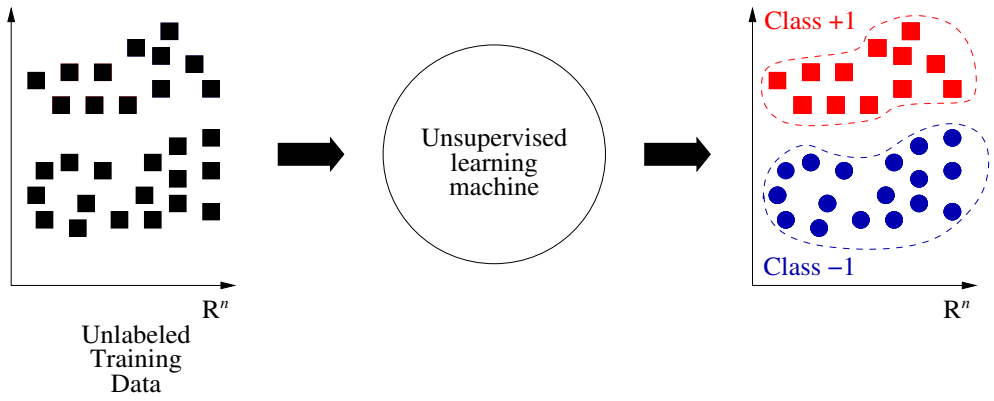



**Figure 2.2:** Unsupervised learning scheme.

Now, is there anything that can be done when all one has is a collection of training patterns without being told their class membership? The answer is yes, namely *cluster analysis* or *clustering*, a form of unsupervised learning which attempts to discover any underlying structure in the data. Clusters are regions in a hyperspace comprised of a number of similar input vectors grouped together. In a pattern classification problem, each training pattern $\boldsymbol{x_i}$ is constituted by the $n$ features characteristic of that specific pattern, thus a cluster can be described as a region in a $n$–dimensional space containing a relatively high density of points, separated from other clusters by regions containing a relatively low density of points. Notice that in order to identify clusters, the key issue is to specify a similarity measure. The most obvious measure of similarity between two patterns is the distance between them. For that reason, a simple form of cluster analysis might involve computing the matrix of distances between all pairs of patterns in the data set. The distance between patterns in the same cluster should then be much less than the distance between pattern in different clusters.

During the training phase of the unsupervised learning scheme, the learning machine is thus provided with a similarity measure—for example the above discussed distance—that measures the clustering quality of any feasible partition of the unlabeled data. Then—as shown in Fig. 2.2—it finds the partition that extremizes that similarity measure, namely finds a clusterization of the $n$–dimensional space. Typical clustering methods are the $k$–means algorithm, introduced for the first time in (Lloyd, 1982), the Mahalanobis algorithm, discussed in (Mao & Jain, 1996), and the hierarchical algorithm, described in (Day & Edelsbrunner, 1984) and (Kaufman & Rousseeuw, 1990).


**Figure 2.3:** Feature extraction. The classification problem is more easily separable using the pair of features $f_3$ and $f_4$ (right) than using $f_1$ and $f_2$ (left).

### Pattern classification sub–issues

In order to improve the classification performance of learning machines, it may be usually convenient to submit data to some pre–processing techniques whose final aim is to elaborate the data so that the classification task could result easier for the learning machine. Even though those pre–processing techniques are not an integral part of the classification task, they play a fundamental role in it. For that reason, they are usually considered as sub–issues of the pattern classification problem. For a more complete discussion of those sub–issues, see (Duda *et al.*, 2000) and (Tarassenko, 1998).

When dealing with pattern classification problems, the first important step must be taken in the direction of choosing the most appropriate features. This literally means selecting the measurable properties of the phenomena under consideration which can be the most discriminant ones for the specific pattern classification problem faced. For example—as depicted in Fig. 2.3—it could happen that the choice of a pair of features, let us say $f_3$ and $f_4$, makes the classification task more easily separable than using a different pair of features, namely $f_1$ and $f_2$. It is evident that extracting the most discriminant features is a problem–dependent task which requires an—even small—a priori knowledge of the data. Notice, furthermore, that the conceptual boundary between feature extraction and classification proper is somewhat arbitrary. An ideal feature extractor—in fact—would yield a representation that makes the classification trivial. On the other hand, a powerful classifier would not need the help of a sophisticated feature extractor.

The second pre–processing step consists in removing the noise characterizing the data. The definition of noise is very general. Any property of the sensed patterns which is due to randomness in the world or in the sensors, rather than to the true underlying model of the data, can be considered as noise. All non–trivial decision and pattern classification problems involve noise in some form, since all real–world data are transduced from sensors into digital data. It is thus unavoidable that they suffer from the specific noise of the experimental set up. Typical examples are visual noise in video cameras, background noise in audio registrations and so on. Notice that a fundamental task in this context is being able to understand somehow whether the variation in data is due to noise or instead to the complexity underlying the problem under consideration.

### 2.1.3 Validation techniques

The introduction of the so–called validation techniques is motivated by the willingness of finding a solution to two fundamental problems in pattern classification, namely the selection of the learning machine's *model* and the estimation—and validation—of its classification performance.

Almost invariably, all learning machines have one or more free parameters which can be tuned up in order to adapt them to each specific classification problem. For example—as it will be briefly discussed in Section 2.1.5—the free parameters of neural networks are represented by the number of layers in the network and by the weights linking each input pattern to each perceptron. When a pattern classification problem is addressed by using learning machines, the typical approach consists in choosing a specific configurations of the free parameters—namely choosing a specific model for the learning machine—and then in estimating its classification performance. The classification performance is usually estimated by the so–called true error rate, literally the learning machine's error rate on the entire population under exam. The configuration of the free parameters for which the true error rate is minimum corresponds to the optimal learning machine's model for that particular problem.

It is evident that, in the ideal and unrealistic situation in which the data set is comprised of an unlimited number of patterns, the straightforward solution to the problem would be first choosing the learning machine's model that provides the lowest error rate on the entire data set and therefore considering that error rate as the true error rate. Obviously, in real–world applications, only finite data sets are available and typically they are smaller than what it would be desirable.

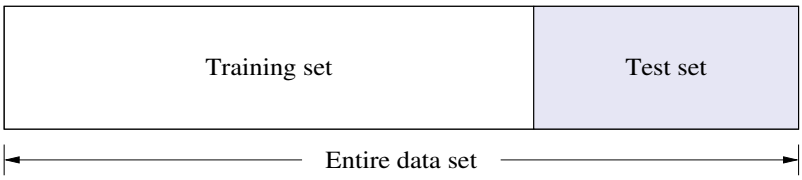



**Figure 2.4:** Holdout method.

In this case, one very crude approach would be to use the entire data set to train the learning machine, to select the model and to estimate the error rate. However that approach suffers from two fundamental drawbacks. First, the final model will normally *overfit* the training data. This basically means that the learning machine results excessively optimized on the training data, thus—following all the small details in them—it looses in generalization performance and gives very poor interpolation on a different data set. Second, the error rate estimate is overly optimistic, typically lower than the true error rate. It is in fact not uncommon to have 100% correct classification on training data.

In order to overcome the above discussed drawbacks, some more sophisticated validation techniques are introduced, namely the holdout, cross–validation and leave–one–out methods.

### Holdout method

An interesting approach consists in splitting the data set into two disjoint subsets, thus applying the so–called *holdout* validation technique. The holdout method—sometimes called *test sample estimation*—partitions the data into two mutually exclusive subsets called training set and test set, in analogy to what discussed in the previous Sections. It is common to designate $2/3$ of the data set as the training set and the remaining $1/3$ as the test set, as depicted in Fig. 2.4. The training set is used to train the learning machine and the trained learning machine is then tested on the test set.

This method suffers from two important drawbacks as well. First, assuming that the learning machine's classification performance increases as more patterns are seen, the holdout method is a pessimistic estimator because only a portion of the data is given to the learning machine for the training phase. Second, since it is a single train–and–test experiment, its estimate of the error rate could be misleading if it happens to get an unfortunate split, namely if it occurs that the test set is composed by all the most difficult patterns of the entire data set.

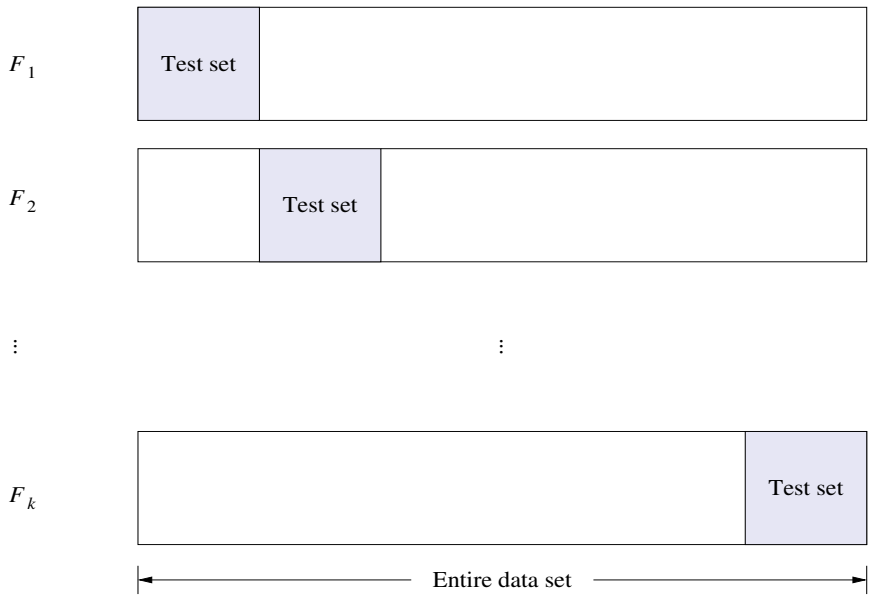



**Figure 2.5:** $k$–fold cross–validation method.

### Cross–validation and leave–one–out methods

A feasible way to overcome those pitfalls is $k$–fold *cross–validation*—known also as *rotation estimation*—a technique in which the data set $\mathcal{D}$ is randomly split into $k$ mutually exclusive folds $\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_k$ of approximately equal size. In a specific case, know as *stratified cross–validation*, the folds are stratified so that they contain approximately the same proportions of labels as the original data set. Here the learning machine is trained and tested $k$ times, namely for each time $t \in \{1, 2, \ldots, k\}$ it is trained on $\mathcal{D} \setminus \mathcal{D}_t$ and tested on $\mathcal{D}_t$, as shown in Fig. 2.5.

The major advantage of this technique with respect to the holdout method is that all the patterns in the data set are used for both training and testing. At the same time, the true error is estimated as the average error rate on test patterns, thus preventing the problems arising from unfortunate splits of the data set:

$$e = \frac{1}{k} \sum_{i=1}^{k} e_i \tag{2.4}$$

It is worth noticing that the estimate in Eq. 2.4 is a number that depends on the division into folds. In (Kohavi, 1995), some interesting considerations concerning the choice of the correct number of folds $k$ are drawn. First, if the number of folds

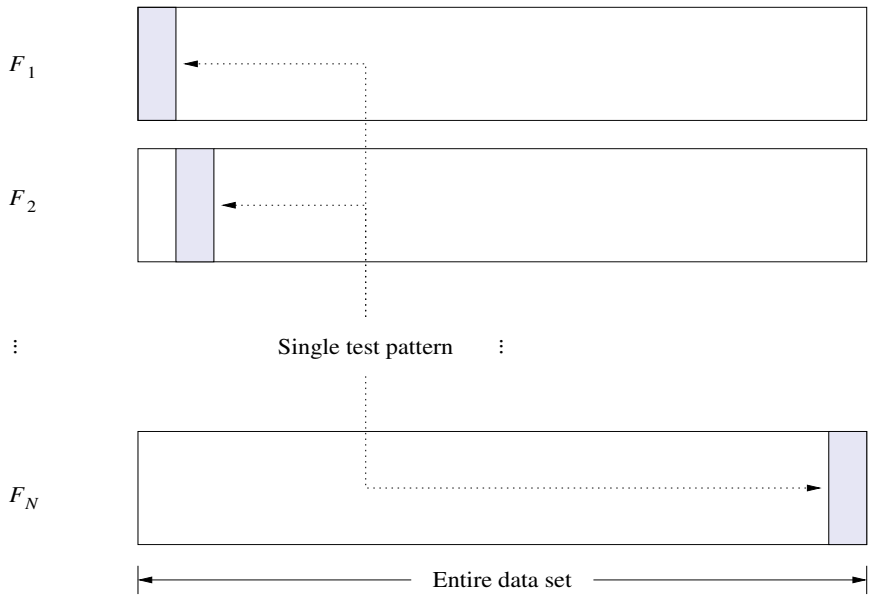



**Figure 2.6:** Leave–one–out method.

$k$ is large, the bias of the true error estimate is generally small, thus the estimator may be considered very accurate. Unfortunately—due to the large number of iterations—the variance of the true error rate estimator as well as the computational times are expected to be large. Second, if the number of folds $k$ is reduced, the bias of the true error estimate is generally large, thus the estimator may be considered conservative or higher than the true error rate. In that case—due to the reduced number of iterations—the variance of the true error rate estimator as well as the computational times are typically small.

In practice, the choice of the number of folds strongly depends on the size of the data set. For large data sets, even a 3–fold cross–validation could be quite accurate. For sparse data sets, it may be necessary to partition the data set in a larger number of folds in order to train on as many patterns as possible. A common choice for $k$–fold cross validation is $k = 10$. Finally, for very sparse data sets, a *leave–one–out* validation technique may be implemented. This approach is analogous to that of cross–validation. The only difference here is that—if $N$ is the number of patterns in the data set—the learning machine is trained $N$ times. In particular, for each time, $N - 1$ patterns are used for training and the remaining for testing, as shown in Fig. 2.6. As for the cross–validation technique, the true error is estimated as the average error rate on test patterns.

### 2.1.4 Performance visualization

Receiver Operating Characteristic (ROC) and Free–response Receiver Operating Characteristic (FROC) curves are advanced techniques for visualizing the performance of a learning machine on a given pattern classification problem. In recent years they have seen an increasing popularity in the machine learning community and in this work they will be used in order to visualize several classification results as well. For that reason, some introductory concepts will be given in the following Section.

As already discussed, a learning machine's model corresponds to a specific configuration of its free parameters. In a pattern classification problem—once the learning machine's model is fixed—an unambiguous mapping from the input pattern to its predicted class does exist. In order to distinguish between the actual positive or negative class membership of the input pattern and that of the predicted one, the labels $\{p^a, n^a\}$ are used for the former, whereas the labels $\{p^p, n^p\}$ for the latter. For each input pattern there are thus four possible outcomes. If the attended class membership of the input pattern is positive ($p^a$) and the predicted one is positive ($p^p$) then it is counted as a *true positive*. If the predicted class membership is negative ($n^p$) then it is counted as a *false negative*. If the attended class membership of the input pattern is negative ($n^a$) and the predicted one is negative ($n^p$) then it is counted as a *true negative*. If the predicted class membership is positive ($p^p$) then it is counted as a *false positive*. Thus, given a learning machine and a set of patterns, a two–by–two *confusion matrix*—also called *contingency table*—representing the dispositions of the set of patterns can be built, as shown in Fig. 2.7. In particular, the numbers along the major diagonal represent the correct decisions made, whereas the numbers off this diagonal represent the errors or the confusion between the classes.

The confusion matrix forms the basis for some important metrics. First, the *True Positive Fraction* ($TPF$) of a learning machine is defined as:

$$TPF = \frac{\text{Positives correctly classified}}{\text{Total positives}} \tag{2.5}$$

Analogously, the *False Positive Fraction* ($FPF$) is defined as:

$$FPF = \frac{\text{Negatives incorrectly classified}}{\text{Total negatives}} \tag{2.6}$$

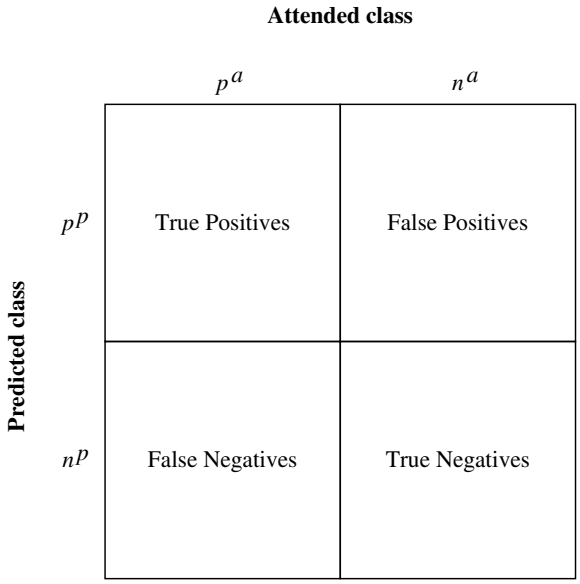



**Figure 2.7:** Confusion matrix.

Two further metrics strictly associated to those discussed above are the *sensitivity*, which corresponds to $TPF$:

$$\text{sensitivity} = TPF \tag{2.7}$$

and the *specificity*, which is given by:

$$\text{specificity} = \frac{\text{True negatives}}{\text{False positives} + \text{True negatives}} = 1 - FPF \tag{2.8}$$

Those metrics are fundamental in order to define ROC and FROC curves and to understand how they can be used as visualization tools for the classification performance of learning machines.

### ROC curve

ROC curves are two–dimensional graphs which represent the relative trade off between the sensitivity and the specificity of a learning machine applied to a pattern classification problem. The reason is that on the $y$ axis of a ROC curve the value $TPF$ obtained by the classifying learning machine is plotted, whereas on the $x$ axis the value $FPF$, as shown in Fig. 2.8.

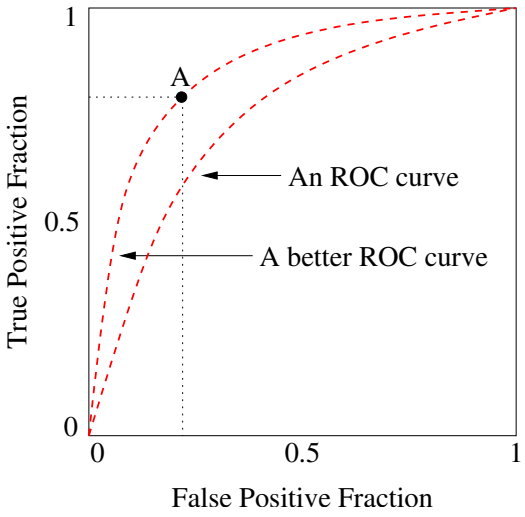



**Figure 2.8:** An ROC curve.

What happens in the typical solution of a pattern classification problem, is that a trained learning machine is tested on a test set, thus predicting a single class membership—$p^p$ or $n^p$—for each input pattern. From the considerations drawn above, it is evident that this yields a single confusion matrix, which in turn corresponds to one single ROC point, as the point A in Fig. 2.8. In order to generate a full ROC curve instead of just a single point, the most common technique consists in varying the free parameters of the learning machine, thus altering the values of $TPF$ and $FPF$ on the same test set. Varying those free parameters, an ROC curve can be generated which shows the trade off between $TPF$ and $FPF$ associated with the different values that the parameters may assume. It would then be possible to trade a lower—or higher—$FPF$ value for a higher—or lower—$TPF$ value by choosing appropriate values for the free parameters in question.

Several points in ROC space deserve some deeper considerations. The lower left point $(0, 0)$ represents a learning machine which is unable of issuing a positive classification. Such learning machine commits in fact no false positive errors but also gains no true positives. The upper right point $(1, 1)$ represents the opposite, namely a learning machine which unconditionally issues positive classifications. Such learning machine classifies incorrectly all the negative patterns but correctly all the positive ones. Finally, the point $(0, 1)$ represents a learning machine which is able of perfect classification. Such learning machine classifies correctly all the positive patterns without false positive errors.

Reasoning informally as in (Fawcett, 2004), one point in ROC space is better than another if it is to the northwest of the first. This in fact means a higher $TPF$ and a lower $FPF$. Learning machines appearing on the left hand–side of an ROC graph, near the $x$ axis, may be thought of as conservative. They make positive classifications only with strong evidence so they make few false positive errors, but they often have low true positive fractions as well. On the contrary, learning machines on the upper right–hand side of an ROC graph may be thought of as liberal. They make positive classifications with weak evidence so they classify nearly all positives correctly, but they often have high false positive fractions.

Being such a useful performance graphing method, ROC curves have been rapidly extended to several research areas, such as medical decision making, machine learning and data mining. In particular, they have long been used in signal detection theory to depict the trade off between benefits—true positives—and costs—false positives—of learning machines, see (Egan, 1975). ROC analysis has been furthermore extended to the medical decision making community for use in visualizing the behavior of diagnostic systems, as described in (Swets, 1988). Recently, (Spackman, 1989) adopted ROC curves in machine learning demonstrating their value in evaluating and comparing algorithms, whereas (Swets *et al.*, 2000) brought ROC curves to the attention of the wider public with a Scientific American article.

### FROC curve

For a specific family of pattern classification problems such as target detection in digital images—where targets can be faces, pedestrians, tumoral masses, geomorphological features and so forth—ROC curves prove to be not very useful. Those systems are in fact typically based on a trained learning machine which is required to recognize specific targets—as the ones mentioned above—by classifying different sub–regions of the digital images under consideration. In such a situation, therefore, what does interest is not the false positive fraction $FPF$ given a specific true positive fraction $TPF$, but rather the average number of false positive errors per–image given such a true positive fraction $TPF$.

In order to deal with that family of pattern recognition problems, FROC curves have thus been introduced, namely plots of the true positive fraction $TPF$ versus the average number of false positive errors per–image.

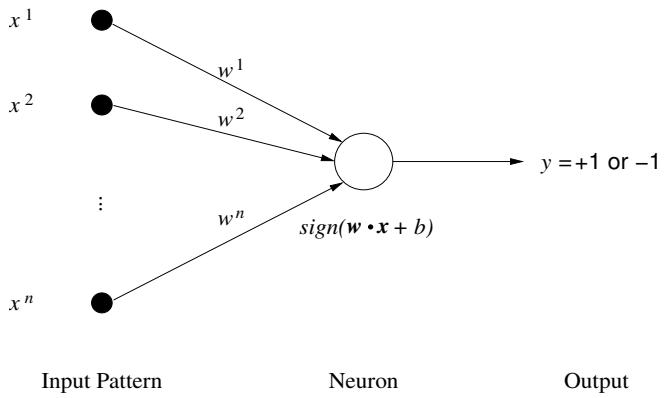



**Figure 2.9:** The Rosenblatt's perceptron.

### 2.1.5 Historical perspective

Before proceeding with the discussion of a specific learning machine such as SVM, it may be useful to have a wider picture of machine learning by reviewing its history and major results. Three periods are particularly relevant:

- **1960s and 1970s:** the first learning machine is introduced (known as the Rosenblatt's perceptron) and the fundamentals of learning theory are posed
- **1980s:** Neural networks are introduced
- **1990s:** Alternatives to neural networks start being explored

#### 1960s and 1970s

The starting point for the mathematical analysis of learning processes dates back to approximately 40 years ago, when F. Rosenblatt introduced the first model of a learning machine, namely the *perceptron* (Rosenblatt, 1962). In a sense, Rosenblatt's perceptron was not new, in fact this model had been discussed in the neurophysiologic literature for many years. However, the unusual aspect was that he described this model as a program for computers and demonstrated that it solves pattern recognition problems using given examples. To construct such a learning rule, the perceptron uses adaptive properties of the simplest neuron model, the so–called McCullogh–Pitts model, as shown in Fig. 2.9. According to that model, the neuron has an input $\boldsymbol{x} = (x^1, \ldots, x^n) \in \mathbb{R}^n$, one output $y \in \{-1, +1\}$

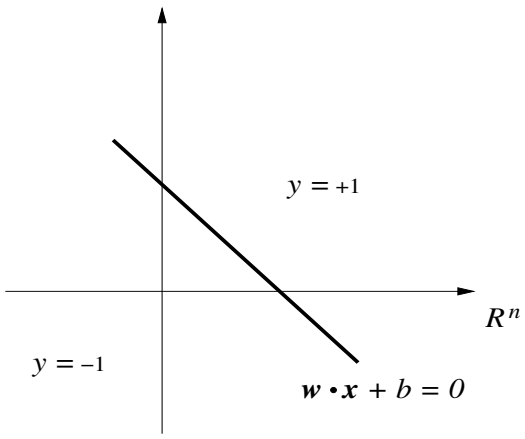



**Figure 2.10:** The separating hyperplane.

and $n$ weights $\boldsymbol{w} = (w^1, \ldots, w^n) \in \mathbb{R}^n$ linking the input to the neuron. The output is connected with the inputs by the functional dependence:

$$y = sign(\boldsymbol{w} \cdot \boldsymbol{x} + b) \tag{2.9}$$

where $\cdot$ is the dot product of two vectors, $b$ is a threshold value and $sign(u) = +1$ if $u > 0$ and $sign(u) = -1$ if $u \leq 0$. Geometrically speaking, the neuron divides the $n$–dimensional hyperspace into two regions, a region where the output $y$ takes the value $+1$ and a region where the output $y$ takes the value $-1$. Those two regions are separated by the hyperplane:

$$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0 \tag{2.10}$$

where the vector $\boldsymbol{w}$ and the scalar $b$ determine the position of the separating hyperplane, as shown in Fig. 2.10. This means that the neuron assigns input patterns represented by the vector of numbers $\boldsymbol{x} = (x^1, \ldots, x^n)$, either to one class or to the other class according to the output value of $y$.

In the specific context of the Rosenblatt's perceptron, *learning* consists of adjusting the $\boldsymbol{w}$ weights so that the neuron performs the classification task correctly, or as close as possible to it. In order to arrive at a weight set which solves the problem, an error feedback is used to adjust the weights during the training phase. The idea is to measure an error $E$ at the output of the neuron and then to minimize it by gradient descent. Starting with an arbitrarily chosen weight vector $\boldsymbol{w}(0)$ and computing the gradient of the error with respect to each weight—for example $\partial E/\partial w^i$ for weight $w^i$—the next vector $\boldsymbol{w}(1)$ is obtained by moving a small

distance in the direction of the steepest descent, namely along the negative of the gradient. For an individual weight $w^i$ in the weight vector $\boldsymbol{w}(0)$, the weight update $\Delta w^i$ is thus given by:

$$\Delta w^i = -\eta \frac{\partial E}{\partial w^i} \tag{2.11}$$

where $\eta$ is a small parameter which sets the step size. In order to specify a suitable error function $E$, consider that for input patterns $\boldsymbol{x}$ taken from one class:

$$y = sign(\boldsymbol{w} \cdot \boldsymbol{x} + b) = +1 \quad \Leftrightarrow \quad \boldsymbol{w} \cdot \boldsymbol{x} + b > 0 \tag{2.12}$$

whereas for input patterns $\boldsymbol{x}$ taken from the other class:

$$y = sign(\boldsymbol{w} \cdot \boldsymbol{x} + b) = -1 \quad \Leftrightarrow \quad \boldsymbol{w} \cdot \boldsymbol{x} + b \leq 0 \tag{2.13}$$

This means that, for all input patterns $\boldsymbol{x}$, it results that:

$$(\boldsymbol{w} \cdot \boldsymbol{x} + b)a > 0 \tag{2.14}$$

where $a$ is the attended value for that particular pattern, namely $+1$ for the first class and $-1$ for the second class. It is thus reasonable to define the following error function:

$$E = -\sum_{\boldsymbol{x} \in \mathcal{M}} (\boldsymbol{w} \cdot \boldsymbol{x} + b)a \tag{2.15}$$

where $\mathcal{M}$ is the set of input patterns $\boldsymbol{x}$ which are misclassified by the current set of weights $\boldsymbol{w}$. Eq. 2.15 is also known as the perceptron criterion. Learning is thus an iterative process whereby all the training patterns are presented in turn several times and, for each pattern presentation, Eq. 2.11 is applied to each weight:

$$\Delta w^i = -\eta \frac{\partial E}{\partial w^i} = -\eta \frac{\partial}{\partial w^i} \left( -\sum_{\boldsymbol{x} \in \mathcal{M}} (\boldsymbol{w} \cdot \boldsymbol{x} + b)a \right) = \eta x^i a \tag{2.16}$$

Very small changes of weights are repeatedly made until a set of weights is obtained which minimizes the error function $E$ over all the patterns in the training set. For any data set which is linearly separable, it can be guaranteed to find a solution in a finite number of steps.

Except for the introduction of the Rosenblatt's perceptron, nothing extraordinary happened as regards the applied analysis of the learning processes during the time between 1960 and 1980. On the contrary, those years were extremely fruitful for the development of the statistical learning theory. In 1962, A. Novikoff proved the first theorem about the perceptron (Novikoff, 1962). This theorem actually started the learning theory, connecting the cause of the generalization ability of a learning machine—namely its ability in recognizing new patterns—with the principle of minimizing the number of errors on the training set. Then, from 1968, the complete framework of statistical learning theory started being developed. First, the essential concepts of the emerging theory, as the VC entropy and the VC dimension, were discovered and introduced for the pattern recognition problem. Using these concepts, then, the bounds for the rate of convergence of a learning machine were obtained, as discussed in (Vapnik & Chervonenkis, 1968). Moreover, the obtained bounds made the introduction of a novel inductive principle possible, namely the Structural Risk Minimization, thus completing the development of the statistical pattern recognition learning theory (Vapnik & Chervonenkis, 1974). Finally, between 1976 and 1981, the results originally obtained were then generalized for the set of real functions (Vapnik, 1979).

### 1980s

The main drawback for the Rosenblatt's perceptron is that it is unable to cope with data sets which are not linearly separable. It is evident that this is a great problem, since real–world data are intrinsically noisy, in other words there are always regions of overlap in the input space such that some of the patterns end up on the wrong side of the decision boundary. In the late 1960s, it was recognized that a way to overcome that problem is the *multi–layered perceptron*, namely an architecture with several layers of neurons, as depicted in Fig. 2.11. However, no further progress was possible at that time, since no learning rule existed for adjusting the weights of the first layer on the basis of the error at the output of the second layer.

In 1986 several authors independently proposed a method for simultaneously updating the weights of a multi–layered perceptron, namely the *back–propagation* method (LeCun, 1986) and (Rumelhart *et al.*, 1986). Multi–layered perceptrons trained with this method were renamed *neural networks*, see (Bishop, 1995) and (Haykin, 1999). The idea of this method is rather simple. First, in order to deal with non linearly separable data, the sum–of–squares error function is introduced

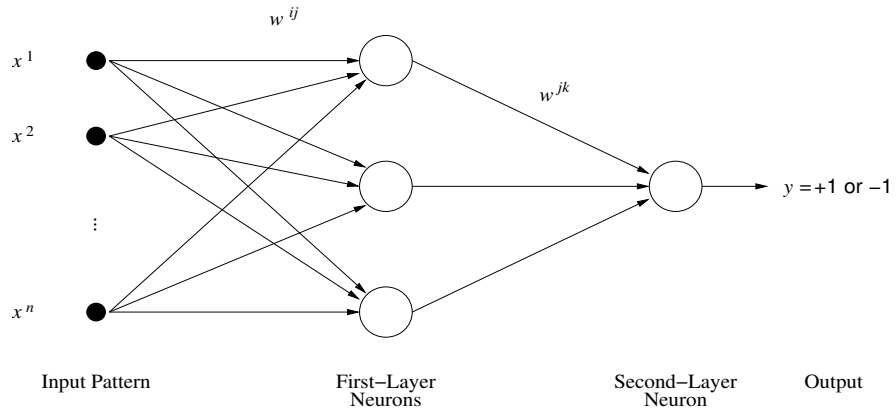



**Figure 2.11:** Multi–layered perceptron or neural network.

as the criterion to minimize:

$$E = \sum_{p=1}^{P} (y^p - a^p)^2 \tag{2.17}$$

where $y^p$ is the single output of the multi–layer perceptron for pattern $p$ and $a^p$ is the corresponding attended value. Second, in order to minimize $E$ using gradient descent as described in 2.11, the error function must be differentiable with respect to every weight in the network. For this reason, instead of the McCullogh–Pitts model of the neuron, a slightly modified model is considered, where the discontinuous function $sign(\boldsymbol{w} \cdot \boldsymbol{x} + b)$ is replaced by the continuous sigmoid approximation:

$$y = S(\boldsymbol{w} \cdot \boldsymbol{x} + b) = tanh(\boldsymbol{w} \cdot \boldsymbol{x} + b) \tag{2.18}$$

thus by a monotonic function which, as shown in Fig. 2.12, has the properties:

$$S(-\infty) = -1 \tag{2.19}$$

$$S(+\infty) = +1 \tag{2.20}$$

Provided that the error function is defined as in Eq. 2.17 and the model adopted is the sigmoid one as described in Eq. 2.18, it is easy to demonstrate that the minimization of $E$ using gradient descent results in the propagation of errors from the layers near the output of the network backwards to the layers near the input of the network. This gives rise to the name of the algorithm.

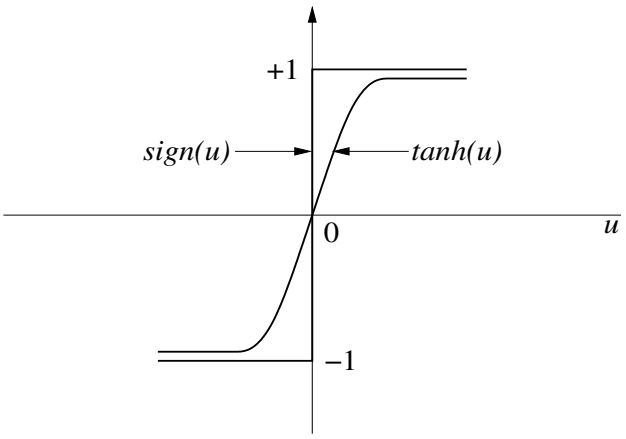



**Figure 2.12:** The discontinuous function $sign(u) = \pm 1$ is approximated by the smooth function $tanh(u)$.

### 1990s

The approach to machine learning has changed in the last years, since more attention is now focused on the alternatives of neural networks. Statistical learning theory is nowadays more popular and attractive for researchers than in the early 1960s. In addition, it now plays a more active role rather than covering only the theoretical and formal aspects of machine learning. In fact, after the completion of the general analysis of learning processes, the research in the area of the synthesis of optimal algorithms, which possess the highest level of generalization ability for any number of observations, was started. Thus, in the last decade, many ideas have appeared in the machine learning community deeply inspired by statistical learning theory. On the contrary to previous ideas of developing learning algorithms inspired by the biological learning process, the new ideas were inspired by attempts to minimize theoretical bounds on the error rate obtained as a results of formal analysis of the learning processes. These ideas—often contradicting the biological paradigm—result in algorithms having nice mathematical properties (such as uniqueness of the solution, simple method of treating a large number of examples, independence of dimensionality of the input space) and excellent performance. In particular, they outperform the state–of–the–art solutions obtained by the old methods.

## 2.2 Support Vector Machine

SVM is one of the shining peaks among the many learning algorithms deeply inspired by statistical learning theory and appeared in the machine learning community in the last decades. Its original formulation is quite recent and is mainly due to (Vapnik & Chervonenkis, 1974; Boser *et al.*, 1992; Guyon *et al.*, 1993; Cortes & Vapnik, 1995; Vapnik, 1995, 1998).

As previously discussed, during the 1990s many learning algorithms arose contradicting the biological paradigm, since more inspired by the minimization of theoretical bounds on the error rate. SVM is not an exception in that. As described in (Burges, 1988), in fact, for a given learning task and with a finite amount of training patterns, SVM is a learning machine which achieves its best generalization performance by finding the right balance between the accuracy obtained on that particular training set and the complexity of the machine, namely its ability in learning any training set without error.

In the next Section, some introductory notions on statistical learning theory will be given, in order to demonstrate that finding the right balance between accuracy and capacity is equivalent to find the minimum of a theoretical bound on the error rate. Then, the mathematical details of SVM will be discussed.

### 2.2.1 Statistical learning theory

Suppose that $l$ training patterns are given:

$$(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_l) \qquad \text{with} \quad \boldsymbol{x}_i \in \mathbb{R}^n \quad \forall i = 1, \ldots, l \tag{2.21}$$

together with the associated labels representing the attended class membership:

$$(y_1, \ldots, y_l) \qquad \text{with} \quad y_i = \pm 1 \quad \forall i = 1, \ldots, l \tag{2.22}$$

Assume also that patterns are generated i.i.d (independently and identically distributed) according to an unknown probability distribution function $P(\boldsymbol{x}, y)$. As already known, learning machines address the task of pattern classification by finding a rule which assigns to each input pattern a class membership. In particular, during the training phase, a mapping $f : \mathbb{R}^n \to \{\pm 1\}$ is created between input patterns and labels, such that the learning machine is expected to correctly classify unseen test examples $(\boldsymbol{x}, y)$. Now, the best mapping $f$ that one can obtain is the one minimizing the *expected error* or *expected risk*:

$$R[f] = \int \frac{1}{2} |y - f(\boldsymbol{x})| \, dP(\boldsymbol{x}, y) \tag{2.23}$$

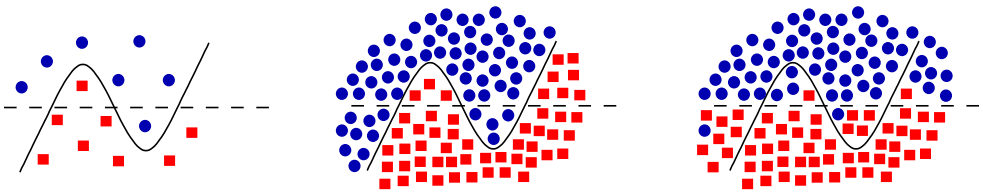


**Figure 2.13:** Overfitting phenomenon. The more complex function obtains a smaller training error than the linear function (left). But only with a larger data set it is possible to decide whether the more complex function really performs better (middle) or overfits (right).

where the function $\frac{1}{2}|y - f(\boldsymbol{x})|$ is called the *loss function*. A further common loss function is for example $(y - f(\boldsymbol{x}))^2$, also known as squared loss function

Unfortunately, the expected error cannot be directly minimized, in fact the probability distribution function $P(\boldsymbol{x}, y)$—from which data are generated—is unknown. In order to estimate a function $f$ that is close to the optimal one, an *induction principle* for risk minimization is therefore necessary. The most straightforward way is probably to approximate the minimum of the risk discussed in Eq. 2.23 by the minimum of the so–called *empirical risk*, namely the measured mean error rate on the training set:

$$R_{emp}[f] = \frac{1}{l}\sum_{i=1}^{l}\frac{1}{2}|y - f(\boldsymbol{x}_i)| \tag{2.24}$$

Notice that no probability distribution function $P(\boldsymbol{x}, y)$ appears here and that for $l \to +\infty$ the empirical risk will converge toward the expected risk.

However, a small error on the training set does not necessarily imply a high generalization ability, namely a small error on an independent test set. As already anticipated in Section 2.1.3, this phenomenon is known as *overfitting*. In particular—as described in (Müller *et al.*, 2001)—given a small training data set as the one on the left example of Fig. 2.13, functions $f$ with higher degrees of complexity may result in a smaller training error. Nevertheless, only with a larger data set—as the ones on the middle and right examples of Fig. 2.13—it is possible to understand which decision reflects the true distribution more closely and does not overfit. One way to avoid this problem is generally to restrict the complexity of the function $f$. The idea is that a simple function as a linear function, explaining most of the data, is preferable to a complex one. These considerations, in turn, give raise to the problem of how to find the optimal complexity of the function.

A specific way of controlling the complexity of a function is given by the Structural Risk Minimization (SRM) principle, see (Vapnik, 1979). In order to understand how it works, it is first necessary to introduce a non–negative integer $h$, called *Vapnik–Chervonenkis dimension* or *VC–dimension*, which describes the complexity of a class of functions. In particular, it measures how many training points can be separated for all possible labellings using functions of that class. Once the concept of VC–dimension is introduced, a nested family of function classes must be constructed:

$$F_1 \subset F_2 \subset \ldots \subset F_k \tag{2.25}$$

whose VC–dimension satisfy:

$$h_1 \leq h_2 \leq \ldots \leq h_k \tag{2.26}$$

Then suppose that the solutions of the empirical risk minimization problem of Eq. 2.24:

$$f_1 \leq f_2 \leq \ldots \leq f_k \tag{2.27}$$

respectively belong to the function classes $F_i$, $i = 1, \ldots, k$. In that context, the SRM principle chooses the function $f_i$ in the class $F_i$ such that the right–hand side of the following bound on the generalization error is minimized:

$$R[f] \leq R_{emp}[f] + \sqrt{\left(\frac{h\left(\log\frac{2l}{h} + 1\right) - \log\left(\frac{\eta}{4}\right)}{l}\right)} \tag{2.28}$$

Here $h$ is the VC–dimension of the function class under consideration, the square root term is called *confidence term* and the bound holds with probability $1 - \eta$, for any $0 \leq \eta \leq 1$.

Three aspects are of great interest in the above bound. First, it is independent of $P(\boldsymbol{x}, y)$. It assumes only that the entire data set—thus both training and test set—is drawn independently according to some $P(\boldsymbol{x}, y)$. Second, it is usually not possible to compute the left–hand side, namely the expected risk. Third, known the VC–dimension $h$, the right–hand side is easily computable. This means that, the selection of the learning machine which maps the input patterns to their class memberships by the function $f : \mathbb{R}^n \rightarrow \{\pm 1\}$ and minimizes the right–hand side of Eq. 2.28, actually corresponds to the selection of the learning machine which gives the lowest upper bound on the expected risk.

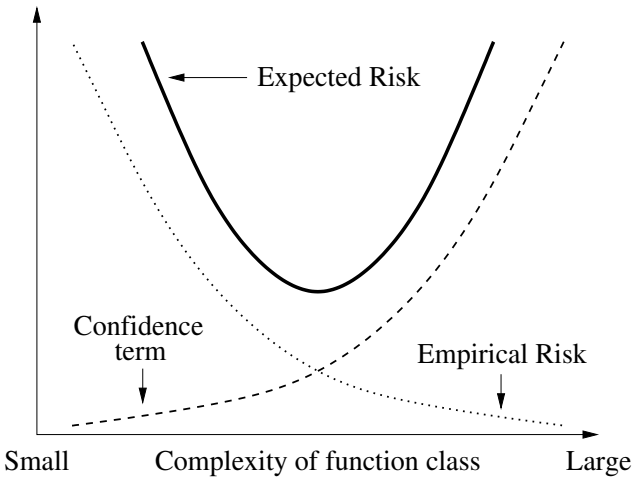



**Figure 2.14:** Schematic illustration of the bound in Eq. 2.28. The dotted line represents the empirical risk $R_{emp}[f]$. The dashed line represents the confidence term. The continuous line represents the expected risk $R[f]$. The best solution is found by choosing the optimal trade off between the confidence term and the empirical risk $R_{emp}[f]$.

Notice that minimizing the expected risk $R[f]$ can be achieved by obtaining a small training error $R_{emp}[f]$ while keeping the function class as small as possible. However, two extreme situations may arose. A very small function class gives a vanishing square root term, but a large training error. On the other hand, a huge function class gives a vanishing empirical error, but a large square root term. Nevertheless, from those considerations, it is evident that the best solution of the problem is usually in between, as shown in Fig. 2.14. In other words, finding the minimum of the expected error, actually means finding the right trade off between the accuracy obtained on that particular training set and complexity of the mapping created by the learning machine. Notice furthermore that the bound of Eq. 2.28 is not unique and similar formulations are available for different loss functions and complexity measures.

In practical problems the bound on the expected error discussed in Eq. 2.28 is neither easily computable nor very helpful. Typical problems are that the VC–dimension of the function class under consideration is unknown or infinite, in which case an infinite number of training data would be necessary. Nevertheless, the existence of bounds is important from a theoretical point of view, since it offers some deeper insights into the nature of learning algorithms.

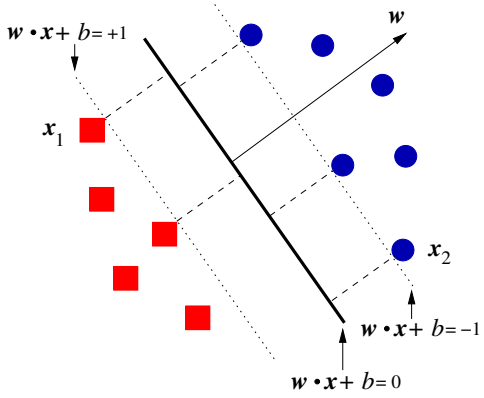



**Figure 2.15:** A hyperplane separating different patterns. The margin is the minimal distance between pattern and the hyperplane, thus here the dashed lines.

### 2.2.2 Linking statistical learning theory to SVM

Linear learning machines such as the perceptronand SVM—as it will be clarified in the next Section—use hyperplanes to separate classes in the feature space, see Fig. 2.15, namely functions of the form:

$$y = sign\,(\boldsymbol{w} \cdot \boldsymbol{x} + b) \tag{2.29}$$

In (Vapnik & Chervonenkis, 1974; Vapnik, 1995) it has been demonstrated that for the class of hyperplanes, the VC–dimension itself can be bounded in terms of another quantity called *margin*, namely the minimal distance of patterns from the hyperplane. In Fig. 2.15 the margin corresponds to the dashed lines. In particular, by rescaling $\boldsymbol{w}$ and $b$ such that the points closest to the hyperplane satisfy $|\boldsymbol{w} \cdot \boldsymbol{x} + b| = 1$—namely transforming the hyperplane to its canonical representation—it is possible to measure directly the margin as a function of $\boldsymbol{w}$. Consider, in fact, two patterns $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ belonging to two different classes and such that $\boldsymbol{w} \cdot \boldsymbol{x}_1 + b = +1$ and $\boldsymbol{w} \cdot \boldsymbol{x}_2 + b = -1$. Then the margin can be calculated as the distance between those two points along the perpendicular, namely as:

$$\frac{\boldsymbol{w}}{||\boldsymbol{w}||}(\boldsymbol{x}_1 - \boldsymbol{x}_2) = \frac{2}{||\boldsymbol{w}||} \tag{2.30}$$

Now, it could be demonstrated that the inequality which links the VC–dimension of the class of separating hyperplanes to the margin is the following:

$$h \leq \Lambda^2 R^2 + 1 \quad \text{and} \quad ||\boldsymbol{w}|| \leq \Lambda \tag{2.31}$$

where $R$ is the radius of the smallest ball around the data. Notice that, due to the inverse proportionality between the margin and $\|\boldsymbol{w}\|$, Eq. 2.31 essentially states that a small VC–dimension is obtained by requiring a large margin. On the other hand, a high VC–dimension is obtained by requiring a small margin. Recalling that the bound described by Eq. 2.28 demonstrates that in order to achieve a small expected error it is necessary to keep small both the training error and the VC–dimension, then—when working with linear learning machines—separating hyperplanes could be constructed such that they maximize the margin and separate the training patterns with as few errors as possible. As it will be demonstrated in the next Section, this result forms the basis of the SVM learning algorithm.

### 2.2.3 Linear SVM

#### The separable case

In order to introduce the SVM learning algorithm, the simplest case to deal with is the so–called *separable case* in which data are linearly separable. As it will be discussed in the following Section, the most general case—namely non–linear SVM trained on non–separable data—results in a similar solution.

Suppose again that $l$ training patterns are given:

$$(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_l) \qquad \text{with} \quad \boldsymbol{x}_i \in \mathbb{R}^n \quad \forall i = 1, \ldots, l \tag{2.32}$$

together with the associated labels representing the attended class membership:

$$(y_1, \ldots, y_l) \qquad \text{with} \quad y_i = \pm 1 \quad \forall i = 1, \ldots, l \tag{2.33}$$

Assume also that they are linearly separable, namely they could be separated by an hyperplane $y = sign\,(\boldsymbol{w} \cdot \boldsymbol{x} + b)$ as the one shown in Fig. 2.15. For such a learning machine, the conditions for classification without training error are:

$$y_i\,(\boldsymbol{w}_i \cdot \boldsymbol{x}_i + b) \geq 1 \quad i = 1, \ldots, l \tag{2.34}$$

The final aim of learning is thus finding $\boldsymbol{w}$ and $b$ such that the expected risk is minimized. According to Eq. 2.28, one strategy is to keep the empirical risk zero by forcing $\boldsymbol{w}$ and $b$ to the perfect separation of the two classes, while at the same time minimizing the complexity term which is a monotonically increasing function of the VC–dimension $h$. Since for a linear learning machine the VC–dimension $h$ is bounded as described in Eq. 2.31, it is thus possible to minimize the VC–dimension by minimizing $\|\boldsymbol{w}\|^2$, namely by maximizing the margin.

The linear learning machine which ensures the lowest expected risk is thus that which gives an empirical risk zero, or perfect separation between the two classes:

$$y_i \left( \boldsymbol{w}_i \cdot \boldsymbol{x}_i + b \right) \geq 1 \quad i = 1, \ldots, l \tag{2.35}$$

and at the same time minimizes the VC–dimension, or maximize the margin between the two classes:

$$\min_{\boldsymbol{w},b} \quad \frac{1}{2} \left\| \boldsymbol{w} \right\|^2 \tag{2.36}$$

In order to solve this convex optimization problem, it is preferable to introduce a Lagrangian $\mathcal{L}$:

$$\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \left\| \boldsymbol{w} \right\|^2 - \sum_{i=1}^{l} \alpha_i \left( y_i \left( \boldsymbol{w}_i \cdot \boldsymbol{x}_i + b \right) - 1 \right) \tag{2.37}$$

with the Lagrangian multipliers satisfying $\alpha_i \geq 0$, $i = 1, \ldots, l$. The Lagrangian $\mathcal{L}$ has thus to be minimized with respect to $\boldsymbol{w}$ and $b$ and to be maximized with respect to $\alpha_i$. The condition that at the saddle points the derivatives vanish:

$$\frac{\partial \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha})}{\partial b} = 0 \tag{2.38}$$

$$\frac{\partial \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha})}{\partial \boldsymbol{w}} = 0 \tag{2.39}$$

leads to:

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{2.40}$$

$$\boldsymbol{w} = \sum_{i=1}^{l} \alpha_i y_i \boldsymbol{x}_i \tag{2.41}$$

By substituting Eq.2.41 in Eq.2.37, the dual quadratic optimization problem is obtained:

$$\max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,k=1}^{l} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j \tag{2.42}$$

subject to:

$$\alpha_i \geq 0, \quad i = 1, \ldots, l \tag{2.43}$$

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{2.44}$$

Thus, by solving the dual optimization problem, the lagrangian multipliers $\alpha_i \geq 0$, $i = 1, \ldots, l$— needed to express the specific $\boldsymbol{w}$ which solves Eq. 2.36—are found. In particular, for each input pattern $\boldsymbol{x}$, the following decision function will be applied:

$$f(\boldsymbol{x}) = sign\left(\sum_{i=1}^{l} \alpha_i y_i \boldsymbol{x} \cdot \boldsymbol{x}_i + b\right) \tag{2.45}$$

The above hyperplane is also known as *Maximal Margin Hyperplane* (MMH).

### The non–separable case

When dealing with noisy data, it could happen that they are not linearly separable. In such a situation it is impossible to keep the empirical error zero, therefore it is necessary to find the best trade off between the empirical risk and the complexity term as discussed for Eq. 2.28. In order to relax hard–margin constraints—thus allowing classification errors—slack variables are introduced as discussed in (Cortes & Vapnik, 1995):

$$y_i \left(\boldsymbol{w}_i \cdot \boldsymbol{x}_i + b\right) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad i = 1, \ldots, l \tag{2.46}$$

In this case, the solution is found by minimizing the VC–dimension and an upper bound on the empirical risk, namely the number of training errors. Thus the quantity to minimize is:

$$\min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \quad \frac{1}{2} \|\boldsymbol{w}\|^2 + C \sum_{i=1}^{l} \xi_i \tag{2.47}$$

In analogy to Eq. 2.36, finding a minimum of the first terms actually means minimizing the VC–dimension of the class of functions under consideration. On the other hand, the second term $\sum_{i=1}^{l} \xi_i$ is an upper bound on the number of the misclassifications on the training set, thus finding a minimum for it actually results in minimizing the empirical risk. In this context, the regularization constant $C > 0$ determines the trade off between the complexity term and the empirical risk.

As in the linearly separable case, the lagrangian multipliers are obtained by solving the following quadratic problem:

$$\max_{\alpha} \quad \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,k=1}^{l} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j \tag{2.48}$$

subject to:

$$0 \leq \alpha_i \leq C, \quad i = 1, \ldots, l \tag{2.49}$$

$$\textstyle\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{2.50}$$

Thus the only difference from the separable case is that the lagrangian multipliers are upper bounded by the constant $C > 0$.

According to the Karush–Kuhn–Tucker (KKT) conditions (Lasdon, 1970)—which state necessary conditions for a set of variables to be optimal for an optimization problem—only those lagrangian multipliers $\alpha_i$, $i = 1, \ldots, l$ corresponding to a training pattern $\boldsymbol{x_i}$ which is on the margin or inside the margin area are non–zero. In fact they assert that:

$$\alpha_i = 0 \quad \Rightarrow \quad y_i f(\boldsymbol{x}_i) \geq 1 \quad \text{and} \quad \xi_i = 0 \tag{2.51}$$

$$0 < \alpha_i < C \quad \Rightarrow \quad y_i f(\boldsymbol{x}_i) = 1 \quad \text{and} \quad \xi_i = 0 \tag{2.52}$$

$$\alpha_i = C \quad \Rightarrow \quad y_i f(\boldsymbol{x}_i) \leq 1 \quad \text{and} \quad \xi_i \geq 0 \tag{2.53}$$

These considerations reveals a fundamental property of SVM, namely that the solution found is sparse in $\boldsymbol{\alpha}$. This is crucial for computational times, since sparsity guarantees that the expansion discussed in Eq. 2.41 is calculated on the restricted number of patterns $\boldsymbol{x}_i$ corresponding to $\alpha_i > 0$, also known as *support vectors*.

The KKT conditions are also useful in order to compute the threshold $b$ in Eq. 2.45. In fact, from Eq. 2.52, it follows that:

$$y_i \left( \sum_{j=1}^{l} \alpha_j y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j + b \right) = 1 \tag{2.54}$$

and by averaging on the training patterns it could be possible to extrapolate a stable solution for $b$:

$$b = \frac{1}{|I|} \sum_{j \in I} \left( y_i - \sum_{i=1}^{l} \alpha_j y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j \right) \tag{2.55}$$

where $I = \{i : 0 < \alpha_i < C\}$

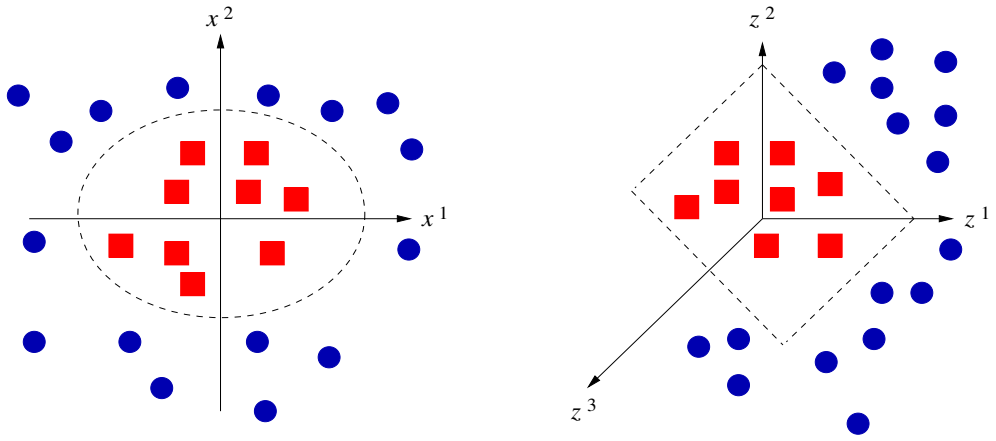



**Figure 2.16:** Non–linearly separable patterns in two–dimensions (left). By re–mapping them in the three–dimensional space of the second order monomials (right) a linear hyperplane separating those patterns can be found.

### 2.2.4 Non–linear SVM

SVM can afford more complex decision functions by re–mapping input patterns onto a higher dimensional space in which the separation between the two classes can be performed by a hyperplane:

$$\Phi : \mathbb{R}^n \rightarrow \mathcal{H} \tag{2.56}$$

$$\boldsymbol{x} \rightarrow \Phi(\boldsymbol{x}) \tag{2.57}$$

Suppose for example that some non–linearly separable patterns are given in two dimensions, as shown on the left picture of Fig. 2.16. By re–mapping them onto the three–dimensional space of the second order monomials:

$$\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3 \tag{2.58}$$

$$\left(x^1, x^2\right) \rightarrow \left(\left(x^1\right)^2, \sqrt{2}x^1x^2, \left(x^2\right)^2\right) = \left(z^1, z^2, z^3\right) \tag{2.59}$$

a linear hyperplane separating those patterns can be found, as shown on the right picture of Fig. 2.16.

The SVM optimization problem involves only the dot products among the training patterns $\boldsymbol{x}_i$, as it is evident from Eq. 2.45. Therefore, the non–linear mapping $\Phi : \mathbb{R}^n \rightarrow \mathcal{H}$ that maps the patterns $\boldsymbol{x}_i$ onto the new space—generally

an Hilbert space—does not need to be given explicitly. It is however necessary to specify the dot product of any of the two images $\Phi(\boldsymbol{x})$ and $\Phi(\boldsymbol{y})$ in $\mathcal{H}$ through a *kernel function* $K$ defined over $C \times C$, where $C$ is a compact subset of $\mathbb{R}^n$ including the training and test patterns:

$$K(\boldsymbol{x}, \boldsymbol{y}) \equiv \Phi(\boldsymbol{x}) \cdot \Phi(\boldsymbol{y}) \tag{2.60}$$

In order to assure that the above definition is well posed, $K$ must satisfy the Mercer's conditions, see (Mercer, 1909). More specifically, $K(\boldsymbol{x}, \boldsymbol{y})$ must be symmetric and continuous over $C \times C$. Furthermore the integral operator over $L^2(C)$:

$$\int_C K(\boldsymbol{x}, \boldsymbol{y}) f(\boldsymbol{x}) d\boldsymbol{x} \tag{2.61}$$

must be positive, namely:

$$\int_{C \times C} K(\boldsymbol{x}, \boldsymbol{y}) f(\boldsymbol{x}) f(\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y} \tag{2.62}$$

for all $f \in C^2$. Once the conditions discussed above are satisfied, it is possible to find a mapping $\Phi$ of the input patterns onto the Hilbert space $\mathcal{H}$ such that Eq. 2.60 defines a scalar product in $\mathcal{H}$. The MMH in $\mathcal{H}$ can thus be written in terms of the input patterns in $\mathbb{R}^n$, giving the following expression for the decision function:

$$f(\boldsymbol{x}) = sign\left(\sum_{i=1}^{l} \alpha_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b\right) \tag{2.63}$$

The only difference from the MMH described in Eq. 2.45 for the linear separable case is that here the dot products $\boldsymbol{x} \cdot \boldsymbol{x}_i$ in $\mathbb{R}^n$ are substituted by the value $K(\boldsymbol{x}, \boldsymbol{x}_i)$ of the kernel function. Common kernel functions are:

- Polynomial kernel of degree $d$:

$$K(\boldsymbol{x}, \boldsymbol{y}) = (\gamma \boldsymbol{x} \cdot \boldsymbol{y} + r)^d \tag{2.64}$$

- Radial basis kernel:

$$K(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\gamma \|\boldsymbol{x} - \boldsymbol{y}\|^2\right) \tag{2.65}$$

- Sigmoidal kernel:

$$K(\boldsymbol{x}, \boldsymbol{y}) = \tanh(\gamma \boldsymbol{x} \cdot \boldsymbol{y} + r) \tag{2.66}$$

where $\gamma$, $r$ and $d$ are kernel parameters selected by the user.

## 2.3 Recursive Feature Elimination

A challenging task for pattern classification consists in trying to reduce the dimensionality $n$ of the feature space, by finding a restricted number of features yielding good classification performances. A lot of work has been done in that direction in the last years, as for example (Kohavi & John, 1997; Kearns *et al.*, 1997). The main reason is probably that feature elimination is fundamental in order to reduce the computational times required to solve pattern classification problems and in some cases also to improve the classification performances. The so–called *curse of dimensionality* from statistics theory, in fact, asserts that the difficulty of an estimation problem increases drastically with the dimension $n$ of the space, since—in principle—exponentially many patterns are required in order to sample the space properly.

In the following, new aspects of the applicability of SVM in knowledge discovery and data mining will be discussed. In Section 2.2, in fact, SVM was introduced as a tool for the solution of pattern classification problems. Here it will be demonstrated that SVM is also very effective for discovering informative attributes of the data set, namely critically important features. To this purpose, first an overview of some feature ranking methods—and in particular of the so–called Optimal Brain Damage (OBD)—will be drawn. Feature ranking is, in fact, the first task to be addressed toward the elimination of unimportant features and in particular the OBD method forms the basis for using linear SVM in such a context. Then, an introduction to feature elimination strategies and in particular to Recursive Feature Elimination (RFE) will be outlined. Finally, it will be demonstrated how RFE can be implemented by using SVM in its very general, non–linear formulation. This algorithm is usually known as SVM–RFE algorithm. The notation followed is mainly derived from (Guyon *et al.*, 2002).

### 2.3.1 Feature–ranking

The main idea behind feature ranking methods is to define the importance of each single feature according to its contribute to the learning machine predictive accuracy. The final aim is to obtain a ranked list of features, from which those having an important contribute may be selected, whereas those having a small contribute can be discarded. This permits to eliminate all those features which are useless for discrimination purposes, or at least represent noise.

Several methods evaluating how well an individual feature contributes to the separation of the two classes have been described in literature. For example, in (Golub *et al.*, 1999) the following correlation coefficient has been used as ranking criterion:

$$r^i = \frac{\mu^i(p^a) - \mu^i(n^a)}{\sigma^i(p^a) + \sigma^i(n^a)} \tag{2.67}$$

where $\mu^i$ and $\sigma^i$ are respectively the mean and the standard deviation of the feature $i$ for all the patterns whose attended class is positive ($p^a$) or negative ($n^a$). Large positive $r^i$ values indicate strong correlation with class $p^a$, whereas large negative $r^i$ values indicate strong correlation with class $n^a$. Then, by selecting an equal number of features with positive and with negative correlation coefficients, it is possible to represent both the two classes. Other approaches—as the one described in (Furey *et al.*, 2000)—have used the absolute value $\left|r^i\right|$, whereas others—as the one described in (Pavlidis *et al.*, 2001)—have used the coefficient:

$$r^i = \frac{\left(\mu^i(p^a) - \mu^i(n^a)\right)^2}{\left(\sigma^i(p^a) + \sigma^i(n^a)\right)^2} \tag{2.68}$$

An important drawback characterizing all those kinds of feature ranking methods based on correlation coefficients, however, rely on the implicit orthogonality assumptions that they make. In fact, each correlation coefficient $r^i$ is computed by using only the informations on that single feature, thus without taking into account the mutual informations between features. This is, of course, one major problem, since features are typically correlated.

In order to overcome that problem, it is necessary to work with *multivariate learning machines*, namely learning machines which are optimized during the training phase to handle multiple features simultaneously. SVM—for example—is a typical multivariate learning machine. In the context of multivariate learning machines, however, features are no more ranked according to simple coefficients as the ones described above, but rather according to their influence on the change of a cost function $J$. Here $J$ is the function that the learning machine has to minimize in order to solve the classification problem, see (Kohavi & John, 1997). In the specific case of linear separable SVM, for example, the cost function $J$ which has to be minimized—under the conditions represented by Eq. 2.35—is:

$$J = \frac{1}{2}\|\boldsymbol{w}\|^2 \tag{2.69}$$

The main idea behind this technique is thus to compute the changes $\Delta J^i$ in the cost function caused by removing each feature $i = 1, \ldots, n$ and then to rank all the features accordingly. In particular, the smaller the change $\Delta J^i$ in the cost function is, the lower the contribution of the feature $i$ to the learning machine predictive accuracy is. On the other hand, the larger the change $\Delta J^i$ is, the higher its contribution is.

In order to calculate those changes in the cost function of linear discriminant learning machines, (LeCun *et al.*, 1990) suggests for example to approximate $\Delta J^i$ by expanding it in Taylor series to the second order. At the optimum of $J$, the first order term can be therefore discarded, thus obtaining:

$$\Delta J^i \approx \frac{1}{2} \frac{\partial^2 J}{\partial (w^i)^2} (\Delta w^i)^2 \tag{2.70}$$

where $\boldsymbol{w} = (w^1, \ldots, w^n)$ is the $n$–dimensional vector of weights and the change in weight $\Delta w^i = w^i$ corresponds to the removal of the feature $i$. This method—also known as Optimum Brain Damage (OBD)—thus suggests that for linear discriminant learning machines whose cost function $J$ is a quadratic function of the weights $w^i$—such as linear SVM—features can be simply ranked according to the value $(w^i)^2$.

### 2.3.2 Recursive elimination of ranked features

The criterion discussed in Eq. 2.70—and more in general the use of the change $\Delta J^i$ in the cost function as ranking criterion—are actually concerned with the removal of one single feature at a time. However, in order to obtain a small subset of relevant features—in particular when starting with a huge number of them—it may be necessary to remove more that one feature at a time. This problem can be overcome by using the following very general iterative procedure, known as Recursive Feature Elimination (RFE):

1. Train the learning machine, namely optimize its weights $w^i$ by minimizing the cost function $J$

2. Compute the ranking criterion $\Delta J^i$ for each feature $i$

3. Remove a subset of features characterized by the smallest ranking values

It is evident that—with this approach—it is possible to quickly converge at a small subset of relevant features by removing chunks of features each time and by re–training the learning machine after each elimination. Notice that, in order to find the best trade off between efficiency and preservation of classification accuracy, it may be convenient to remove larger subsets of features during the first steps of the iterative procedure, then recursively decreasing the dimensions of those chunks down to one feature at a time for the last steps.

It is well worth noticing that if features are removed one at a time, there is also a corresponding feature ranking. However, when removing chunks of features, those that are top ranked—namely eliminated last—are not necessarily the ones that are individually most relevant. Only taken together as a subset they are optimal.

### 2.3.3 SVM–RFE

The method of recursively eliminating features on the basis of the smallest change in cost function described above can be used in principle with every multivariate learning machine. In particular, it can be used with linear SVM, by using the OBD approximation previously sketched. However, it can be also extended to non–linear SVM and to all kernel methods in general.

As already discussed, in fact, the cost function that non–linear SVM has to minimize under specific conditions is:

$$J = \frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{H}\boldsymbol{\alpha} - \boldsymbol{\alpha}^T \mathbf{1} \tag{2.71}$$

where $\boldsymbol{H}$ is the matrix with elements $y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $\mathbf{1}$ is an $l$–dimensional vector of ones. In order to compute the change in the cost function by removing the feature $i$, one has to compute the matrix $\boldsymbol{H}(-i)$, where the notation $(-i)$ means that the feature $i$ has been removed. The variation in the cost function $J$ is thus:

$$\Delta J^i = \frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{H}\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{H}(-i)\boldsymbol{\alpha} \tag{2.72}$$

where no change in the value of $\boldsymbol{\alpha}$ has been assumed. The feature—or chunk of features—corresponding to the smallest $\Delta J^i$ is then removed, SVM is trained once again with the new smaller set of features. The procedure can thus be iterated feature after feature—or chunk after chunk—until a reasonable small number of features survives or the performances of the classifier start degrading.

Notice that, in the linear case, calculating the change in the cost function is particularly straightforward. In fact, being:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i \cdot \boldsymbol{x}_j \tag{2.73}$$

and:

$$\boldsymbol{\alpha}^T \boldsymbol{H} \boldsymbol{\alpha} = \|\boldsymbol{w}\|^2 \tag{2.74}$$

then from Eq. 2.72 it results that:

$$\Delta J^i = \frac{1}{2}(w^i)^2 \tag{2.75}$$

This is exactly the results obtained in Eq.2.70 by the OBD method for the specific case of linear SVM.

Chapter 3

# Image Representation

*In this Chapter, an introduction to digital imaging—together with a detailed discussion of some advanced imaging techniques—will be given. The idea is to provide the reader with a clearer picture of the approaches that will be adopted in this work. Specifically, with those adopted in order to find the crop's image representation which supplies the best classification performance. It is well worth reminding, in fact, that the novel technique pursued in the experimental part of this work consists of classifying the entire pixels of each mammographic crop—or at least a transformed version of them—thus without extracting any a priori information. In this sense, the classification features used here are different according to the image representation chosen for the crop, as for example a raw pixel–based representation or a transformed one. To this aim, Section 3.1 will introduce some very basic aspects of digital image representation and enhancement related to the pixel–based image representation of an image. Section 3.2 will deal with one of the most famous image processing techniques of the last decades—namely the wavelet transform—which has in its multi–resolution nature a powerful tool. In Section 3.3, a different multi–resolution technique—providing also a very high orientation selectivity—will be discussed, namely steerable filters. Finally, in Section 3.4, a new born rank–based technique—introduced for the first time in 2002 for face detection problems—will be discussed. This very promising technique—never been applied to imaging problems different from face detection, such as for instance medical imaging—is known as ranklet transform.*

## 3.1 Digital Image Representation Using Pixels

In this Section, a brief overview of the fundamental aspects of digital image representation by means of pixels will be first given. Some considerations about well known image processing techniques such as image resizing and histogram equalization will be then drawn. Notice, in particular, that this does not want to be a complete survey of digital imaging, but rather an overview of the specific techniques that will be used in the rest of this work. (Gonzalez & Woods, 1992) will be mainly followed.

### 3.1.1 Basic concepts

Images can be thought of as two–dimensional functions $f(x, y)$, whose value or intensity at spatial coordinates $(x, y)$ is a positive scalar quantity proportional to the energy radiated by its physical source. In particular, $f(x, y)$ must be non–zero and finite. Monochromatic images are those images whose values are said to span the gray scale. Specifically, the intensity of a monochromatic image at any coordinates $(x, y)$ is referred to as the *gray level* of the image at that point, namely:

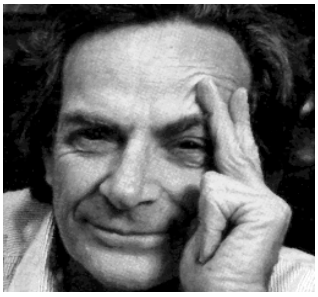
$$f(x, y) = l \tag{3.1}$$

Being $f(x, y)$ non–zero and finite, it turns out that the gray level $l$ must lie in the interval $[L_{\min}, L_{\max}]$, also known as the *gray scale*:

$$L_{\min} \leq l \leq L_{\max} \tag{3.2}$$

Typically, this interval is shifted to the interval $[0, L-1]$, where $l = 0$ is considered black and $l = L - 1$ is considered white on the gray scale. Intermediate values represent all the shades of gray varying from black to white.

Images originally come as continuous functions of both coordinates and intensity. In order to digitize them, they have to be sampled in both coordinates and intensity. The former operation is usually referred to as *sampling*, whereas the latter as *quantization*. In particular, in order to sample an image, equally samples along both $x$ and $y$ axis must be taken. The set of the discrete locations thus obtained represents the sampled function. Notice that sampling is determined by the sensor arrangement used to generate the image. On the other hand, in order to quantize an image, its intensity values must be converted into digital quantities. This means that the gray scale must be divided into a finite number of gray–levels, then each sampled discrete location is assigned a gray level according to its mean intensity value.



$$\longrightarrow \begin{pmatrix} 75 & 97 & \dots & 0 \\ 75 & 81 & \dots & 0 \\ 87 & 81 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 147 & 171 & \dots & 237 \\ 152 & 205 & \dots & 237 \\ 152 & 225 & \dots & 232 \end{pmatrix}$$

**Figure 3.1:** A $256 \times 256$ pixels picture of R. Feynman (left). Some elements of its matrix notation (right).

Sampling and quantization allow to represent digital images as matrices. This is fundamental, since matrix notation forms the basis of the mathematical framework describing the image processing theory. In particular, an $M \times N$ matrix can be expressed in the following compact form:

$$f(x,y) = \begin{pmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(0,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-2,0) & f(M-2,1) & \dots & f(M-2,N-1) \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{pmatrix} \tag{3.3}$$

In such a context, the right side of Eq. 3.3 is by definition a digital image, whereas each one of its elements is called *picture element* or—more conventionally—*pixel*. Fig. 3.1 shows an image and some elements of its matrix notation.

Sampling is the most important factor determining the *spatial resolution* of an image, namely the smallest discernible spatial detail in an image. On the other hand, quantization determines the *gray–level resolution* of an image, namely the smallest discernible change in gray level. Notice that, in choosing the number of samples used to generate the digital image, a considerable discretion there exists. Contrarily, due to hardware considerations, the number of gray levels is usually an integer power of 2. A typical number of gray levels is 8 bits, or at least 16 bits for specific applications requiring high gray–level resolution. In such a context, a digital image with $L$ gray levels and size $M \times N$ is commonly referred to as a digital image having spatial resolution of $M \times N$ pixels and gray–level resolution of $L$ levels.

### 3.1.2 Image resizing

*Image resizing* is a technique which allows to manipulate images by changing their size. It is mainly based on two separate steps. First, on the creation of new pixel locations, by laying a new imaginary grid of pixels over the original image. Second, on the assignment of gray levels to the new locations. The fastest way to perform the gray levels assignment is the *nearest neighbor*—or *linear*—*interpolation*. In practice, each new pixel in the grid is assigned the gray level of the closest pixel in the original image. This approach has however the undesirable characteristic of producing a checkerboard effect in the resized image. In order to overcome this problem, *bi–linear interpolation* can be used, namely a more elaborated technique producing less evident artifacts than those of linear interpolation. In particular, it uses the four nearest neighbors of the new pixel in the grid to determine its gray level. Suppose for example that $(x, y)$ represent the coordinates of a point on the imaginary grid previously introduced and let $v(x, y)$ denote the gray level assigned to it. Bi–linear interpolation assigns the gray level in the following way:

$$v(x, y) = ax + by + cxy + d \tag{3.4}$$

Here the four coefficients $a, b, c, d$ are determined from the four equations in four unknowns that can be written using the four nearest neighbors of point $(x, y)$.

Fig. 3.2 shows the effects of image resizing on the same R. Feynman's picture previously depicted in Fig. 3.1. In particular, left column shows the original $256 \times 256$ pixels image resized by means of linear interpolation to $64 \times 64$ and $32 \times 32$ pixels. Although images b) and c) show the different dimensional proportions with respect to the original image a), they do not give information about the effects on the image quality resulting after the resizing step. To this aim, it is necessary to bring the resized images up to the original $256 \times 256$ pixels size by *pixel replication*. This means that, for example, in order to double the size of an image each column is duplicated—thus doubling the horizontal direction—then each row. The same procedure is used in order to enlarge the image by any integer number of times, such as triple, quadruple and so forth. For example, middle and right columns in Fig. 3.2 show the original $256 \times 256$ pixels image resized by means of—respectively—linear and bi–linear interpolation to $64 \times 64$ and $32 \times 32$ pixels, then rearranged by means of pixel replication. The improvements in overall appearance are evident in the former case, in particular when comparing images e) and h).

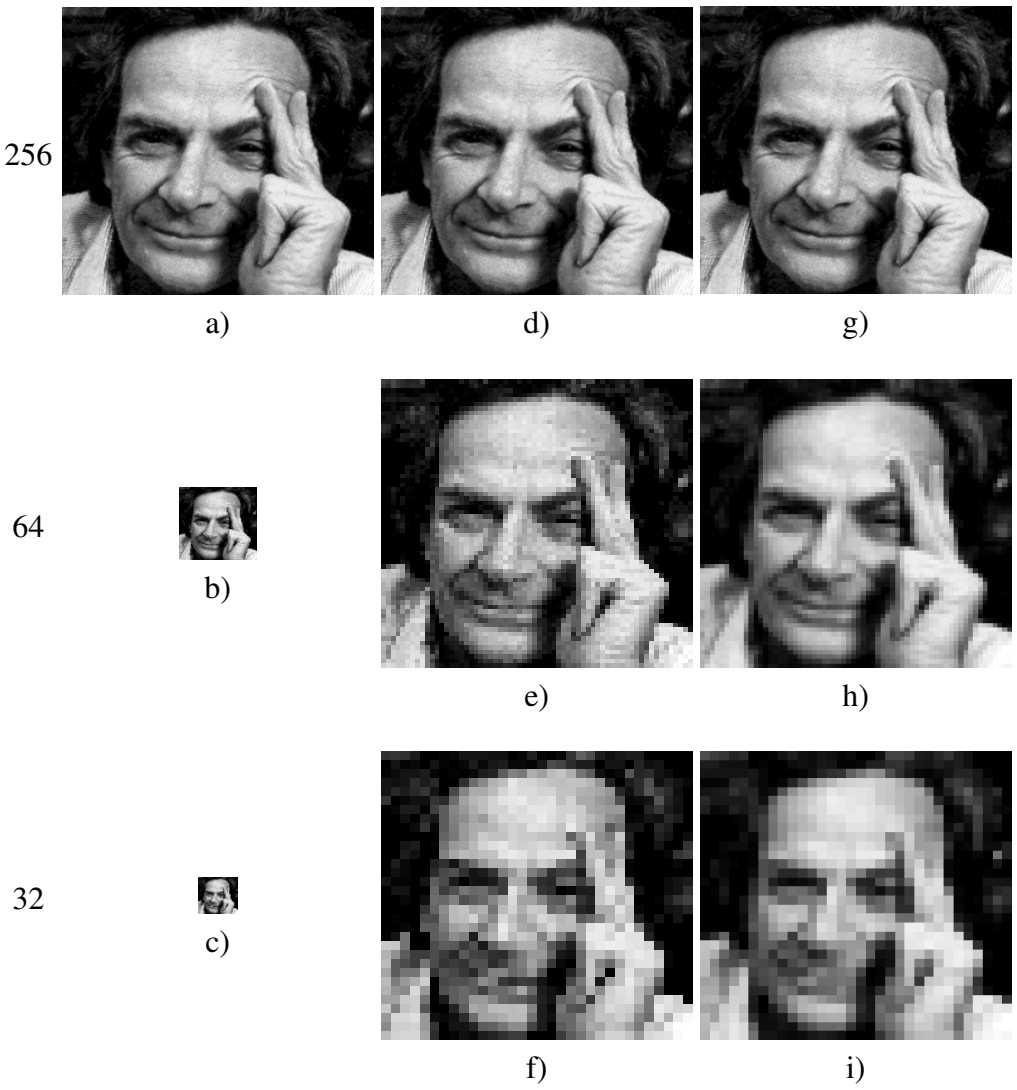



**Figure 3.2:** Image resizing from $256 \times 256$ down to $64 \times 64$ and $32 \times 32$. Left column: linear interpolation. Middle column: linear interpolation plus pixel replication. Right column: bi–linear interpolation plus pixel replication.

### 3.1.3 Image histogram equalization

The histogram of an image with gray levels in the range $[0, L-1]$ is a discrete function $h(r_k) = n_k$, where $r_k$ is the $k^{th}$ gray level and $n_k$ is the number of pixels in the image having gray level $r_k$. It is possible to notice that in a dark image the components of the histogram are concentrated in the low—dark—side of the gray scale. On the other hand, in a bright image those components are biased toward the high—bright—side of the gray scale. See, as an example, images and corresponding histograms of the first two rows in Fig. 3.3. It is possible to notice also that an image with a low contrast has a histogram both narrow and concentrated toward the middle of the gray scale. Conversely, in an image with a high contrast the components of the histogram cover a broad range of the gray scale and the distribution of the pixels is approximately uniform. See, as an example, images and corresponding histograms of the last two rows in Fig. 3.3.

The above observations suggest that when dealing with an image whose pixels tend to occupy the entire range of possible gray levels and to be distributed uniformly, it is reasonable to conclude that it will have an appearance of high contrast and will exhibit a large variety of gray tones. The net effect will be an image showing a great deal of gray–level detail and having a high dynamic range. For these reasons—and since the advantages of having gray–level values that cover the entire gray scale are evident from Fig. 3.3—a transformation function that can automatically achieve this effect has been developed.

In order to spread the histogram of the input image so that the levels of the histogram–equalized image span a fuller range of gray scale, it can be demonstrated that the following transform must be applied:

$$s_k = \sum_{j=0}^{k} \frac{n_j}{n} \quad k = 0, 1, 2, \ldots, L-1 \tag{3.5}$$

where $n$ is the total number of pixels in the image under consideration. The transformed image is thus obtained by mapping each pixel with level $r_k$ in the input image into a corresponding pixel with level $s_k$ in the output image via Eq. 3.5. This transform is usually referred to as *histogram equalization*. Notice that the method just derived is completely automatic, namely given an image the process of histogram equalization consists simply of implementing Eq. 3.5 which is based on information that can be extracted directly from the given image, thus without any further parameter specifications. For more details on this topic see (Gonzalez & Woods, 1992; Castleman, 1996).

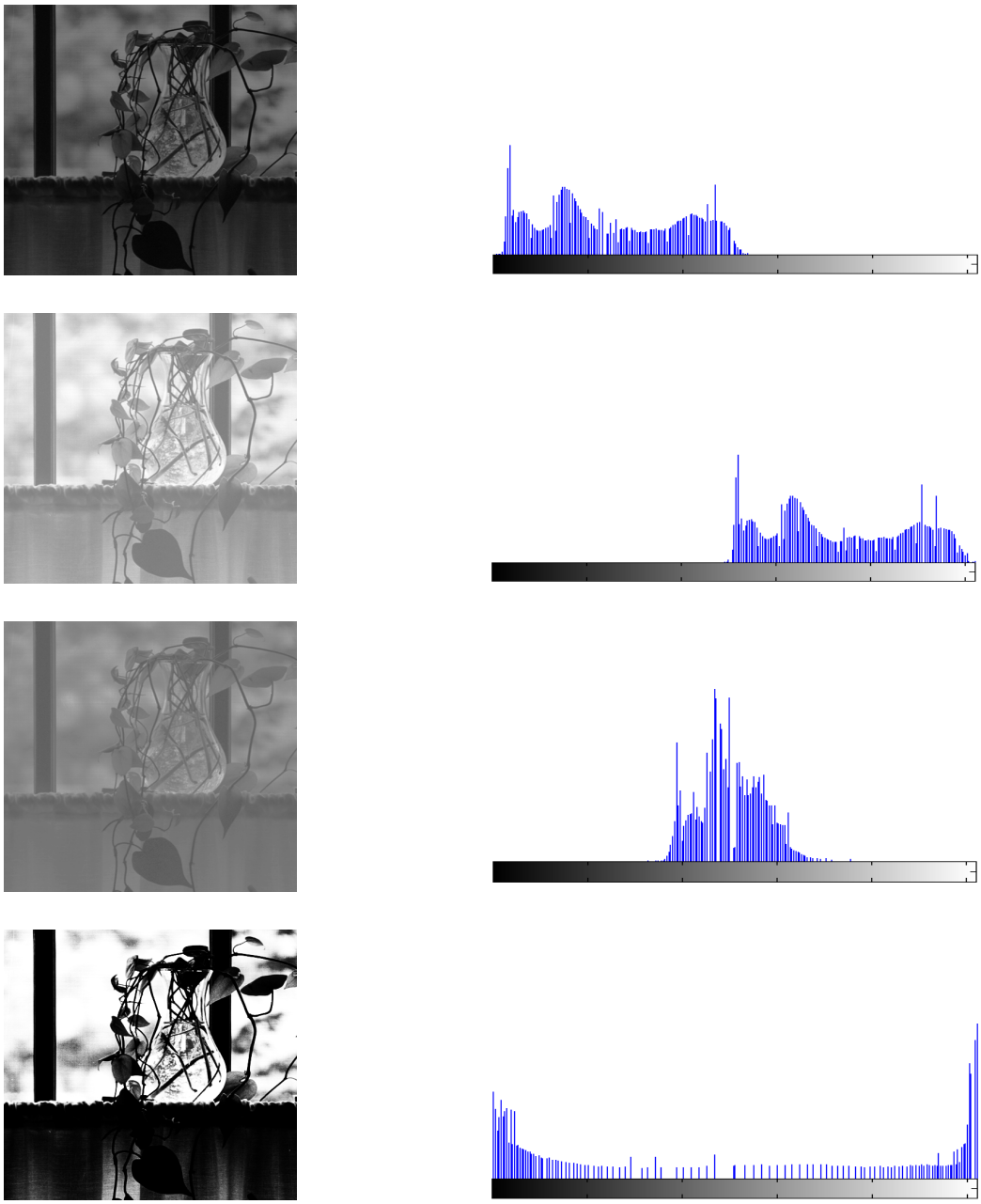


**Figure 3.3:** Dark image (first row). Bright image (second row). Low–contrast image (third row). High–contrast image (fourth row).

## 3.2 Wavelets

Although the Fourier transform has been the mainstay of transform–based image processing since the late 1950s, the theory and applications of a more recent transformation—known as *wavelet transform*—have undoubtedly dominated the scientific publications in mathematical, engineering and related fields throughout the last decades. In particular, the success of the wavelet transform in the image processing community is mainly due to its efficiency and effectiveness in dealing with almost the totality of the most important image processing tasks, such as for example image analysis, compression, de–noising and so on.

The main difference between those two transformations relies on how they represent the signals under analysis. The Fourier transform, in fact, expresses signals as a weighted sum of basic trigonometric functions, namely sinusoids. However, this representation has one major drawback. Sinusoids have perfect compact support in frequency domain, but not in time domain. In particular, they stretch out to infinity in time domain and therefore they cannot be used to approximate *non–stationary signals*, namely those signals whose spectral content changes in time. In this sense, the Fourier representation only provides the spectral content of the signal, with no indication about the time localization of its spectral components. Unfortunately, non–stationary features as drift, trends, abrupt changes and so forth are present in almost all images, other than being generally the most important features to characterize an image.

In order to overcome this problem, the wavelet transform uses compactly supported functions of limited duration in time and frequency. This allows the wavelet transform to provide informations about both when and at what frequencies a specific event occurs. In particular, the precision for this information is limited by the compact support of the wavelet functions used as basis. Other than the above discussed capability of localize an event in both time and frequency, wavelets are so popular also because they represent the foundation of a powerful approach to signal processing called *multi–resolution theory*. This approach unifies different techniques from various research fields, such as sub–band coding from signal processing, quadrature mirror filters from digital speech recognition and pyramidal image processing. In particular, the importance of this approach is concerned with the possibility to analyze signals and images at different scales, thus allowing to search for specific features at a specific resolution, namely from the finest to the coarsest ones. This is also much more appreciable if considered that this approach is easily and quickly implementable as a bank of digital filters.

In the following, a description of the wavelet transform will be given. First, a very brief overview of the historical genesis and development of wavelet theory will be sketched. Second, the wavelet transform will be introduced in its filter banks formulation for both the mono–dimensional and the bi–dimensional case. Finally, the specific case of the Haar wavelet transform will be discussed in detail. In particular, both the mono–dimensional and the bi–dimensional cases will be described, together with its *overcomplete* formulation.

### 3.2.1 Historical perspective

Although the wavelet transform is quite a recently developed approach—and the interest of the image processing community on this topic started only in the late 1980s—it is very difficult to keep up with the number of publications, theses and books devoted to this subject and to its variegate applications. However it is worth trying to individuate the works that most influenced its development.

The first wavelet basis functions were discovered by A. Haar, who introduced in 1910 the functions that are now called *Haar wavelets*. These functions consist simply of a short positive pulse followed by a short negative pulse, as discussed in (Haar, 1910).

For many, however, the real starting point of the modern history of wavelets coincides with the investigations conducted by J. Morlet and A. Grossman on the analysis of seismic signals by means of small and oscillatory window functions with compact support both in time and in frequency. In particular, in (Grossmann & Morlet, 1984) they discussed the development of such functions and associated to them the term *wavelets*. From the results obtained by J. Morlet and A. Grossman, Y. Meyer started later developing wavelets with better localization properties. In particular, in (Meyer, 1987) he described the construction of orthogonal wavelet basis functions with very good time and frequency localization.

The transition from continuous to discrete signal analysis was mainly due to I. Daubechies—a graduate student of A. Grossman—and S. Mallat who published in the late 1980s two fundamental papers establishing a solid mathematical footing for wavelet theory. In particular, I. Daubechies developed in (Daubechies, 1988) the mathematical framework for discretization of time and scale parameters of the wavelet transform. On the other hand, after one year S. Mallat discussed in (Mallat, 1989) the idea of multi–resolution analysis for discrete wavelet transform which turned out to be the corner stone of modern wavelet theory.

The last years have seen an increasing comprehension and development of the theoretical aspects underlying the wavelet mathematical framework, together with an explosion of wavelet applications in both signal and image processing. In particular, from a more theoretical point of view, the last decades have assisted to a repeated search for other wavelet basis functions with different properties and modifications of the multi–resolution algorithms. On the other hand, from a more experimental point of view, they have assisted a large number of wavelet applications coming up, such as image and signal analysis, compression, de–noising, feature and self–similarity detection and much more. Interestingly, all those applications span over several different research fields such as mathematics, physics, economics, seismology, medical imaging, computer graphics, neurophysiology and so on.

### 3.2.2 Wavelets as filter banks

As already anticipated, wavelet theory unifies several research fields, such as sub–band coding, quadrature mirror filters and image pyramids. This variety allows to approach wavelet theory from different directions, in particular following a pure mathematical path or walking along a more applicative way based on filter banks.

The former approach is generally more suited to appreciate the deep theoretical aspects underlying this transform and to recognize the efforts made in the years in order to put in a rigorous mathematical form such an applicative technique. In particular, this approach results really appropriate in order to understand how the transition from the original *continuous wavelet transform* to its discretized version—namely *discrete wavelet transform*—has been carried out, thus giving the concrete opportunity to apply wavelet analysis to the solution of real–world problems.

On the other hand, the applicative approach is definitely more suited when dealing with wavelet applications in signal an image processing, as for the present work. In fact, following the filter–based approach, the mathematical footing of the wavelet transform can be almost by–passed, thus directly introducing the operative definition of discrete wavelet transform as a bank of digital filters. This approach is particularly appropriate in order to deal with the most important characteristic of the wavelet transform, namely its multi–resolution implementation. The multi–resolution wavelet transform, in fact, can be easily implemented by means of a set of recursive nested filter banks, as it will be discussed in the following.

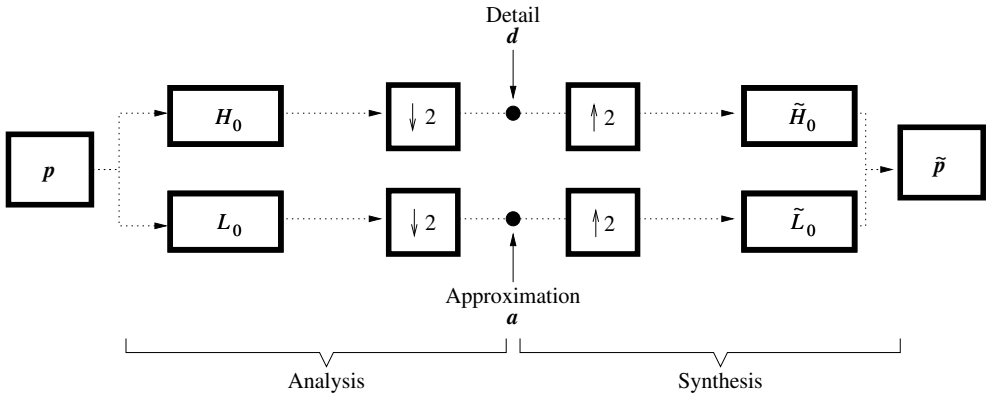



**Figure 3.4:** Discrete wavelet analysis and synthesis in one dimension.

Typical pure mathematical approaches to wavelet theory are adopted for example in (Daubechies, 1992; Meyer, 1992; Vidakovic, 1996; Mallat, 1998). On the other hand, more applicative paths are followed for example by (Strang & Nguyen, 1996; Hubbard, 1996; Stollnitz *et al.*, 1996; Aldroubi & Unser, 1996).

### Discrete wavelet transform in one dimension

The *Discrete Wavelet Transform* (DWT) in one dimension represents the first topic to discuss in order to have an introductory idea of the wavelet transform. In particular, given a one–dimensional discrete–time signal $\boldsymbol{p} = (p_1, \ldots, p_N)$, its discrete wavelet transform—or *wavelet analysis*—is formed through the decomposition of $\boldsymbol{p}$ into the signals $\boldsymbol{a}$ and $\boldsymbol{d}$ via *analysis filters* $H_0$ and $L_0$, as shown in Fig. 3.4. Here $H_0$ is a high–pass filter whose output signal $\boldsymbol{d}$ represents the high frequency or *detail* part of the original signal $\boldsymbol{p}$. On the other hand, filter $L_0$ is a low–pass filter whose output signal $\boldsymbol{a}$ represents the low frequency or *approximation* part of the original signal $\boldsymbol{p}$. Typically, the samples of the approximation $\boldsymbol{a}$ and detail $\boldsymbol{d}$ are referred to as *wavelet coefficients*. All filtering is performed in the time domain by convolving each filter's input with its impulse response $\boldsymbol{l}_0$ and $\boldsymbol{h}_0$, namely its response to a unit amplitude impulse function. Furthermore, in order not to wind up with twice as much data as started, *sub–sampling* units are introduced at the output of each filter. Throwing away every second data point, in fact, they allow to end up with the same number of samples as for the original signal, even though this introduces in the wavelet coefficients a type of error named *aliasing*. For more informations on aliasing see (Strang & Nguyen, 1996).

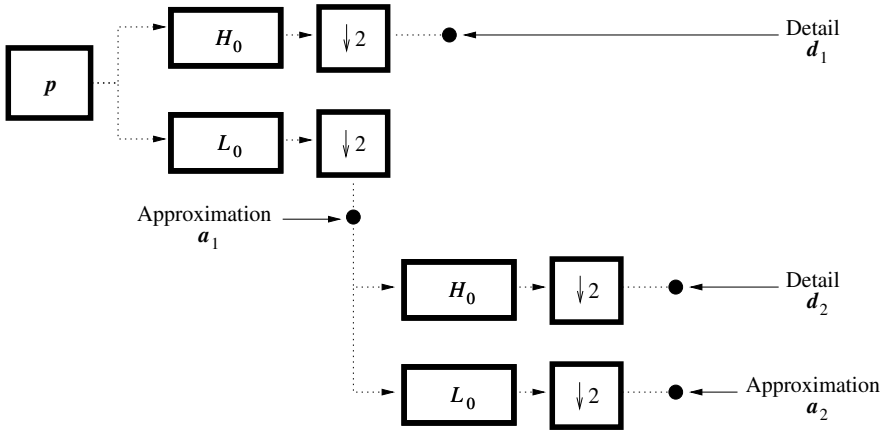



**Figure 3.5:** Multi–resolution discrete wavelet transform in one dimension (2 decomposition levels).

The process of assembling back the approximation $\boldsymbol{a}$ and the detail $\boldsymbol{d}$ is called *Inverse Discrete Wavelet Transform* (IDWT) or *wavelet synthesis*. Here wavelet coefficients are first processed by means of *up–sampling* units which lengthen their input signals by inserting zeros between samples. This step is crucial in order to reconstruct a signal with the same number of samples as the original one. Subsequent recombination of the up–sampled signals via *synthesis filters* $\tilde{H}_0$ and $\tilde{L}_0$ then yields the output signal $\tilde{\boldsymbol{p}}$. In particular, it can be demonstrated that filters $H_0$, $L_0$, $\tilde{H}_0$ and $\tilde{L}_0$ can be designed in such a way that the aliasing effects introduced during the analysis phase cancel out and that no distortion is introduced. In this way, it is possible to perfectly reconstruct the original signal, namely $\boldsymbol{p} = \tilde{\boldsymbol{p}}$. The first set of filters satisfying these constraints were proposed by (Croisier *et al.*, 1976) and the resulting filter bank was called *Quadrature Mirror Filters* (QMF). Later on, several families of filters were introduced, such as for example *Conjugate Quadrature Filters* (CQF), *Orthonormal Filters* (OF) and so on. An example of CQF is given in (Smith & Barnwell, 1986), whereas examples of OF include the Daubechies filters introduced in (Daubechies, 1988) and the Smith and Barnwell filters introduced in (Smith & Barnwell, 1984). Notice, finally, that it can be demonstrated that the convolution of a discrete signal with the impulse response of a digital filter can be easily expressed in terms of matrix products. In particular, with such a choice of filters, the synthesis matrices result being the transposes of the analysis matrices, namely the wavelet transform is *self–inverting*. This is clearly a very appreciable property.

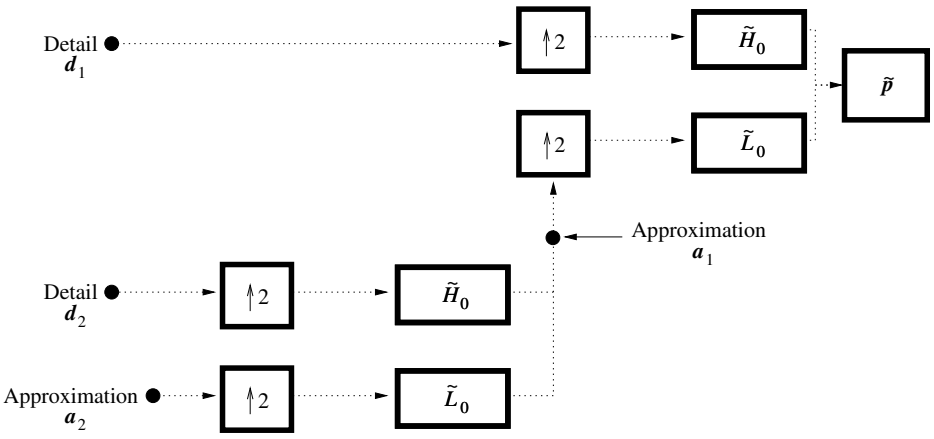



**Figure 3.6:** Multi–resolution inverse discrete wavelet transform in one dimension (2 decomposition levels).

### Multi–resolution discrete wavelet transform in one dimension

As anticipated, the multi–resolution version of the discrete wavelet transform—namely *multi–resolution discrete wavelet transform*—can be implemented by simply iterating the analysis filter bank shown in Fig. 3.4. This allows to create a multi–level structure which enables for computing the wavelet transform at successive resolutions, namely from the finest to the coarsest ones. Fig. 3.5, for example, shows a two–level implementation of the multi–resolution discrete wavelet transform. In particular, the first filter bank splits the original signal into a first–level detail component $\boldsymbol{d}_1$ and into a first–level approximation component $\boldsymbol{a}_1$. The second filter bank, then, splits the first–level approximation component $\boldsymbol{a}_1$ into a second–level detail component $\boldsymbol{d}_2$ and into a second–level approximation component $\boldsymbol{a}_2$.

Similarly to the single–resolution case, an inverse transform for the reconstruction of the original signal can be formulated, see Fig. 3.6. In the specific case of a two–level decomposition, it uses first the second–level approximation $a_2$ and detail $d_2$ to reconstruct the first–level approximation $a_1$. The signal is then reconstructed by using the first–level approximation $a_1$ and detail $d_1$. As one might expect, perfect reconstruction can be achieved by using a set of filters which satisfy the constraints discussed above, namely requiring that the aliasing effects introduced during the analysis phase cancel out and that no distortion is introduced.

It is evident that the two–level implementation discussed here can be easily extended to any number $L$ of decomposition levels, or at least until the approximation and detail components produced by the analysis consist of a single sample. In particular, in the general case, the discrete wavelet transform of the original signal $\boldsymbol{p}$ is comprised of a concatenation of the signals $[\boldsymbol{a}_L, \boldsymbol{d}_L, \boldsymbol{d}_{L-1}, \ldots, \boldsymbol{d}_2, \boldsymbol{d}_1]$, namely a concatenation of the approximation component correspondent to the last decomposition level and all the detail components at all levels. Remarkably, the number of samples of the analyzed signal is equal to that of the original signal. Notice, in particular, that this is due to the analysis at each level of a *sub–sampled* version of the approximation component of the precedent level. The sub–sampling operator, in fact, reduces the number of samples in the approximation component by half level after level. Notice, furthermore, that this strategy is fundamental in order to achieve multi–resolution. Following this way, in fact, at each level the wavelet decomposition is performed on an approximation component whose number of samples—namely resolution—is exactly half that of the precedent approximation component.

### Discrete wavelet transform in two dimensions

The one–dimensional filters discussed above can be used as two–dimensional separable filters for the processing of images. The idea behind this bi–dimensional approach is to analyze an image $\boldsymbol{p} = \boldsymbol{p}_1, \ldots, \boldsymbol{p}_N$—where $\boldsymbol{p}_j = (p_j^x, p_j^y)$ for all $j = 1, \ldots, N$—by first applying the filters along one dimension—for example horizontally, namely along rows—then along the other dimension—thus vertically, namely along columns—as shown in Fig. 3.7. Notice that here the sub–sampling operation is performed twice, namely after each filtering operation. The resulting filtered images—denoted as $\boldsymbol{a}, \boldsymbol{d}^{\mathsf{V}}, \boldsymbol{d}^{\mathsf{H}}, \boldsymbol{d}^{\mathsf{D}}$—are respectively referred to as the approximation, vertical detail, horizontal detail and diagonal detail of the original image. In particular, the approximation gives global informations about the image under analysis, whereas the details measure its intensity variations along different directions. For example, the horizontal detail $\boldsymbol{d}^{\mathsf{H}}$ measures variations along the columns of the image—as for example horizontal edges—the vertical detail $\boldsymbol{d}^{\mathsf{V}}$ along the rows—as for example vertical edges—whereas the diagonal detail $\boldsymbol{d}^{\mathsf{D}}$ along diagonals.

Fig. 3.8 shows the synthesis filter bank reversing the analysis process in two dimensions. As would be expected, the reconstruction algorithm is similar to the one–dimensional case.

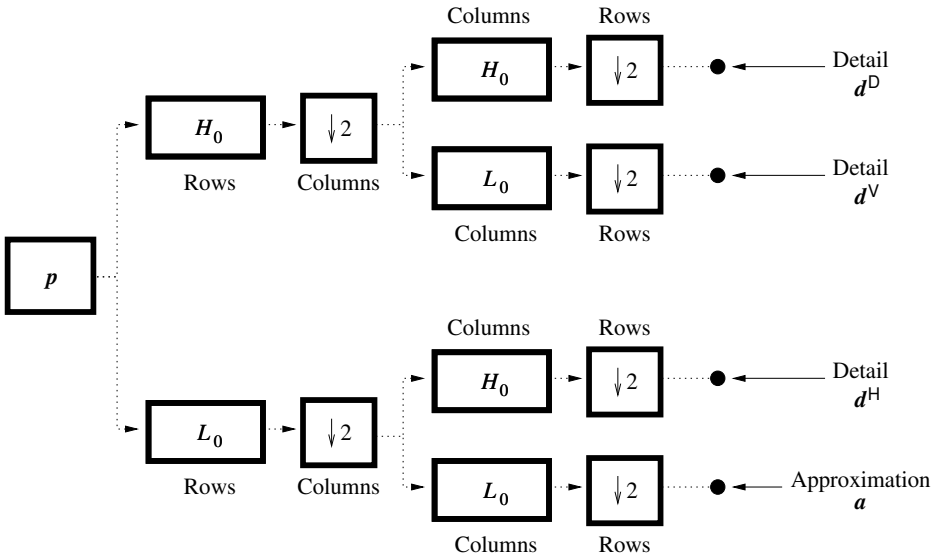



**Figure 3.7:** Discrete wavelet transform in two dimensions.

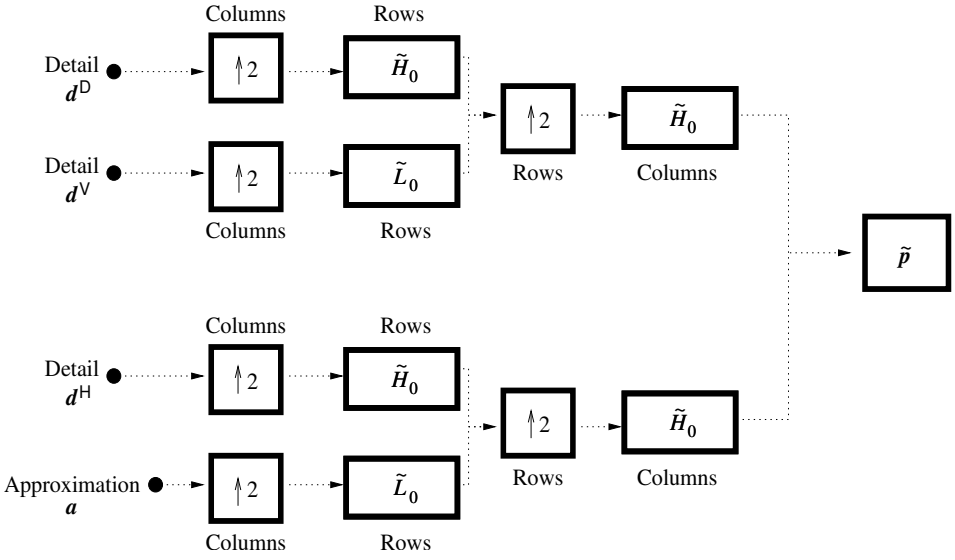



**Figure 3.8:** Inverse discrete wavelet transform in two dimensions.

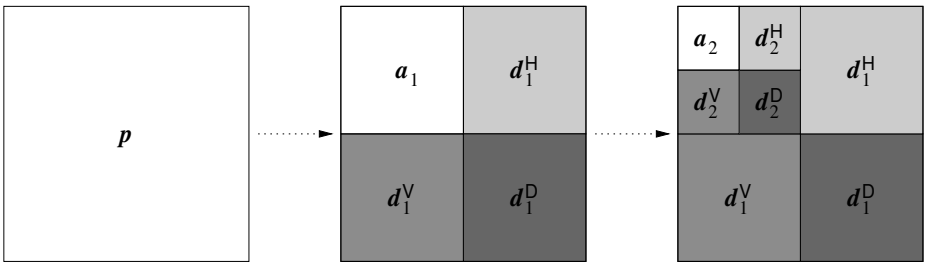



**Figure 3.9:** Multi–resolution discrete wavelet transform in two dimensions (2 decomposition levels). Original image (left). Analyzed image after the first decomposition level (middle). Analyzed image after the second decomposition level (right).

#### Multi–resolution discrete wavelet transform in two dimensions

As for the mono–dimensional case, the transform can be performed at successive resolutions by simply iteratively applying the analysis filter banks described above to the approximations obtained at each level. In this way, the bi–dimensional wavelet transform of the original image $\boldsymbol{p}$ is given by a concatenation of the images $\left[\boldsymbol{a}_L, \boldsymbol{d}_L^{\mathsf{V,H,D}}, \boldsymbol{d}_{L-1}^{\mathsf{V,H,D}}, \ldots, \boldsymbol{d}_2^{\mathsf{V,H,D}}, \boldsymbol{d}_1^{V,H,D}\right]$, namely the concatenation of the approximation component correspondent to the last decomposition level and all the detail components at all the levels. The schematic result of a bi–dimensional wavelet decomposition is shown in Fig. 3.9.

Notice that—due to the sub–sampling operators—also for the multi–resolution discrete wavelet transform in two dimensions the number of pixels of the analyzed image is equal to that of the original one. The inverse transform can be obtained by tying several filter banks as the one shown in Fig. 3.8. Furthermore, it can be demonstrated that—expressing all the procedure in matrix language—the synthesis matrices result being the transposes of the analysis matrices, namely also the bi–dimensional wavelet transform is self–inverting.

### 3.2.3 Haar wavelet transform

The importance of the Haar wavelet transform stems from the fact that its basis functions are the oldest and simplest known orthonormal wavelets, see (Haar, 1910). Furthermore, it will be intensively used in the rest of this work.

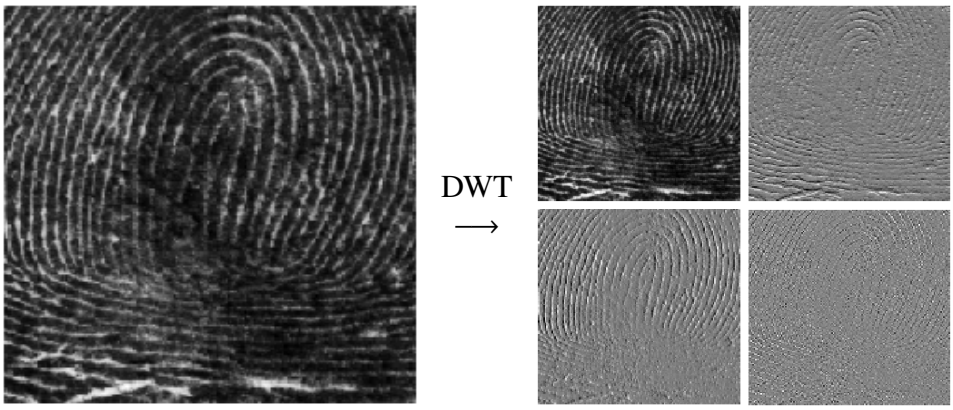



**Figure 3.10:** A two–level discrete Haar wavelet transform in two dimensions applied to a fingerprint image.

### Multi–resolution discrete Haar wavelet transform

The *multi–resolution discrete Haar wavelet transform* is performed by following exactly the same considerations drawn for the general multi–resolution discrete wavelet transform—both in one and two dimensions—but with the following specific choice for $\boldsymbol{l}_0$ and $\boldsymbol{h}_0$, namely the impulse responses of filters $L_0$ and $H_0$:

$$\boldsymbol{l}_0 = \left[+\frac{1}{\sqrt{2}}, +\frac{1}{\sqrt{2}}\right] \tag{3.6}$$

$$\boldsymbol{h}_0 = \left[+\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right] \tag{3.7}$$

All filtering is thus performed by convolving the input signal—or image—with the impulse responses just described. In particular, in order to perform the multi–resolution discrete Haar wavelet transform in one dimension, the scheme described in Fig. 3.5 will be followed, whereas in order to perform the multi–resolution discrete Haar wavelet transform in two dimensions, that in Fig. 3.7.

As an example, Fig. 3.10 shows a one–level discrete Haar wavelet transform in two dimensions of a fingerprint image. It is evident here that details account for different structures of the image, namely horizontal, vertical and diagonal ridges. Notice, furthermore, that the original image size is $296 \times 296$ pixels, whereas the size of the approximation and details at first level is the half, namely $148 \times 148$ pixels. For this reason the transformed image has exactly the same size of the original one.

Before going on with the overcomplete case, it is important to stress an important feature of the multi–resolution discrete Haar wavelet transform, which will be very useful when dealing with ranklets. As anticipated in the introductory part of Section 3.2, a more purely mathematical approach considers the wavelet transform as a weighted sum of compactly supported functions—also referred to as *wavelet basis functions*—of limited duration in time and frequency. The relationship between this more purely mathematical approach and the filter–based approach discussed here is—very broadly—that the weights of the sum can be thought of as the wavelet coefficients found by the analysis through the filter bank, whereas the wavelet basis functions as a continuous version of the digital filters. For a more rigorous approach, see (Mallat, 1989; Stollnitz *et al.*, 1996).

In such a context, the wavelet basis functions of the Haar transform are quite simple. In the one–dimensional case, they are two piecewise–constant functions:

$$\phi(x) = \begin{cases} +\frac{1}{\sqrt{2}} & 0 \leq x < 1 \\ 0 & \text{otherwise} \end{cases} \qquad \psi(x) = \begin{cases} +\frac{1}{\sqrt{2}} & 0 \leq x < 1/2 \\ -\frac{1}{\sqrt{2}} & 1/2 \leq x < 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

In the two–dimensional case, they are the four tensor products of one–dimensional wavelet basis functions, namely:

$$\phi\phi(x,y) = \phi(x)\phi(y) \quad \text{Approximation} \tag{3.9}$$

$$\phi\psi(x,y) = \phi(x)\psi(y) \quad \text{Horizontal detail} \tag{3.10}$$

$$\psi\phi(x,y) = \psi(x)\phi(y) \quad \text{Vertical detail} \tag{3.11}$$

$$\psi\psi(x,y) = \psi(x)\psi(y) \quad \text{Diagonal detail} \tag{3.12}$$

A pictorial representation of the two–dimensional Haar wavelet basis functions—also referred to as *Haar wavelet supports*—is given in Fig. 3.19.

### Multi–resolution overcomplete Haar wavelet transform

The *overcomplete wavelet transform* (OWT) removes the sub–sampling operation from the traditional discrete wavelet transform in order to produce a redundant representation. Over the years, several appellations has been given to it, including *un–decimated discrete wavelet transform*, *stationary wavelet transform* and *à trous algorithm*. In particular, the multi–resolution overcomplete Haar wavelet transform is an overcomplete wavelet transform which performs the wavelet analysis by using the Haar filters described in Eq. 3.6 and Eq. 3.7.

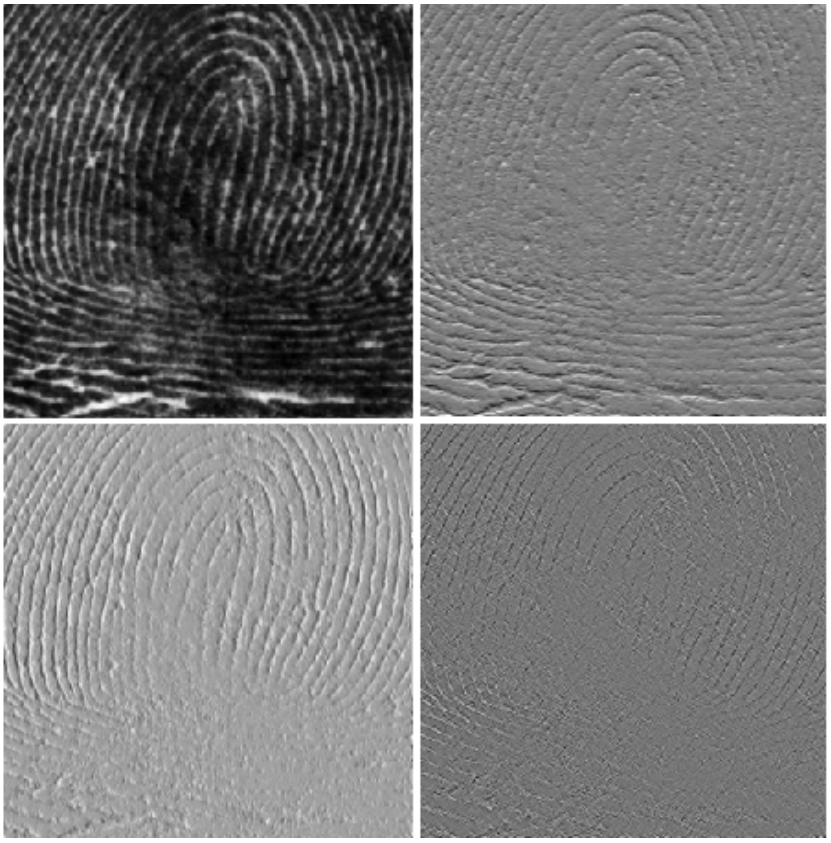


**Figure 3.11:** A two–level overcomplete Haar wavelet transform in two dimensions applied to the same fingerprint image analyzed in Fig. 3.10.

This transform has two fundamental properties. First, since no sub–sampling is applied, the number of samples in the transformed signal—or image—is much higher than that of the original one, namely it is redundant. Second, the transformed signal—or image—is aliasing–free.

As an example, Fig. 3.11 shows a one–level overcomplete Haar wavelet transform in two dimensions of the same fingerprint image analyzed in Fig. 3.10. As for the multi–resolution discrete Haar wavelet transform, details account differently for horizontal, vertical and diagonal ridges. On the other hand, differently from the multi–resolution discrete Haar wavelet transform, the size of the approximation and details at first level is equal to that of the original image, namely $296 \times 296$ pixels. This gives an idea of the high redundancy of this approach.

## 3.3 Steerable Filters

*Steerable filters* belong to a family of recursive multi–resolution transforms—such as wavelets—which in the last decades proved to be particularly useful in a wide variety of image processing applications. The first works describing the theory and implementations of this transform date back to the 1990s, as for example (Freeman & Adelson, 1991; Simoncelli *et al.*, 1992; Freeman, 1992; Simoncelli & Freeman, 1995; Karasaridis & Simoncelli, 1996). In the following years, a lot of publications came along dealing with several advanced applications of steerable filters to image processing, such as orientation analysis, noise removal, image enhancement, transient detection. However, out of the multitude of such works, those which probably deserve a specific mention are (Simoncelli & Farid, 1996; Simoncelli & Adelson, 1996; Simoncelli & Portilla, 1998; Portilla & Simoncelli, 2000; Portilla *et al.*, 2003; Jacob & Unser, 2004).

Going into detail of such a technique, the steerable approach consists of a linear transform in which an image is decomposed into a collection of different sub–bands localized both in resolution and orientation. The basis filters used for this transform are directional derivative operators that come in different resolutions and orientations. By changing the derivative order, the number of orientations may be adjusted, for example first derivatives yield two orientations, whereas second derivatives yield three orientations and so forth. In particular, the term *steerable* refers to the possibility of synthesizing those filters at arbitrary orientations.

The analogies between the steerable and the wavelet transform are several. First, the steerable transform is computed recursively using convolution and sub–sampling operations exactly as for the wavelet transform. Second, the multi–resolution scheme enabling to compute the two transforms at different resolutions is analogous. Third, as for the wavelet transform, the steerable transform is *self–inverting*, namely the matrix corresponding to the inverse transform is equal to the transpose of the forward transform matrix. However, also important differences there exist. The steerable representation of an image is *translation–invariant*, namely the sub–bands are aliasing–free or equivariant with respect to translation. It is also *rotation–invariant*, in other words the sub–bands are steerable or equivariant with respect to rotation. This proves to be very useful in applications requiring that the position or orientation of the image structure is encoded in the transformed image. At the same time, this represents also the primary drawback for steerable representation, since in that way the representation is *overcomplete* by a factor of $\frac{4k}{3}$, where $k$ is the number of orientation bands.

In the following, steerable filters will be carefully examined by walking mainly along the road traced by (Freeman & Adelson, 1991). In particular, a very explanatory example will be first given in order to get some familiarity with the steerable philosophy. Second, some theoretical results concerning the conditions for the steerability of continuous functions will be introduced, together with their extension to discretely sampled functions. Third, the recursive application of the steerable transform at different resolutions—namely the so–called *multi–resolution steerable pyramid*—will be discussed. Finally, a very interesting and useful family of asymmetric and steerable filters—known as *wedge filters*—will be described.

### 3.3.1 An introductory example

A circularly symmetric Gaussian function with scaling and normalization constants set to one can be written in Cartesian coordinates in the following way:

$$G(x, y) = e^{-(x^2+y^2)} \tag{3.13}$$

Let $G_n$ be the $n^{th}$ derivative of a Gaussian along the $x$ axis and $\theta$ the rotation operator, such that for any function $f(x, y)$, $f(x, y)^\theta$ represents the function $f(x, y)$ rotated through an angle $\theta$ about the origin. With this in mind, the first derivative of a Gaussian along the $x$ axis is:

$$G_1^0 = \frac{\partial}{\partial x} e^{-(x^2+y^2)} = -2xe^{-(x^2+y^2)} \tag{3.14}$$

and its $\pi$–rotated version is:

$$G_1^\pi = \frac{\partial}{\partial y} e^{-(x^2+y^2)} = -2ye^{-(x^2+y^2)} \tag{3.15}$$

Now, it can be easily demonstrated that a Gaussian $G_1^\theta$ at an arbitrary orientation $\theta$ can be expressed as a linear combination of $G_1^0$ and $G_1^\pi$, namely:

$$G_1^\theta = cos(\theta)G_1^0 + sin(\theta)G_1^\pi \tag{3.16}$$

In such a framework, $G_1^0$ and $G_1^\pi$ are called the *basis filters* for $G_1^\theta$, whereas $cos(\theta)$ and $sin(\theta)$ are known as the corresponding *interpolation functions* for those basis filters. In particular, the derivative of Gaussian filters discussed above offer a very simple idea of steerability, since they can be synthesized at arbitrary orientations by means of a linear combination of the basis filters, see Fig. 3.12.

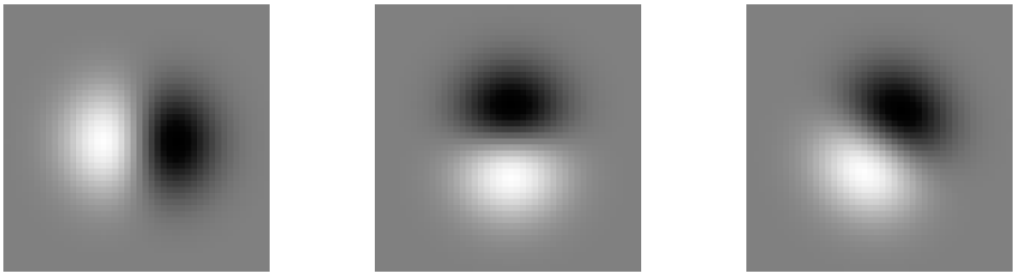


**Figure 3.12:** First derivatives of Gaussians at different orientations. The first derivative $G_1^0$ of a Gaussian along the $x$ axis is represented on the left. Its $\pi$–rotated version $G_1^\pi$ is represented on the middle. Formed by a linear combination of the above linear filters, $G_1^{2\pi/3}$ is represented on the right. Figure borrowed from (Freeman & Adelson, 1991).

### 3.3.2 Steerability

A lot of work has been done in order to find the theoretical conditions under which a function $f(x, y)$ steers, in other words it can be expressed as a linear combination of rotated version of itself:

$$f^\theta(x, y) = \sum_{j=1}^{M} k_j(\theta) f^{\theta_j}(x, y) \tag{3.17}$$

In both (Freeman & Adelson, 1991; Simoncelli *et al.*, 1992) a complete framework of theorems and proofs dealing with that aspect is given. In particular, two situations are considered, namely the case in which the function $f$ can be expanded in a Fourier series in polar angle $\phi$ and that in which it can be expressed as polynomials in Cartesian coordinates $x$ and $y$.

Suppose for instance that a function $f$ which can be expanded in a Fourier series in polar angle $\phi$ is given:

$$f(r, \phi) = \sum_{n=-N}^{+N} a_n(r) e^{in\phi} \tag{3.18}$$

Suppose also that it has a finite number $T$ of frequencies $-N \leq n \leq +N$ for which $f(r, \phi)$ has non–zero coefficients $a_n(r)$, namely it is a band–limited function in angular frequency. For example, the function $cos(\phi) = \frac{e^{+i\phi} + e^{-i\phi}}{2}$ has $T = 2$. Under

such conditions, it can be demonstrated that the function $f(r, \phi)$ can be steered in the following way:

$$f^{\theta}(r, \phi) = \sum_{j=1}^{M} k_j(\theta) g_j(r, \phi) \tag{3.19}$$

where $g_j(r, \phi)$ can be any set of functions. In particular, it can also be demonstrated that the minimum number of basis functions required to steer $f^{\theta}(r, \phi)$ is exactly $T$.

Similar conditions are valid also for functions $f$ which can be expressed as polynomials in Cartesian coordinates $x$ and $y$. Specifically, suppose that the function $f$ can be written as:

$$f(x, y) = W(r) P_N(x, y) \tag{3.20}$$

where $W(r)$ is an arbitrary windowing function and $P_N(x, y)$ is an $N^{th}$ order polynomial in $x$ and $y$. Then, it can be demonstrated that $f(x, y)$ can be synthesized at any orientation by linear combinations of $2N+1$ basis functions. Notice, however, that differently from the former situation, here the number of basis functions represents the number of sufficient basis functions needed to steer $f(x, y)$, not their minimum number.

Those results are quite interesting, since they state that steerability is a property common to a wide variety of functions, namely to all functions which can be written as Fourier series in polar angle $\phi$ or as a product of an arbitrary windowing function and a $N^{th}$ order polynomial in $x$ and $y$. In particular, the derivatives of a Gaussian function discussed in Section 3.3.1 are all steerable, since they are obtained by means of the product of a radially symmetric window function and a polynomial in $x$ and $y$.

In designing steerable filters useful for image processing purposes, however, it is necessary to fulfill some further requirements. First, when dealing with motion, texture and orientation analysis, it may be helpful to synthesize filters of a given frequency response with arbitrary phase. This allows to analyze spectral strength independently of phase. In order to achieve that, it is necessary to express functions $f$ as a linear combination of steerable *quadrature* pair of filters, namely pair of filters having the same frequency response but different in phase by $\pi$. The same concept can be expressed by saying that they are the Hilbert transform of each other. For example, the first and second rows of Fig. 3.13 show respectively the second derivative $G_2 = (4x^2 - 2)e^{-(x^2+y^2)}$ of a Gaussian and an approximation $H_2$ to its Hilbert transform. In particular, by means of the seven basis filters of $G_2$ and $H_2$ shown, $G_2$ can be shift arbitrarily in both phase and orientation.

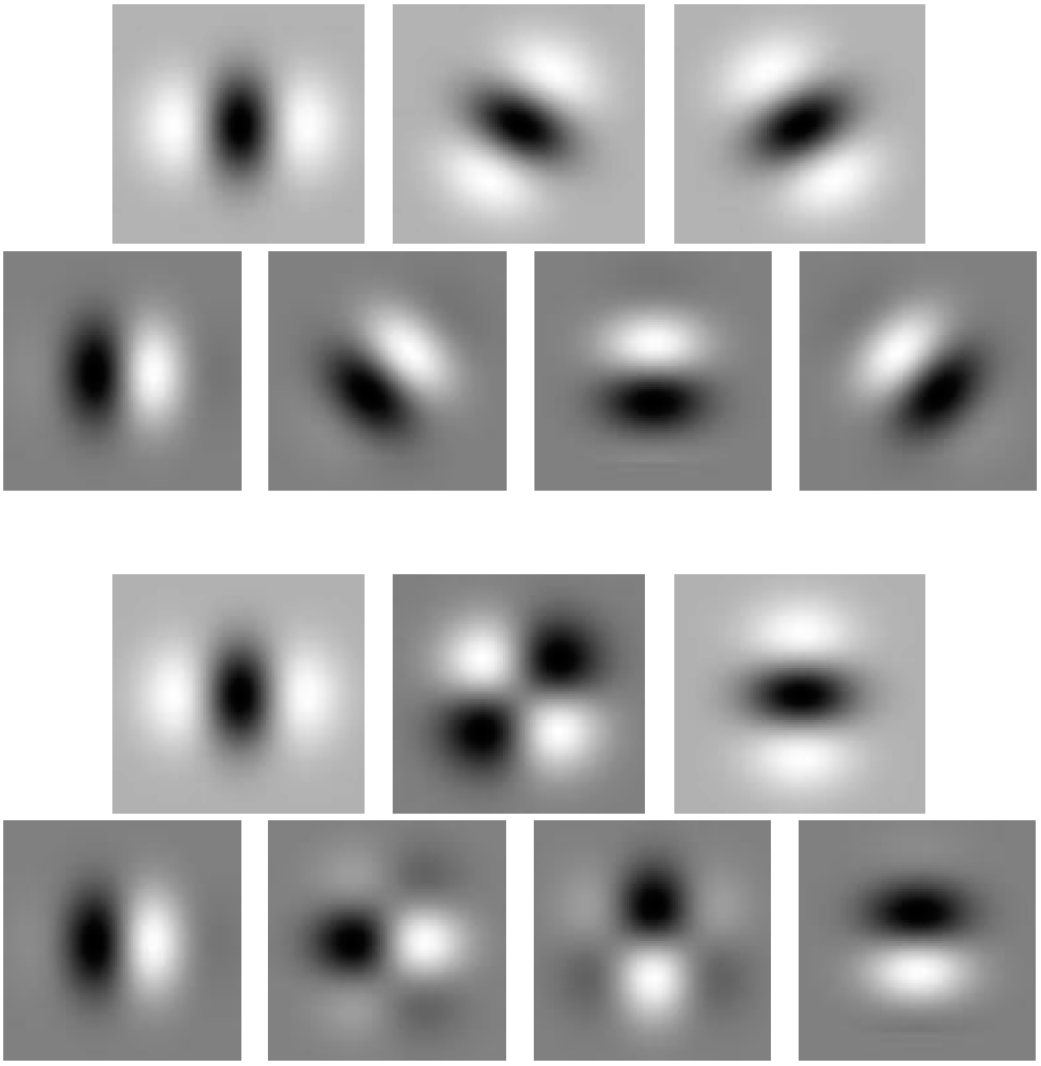


**Figure 3.13:** Quadrature filters. The second derivative $G_2 = (4x^2 - 2)e^{-(x^2+y^2)}$ of a Gaussian is represented on first row. The approximation $H_2$ to its Hilbert transform is represented on second row. The $x$–$y$ separable basis sets for $G_2$ and $H_2$ are respectively shown on third and fourth rows. Figure borrowed from (Freeman & Adelson, 1991).

The second important aspect to consider—when designing steerable filters—is their *x–y separability*. In fact, in all image processing applications a crucial point is to to have a high computational efficiency and—to this purpose—separability is fundamental. Fortunately, all functions $f$ which can be written as an $N^{th}$ order polynomial in $x$ and $y$ admit an $x$–$y$ separable basis, even though the number of the basis functions may prove to be large. In most cases, however, it is possible to find an $x$–$y$ separable basis which contains only the minimum number of basis filters. For more details on this aspect, see (Freeman & Adelson, 1991). As an example, the third and fourth rows of Fig. 3.13 show respectively the $x$–$y$ separable basis sets for $G_2$ and $H_2$.

Notice, finally, that the considerations drawn above for continuous functions can be extended to discretely sampled functions. In fact, if a continuous function is steerable, then its sampled version is steerable as well, since the order of sampling and steering is interchangeable. In this sense, a digital steerable filter can be obtained by sampling its continuous version.

### 3.3.3 Multi–resolution steerable pyramid

As already anticipated, one interesting application of steerable filters is in the analysis of images by means of a multi–resolution and self–inverting approach similar to that used by the wavelet transform. In particular—as for the wavelet decomposition—the steerable pyramid algorithm is based on recursive application of filtering and sub–sampling operations. The input image is partitioned into low–pass and high–pass sub–bands using filters $H_0$ and $L_0$. The low–pass sub–band is then further sub–sampled into low–pass and oriented band–pass sub–bands using filters $L_1$ and $B_k$, with $k$ variable according to the derivative filters used. For example, in Fig. 3.14 a single–stage of a first derivative steerable transform is shown. Notice that the number of band–pass filters $B_k$ is given by the derivative order of the filters plus one. Then, the pyramid structure is achieved by applying the single–stage transform recursively to the low–pass sub–band of the previous single–stage transform. In Fig. 3.14 this is achieved by inserting the portion of the diagram enclosed in the dashed box at the location of the filled circle. In order to have a clearer idea of the practical results obtained by such a steerable transform, in Fig. 3.15 a three–level steerable pyramid decomposition of a synthetic image representing a disk is depicted. Shown are the band–pass images obtained at six different orientations by using five order derivative filters. Shown is also the final low–pass image, whereas the initial high–pass image is not.

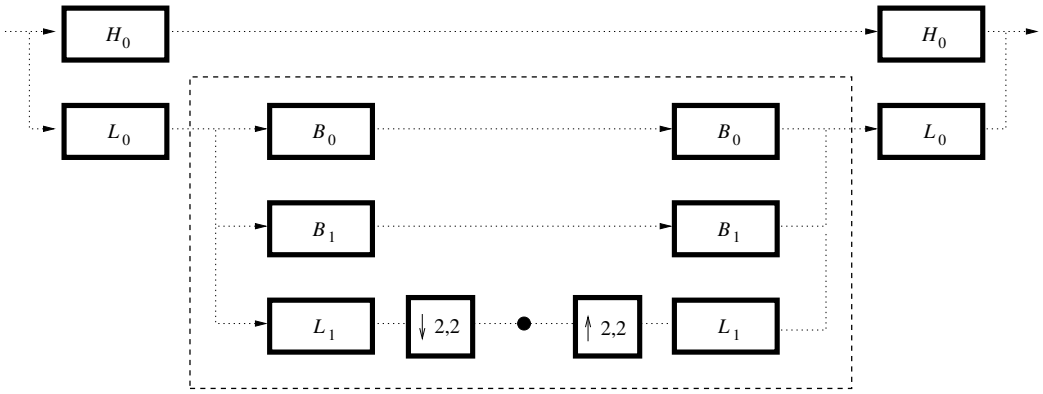



**Figure 3.14:** System diagram for a first derivative steerable pyramid.

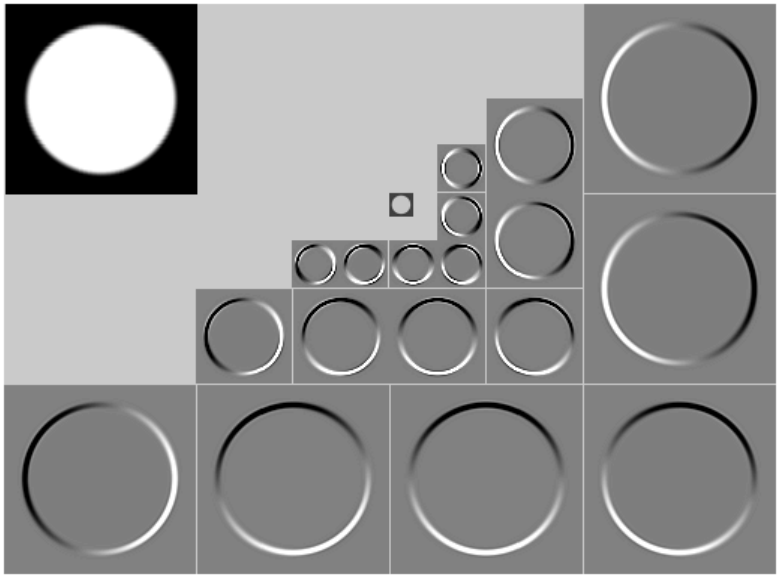


**Figure 3.15:** Steerable pyramid decomposition of the disk represented on top–left. Five order derivative steerable filters have been used. Shown are the six orientations at three different resolutions and the final low–pass image.

When dealing with steerable pyramids, it is well worth noticing that—in order to produce a usable transform—the filters $H$, $L$ and $B_k$ are highly constrained. The first family of constraints concerns the radial component of their Fourier transforms which must guarantee perfect reconstruction. To this aim, the $L_1$ filter should be constrained to have a zero response for frequencies higher than $\pi/2$ in both $\omega_x$ and $\omega_y$ axis of the Fourier domain. This ensures the elimination of aliasing. Then, the transfer function of the system should be equal to one, thus avoiding amplitude distortions. Finally, the low–pass branch of the diagram must be unaffected by insertion of the recursive portion of the system. On the other hand, the second family of constraints concerns specifically the band–pass filters $B_k$. In particular, the constraints are derived from imposing that their angular orientation tuning is constrained by the properties of steerability discussed in Section 3.3.2. For more details on both these two families of constraints, see (Simoncelli & Freeman, 1995; Karasaridis & Simoncelli, 1996). Notice also that unlike the wavelet transform, the steerable pyramid is significantly overcomplete. In particular, there are $\frac{4k}{3}$ times as many coefficients in the representation as in the original image. This redundancy limits its efficiency in terms of computational times, but sensibly increases its convenience for many image processing task for which orientation analysis is important.

### 3.3.4 Steerable wedge filters

As already discussed, steerable filters can be obtained at any orientation by means of linear combinations of directional derivatives of Gaussians, along with steerable approximations to their Hilbert transforms. Such basis, however, suffer from one drawback, namely they are always either symmetric or anti–symmetric with respect to the origin. In particular, an even–order derivative is always symmetric, whereas an odd–order is always anti–symmetric, see Fig. 3.12 and Fig. 3.13. This proves to be undesirable for many applications since in both symmetric and anti–symmetric case this determines that their *orientation energy*—namely the sum of their squared responses as a function of the orientation $\phi$—is always symmetric. In other words, their *orientation map*—namely the plot of the orientation energy as a function of $\phi$—is alway periodic with period $\pi$ regardless of the image considered. For example, while the orientation maps corresponding to the vertical line and cross in Fig. 3.16 are quite expected, those corresponding to the half–line and corner are not. In order to overcome this problem, researchers started exploring asymmetric filters for orientation analysis, see for example (Perona, 1992).

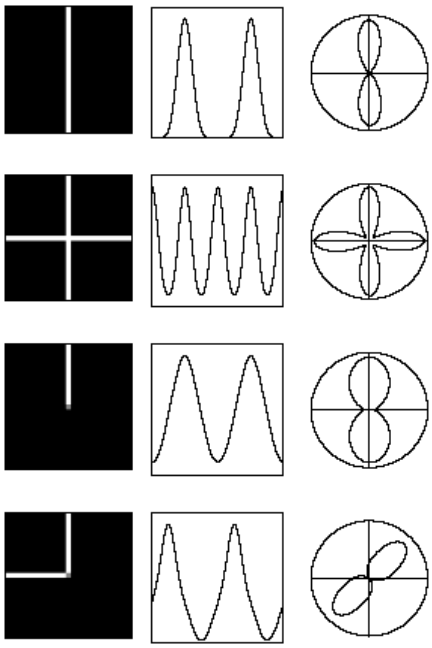


**Figure 3.16:** Orientation maps computed by means of the set $G_4/H_4$ of steerable filters described in (Freeman & Adelson, 1991). On the left, some synthetic images. On the middle, corresponding orientation energies computed using filters centered on the image ($\phi \in [0, 2\pi]$). On the right, orientation maps of the corresponding energies. Notice the symmetry also in the orientation maps of asymmetric images such as the half–line and the corner, namely third and fourth rows. Figure borrowed from (Simoncelli & Farid, 1995).

A class of both steerable and asymmetric filters suited for orientation analysis is represented by *wedge filters*, see (Simoncelli & Farid, 1995, 1996). In Fig. 3.17, for example, a set of ten steerable wedge basis filters is represented. As for the steerable filters discussed above, a steerable wedge filter can be obtained at any orientation from a linear combination of such basis filters. Other than the being steerable and asymmetric, they are also designed to produce an optimally localized oriented energy map. This proves to be crucial when dealing with images characterized by asymmetric structures, such as line endings or corners. For example, Fig. 3.18 shows orientation maps for a set of 18 steerable wedge filters. It is evident here that—differently from the results obtained by the $G_4/H_4$ filter set—steerable wedge filters respond as desired also to half–lines and corners.

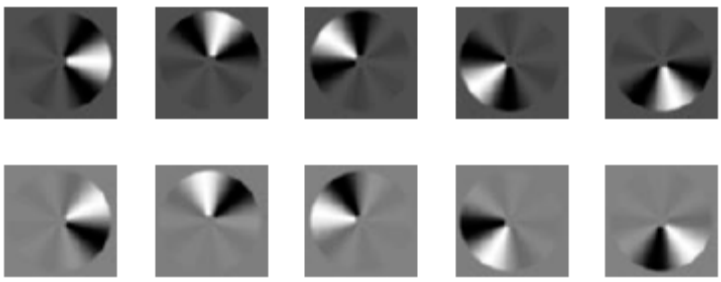


**Figure 3.17:** A set of ten steerable wedge basis filters. Figure borrowed from (Simoncelli & Farid, 1996).

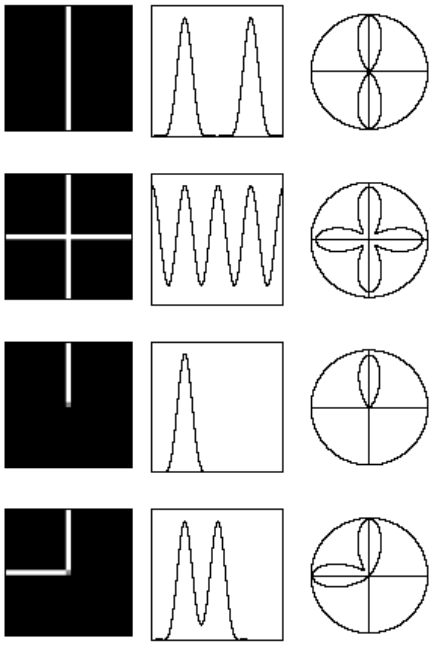


**Figure 3.18:** Orientation maps computed by means of the set of 18 steerable wedge filters described here. On the left, some synthetic images. On the middle, corresponding orientation energies computed using filters centered on the image ($\phi \in [0, 2\pi]$). On the right, orientation maps of the corresponding energies. Notice that the responses match the underlying images. Figure borrowed from (Simoncelli & Farid, 1995).

## 3.4 Ranklets

Ranklets have been introduced for the first time in (Smeraldi, 2002) as a family of non–parametric, orientation selective and multi–resolution features modeled on Haar wavelets. At the beginning, they have been mainly applied to pattern classification problems and in particular to face detection. For example, in both (Smeraldi, 2003a) and (Franceschi *et al.*, 2004) they have been used in order to encode the appearance of image frames representing potential face candidates. Later on, ranklets have been tested on the estimation of the 3D structure and motion of a deformable non–rigid object from a sequence of uncalibrated images, as described in (Del Bue *et al.*, 2004). Recently, an extension of the ranklet transform to hexagonal pixel lattices has been discussed in (Smeraldi & Rob, 2003), whereas a notion of completeness has been given in (Smeraldi, 2003b).

Since the beginning of 2004, ranklet–based techniques started being applied also to pattern classification problems concerning the detection of tumoral masses in digital mammograms. In particular, in (Masotti, 2004) ranklets have been used as image representations encoding crops of mammographic digital images. Tests demonstrated that ranklets perform definitely better than more traditional techniques as the ones discussed in (Angelini *et al.*, 2004), namely pixel–based and wavelet–based image representations. Later on—in order to discover which ranklet features influence most the classification performance—Recursive Feature Elimination (RFE) has been applied to the same problem, thus demonstrating that important ranklet features are just a few, see (Masotti, 2005).

In the following, the ranklet transform and its properties will be discussed. First, it will be shown that its non–parametric property derives from the fact that it is based on non–parametric statistics, namely statistics dealing with the relative order of pixels rather than with their intensity values. To this purpose, some introductory details on non–parametric statistics will be given. Second, it will be demonstrated that its orientation selective property derives from the fact that it is modeled on bi–dimensional Haar wavelets. This means that—in analogy to the wavelet transform—the vertical, horizontal and diagonal ranklet coefficients can be computed for each image. Third, it will be shown that the multi–resolution property derives from the fact that the ranklet transform can be calculated at different resolutions by means of a suitable stretch and shift of the Haar wavelet supports. Finally, a notion of completeness will be introduced for the ranklet transform, together with some considerations concerning the analogies existing between ranklets and what happens in mammalian retinal coding scheme.

### 3.4.1 Non–parametric statistics

#### Rank transform

Given a set of $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N$ pixels, the *rank transform* $\pi$ substitutes each pixel intensity value with its relative order—*rank*—among all the other pixels (Zabih & Woodfill, 1994). Here it follows an example:

$$\begin{pmatrix} 55 & 99 & 25 & 153 \\ 26 & 75 & 92 & 200 \\ 21 & 64 & 88 & 154 \\ 101 & 190 & 199 & 222 \end{pmatrix} \xrightarrow{\pi} \begin{pmatrix} 4 & 9 & 2 & 11 \\ 3 & 6 & 8 & 15 \\ 1 & 5 & 7 & 12 \\ 10 & 13 & 14 & 16 \end{pmatrix} \tag{3.21}$$

In case the set of $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N$ pixels contains pixels with equal intensity values, *mid–ranks* are introduced. They are computed assigning to each group of pixels with equal intensity values the average of the ranks they occupy, for example:

$$\begin{pmatrix} 55 & 99 & \mathbf{25} & 153 \\ \mathbf{25} & \mathbf{64} & 92 & 200 \\ 21 & \mathbf{64} & \mathbf{64} & 154 \\ 101 & 190 & 199 & 222 \end{pmatrix} \xrightarrow{\pi} \begin{pmatrix} 4 & 9 & \mathbf{2.5} & 11 \\ \mathbf{2.5} & \mathbf{6} & 8 & 15 \\ 1 & \mathbf{6} & \mathbf{6} & 12 \\ 10 & 13 & 14 & 16 \end{pmatrix} \tag{3.22}$$

#### Wilcoxon test

The rank transform and the *Wilcoxon test* are strictly related. Given a set of $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N$ pixels—in fact—suppose they are split into the two sub–sets $\mathsf{T}$ and $\mathsf{C}$, with $n$ and $m$ pixels each, so that $n + m = N$. In order to state whether the $n$ pixels in $\mathsf{T}$ have significantly higher intensity values than the $m$ pixels in $\mathsf{C}$, all the pixels are ranked, then the Wilcoxon test $W_S$ is introduced (Lehmann, 1995) and defined as the sum of the $n$ ranks $\pi(\boldsymbol{p}_i)$ in $\mathsf{T}$:

$$W_S = \sum_{i=1}^{n} \pi(\boldsymbol{p}_i) \tag{3.23}$$

The $n$ pixels in $\mathsf{T}$ are then judged to have significantly higher intensity values than the $m$ pixels in $\mathsf{C}$ if the Wilcoxon test is above a critical value $\tau$, in other words $W_S > \tau$. The value of $\tau$ determines the confidence level of the test.

### Mann–Whitney test

In order to deal with a test equivalent to the Wilcoxon test—but with an immediate interpretation in terms of pixels comparison—the *Mann–Whitney test* $W_{XY}$ is introduced (Lehmann, 1995):

$$W_{XY} = W_S - \frac{n(n+1)}{2} \tag{3.24}$$

As can be easily demonstrated, the value of the Mann–Whitney test $W_{XY}$ is equal to the number of pixel pairs $(\boldsymbol{p}_m, \boldsymbol{p}_n)$, with $\boldsymbol{p}_m \in \mathsf{T}$ and $\boldsymbol{p}_n \in \mathsf{C}$, such that the intensity value of $\boldsymbol{p}_m$ is higher than the intensity value of $\boldsymbol{p}_n$. Therefore, its values range from 0 to the number of pairs $(\boldsymbol{p}_m, \boldsymbol{p}_n) \in \mathsf{T} \times \mathsf{C}$, namely $mn$. Notice, however, that in order to compute the value of $W_{XY}$, these pairwise comparisons are never carried out explicitly. This, in fact, would results in approximately $O(N^2)$ operations, thus in huge computational times. On the contrary, its value is obtained by the application of the rank transform to the set of pixels $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N$, thus leading to only $NLogN$ operations.

## 3.4.2 Orientation selectivity

### Haar wavelet supports

As it will be clarified in the following, the non–parametric property of the ranklet transform derives from the fact that it is based on non–parametric transforms such as the rank transform and in particular on the Mann–Whitney test. Similarly, its orientation selective property derives from the fact that it is manly modeled on bi–dimensional Haar wavelets. Now, in order to arrive at the ranklet transform definition—thus putting some light onto the above statements—the first step consists of introducing the *Haar wavelet supports*.

Suppose that an image constituted by a set of $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N$ pixels is given. In order to compute the Mann–Whitney test, a possible choice in splitting the $N$ pixels is to split them into two sub–sets $\mathsf{T}$ and $\mathsf{C}$ of size $n = m = N/2$, thus assigning half of the pixels to the sub–set $\mathsf{T}$ and half to the sub–set $\mathsf{C}$. With this in mind, it is possible to define the two sub–sets $\mathsf{T}$ and $\mathsf{C}$ being inspired by the three Haar wavelet supports, as shown in Fig. 3.19. In particular, for the vertical Haar wavelet support—also referred to as $h_{\mathsf{V}}$—the two sub–sets $\mathsf{T}_{\mathsf{V}}$ and $\mathsf{C}_{\mathsf{V}}$ are defined. Similarly, for the horizontal Haar wavelet support $h_{\mathsf{H}}$ the two sub–sets $\mathsf{T}_{\mathsf{H}}$ and $\mathsf{C}_{\mathsf{H}}$ are defined, whereas for the diagonal Haar wavelet support $h_{\mathsf{D}}$ the two sub–sets $\mathsf{T}_{\mathsf{D}}$ and $\mathsf{C}_{\mathsf{D}}$ are defined.

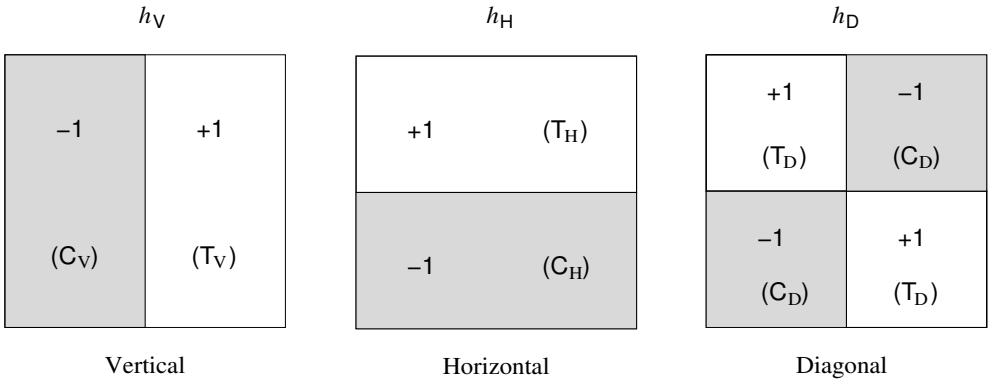



**Figure 3.19:** The three Haar wavelet supports $h_{\mathsf{V}}$, $h_{\mathsf{H}}$ and $h_{\mathsf{D}}$. From left to right, the vertical, horizontal and diagonal Haar wavelet supports.

Notice that, the arbitrariness characterizing the selection of the two sub–sets $\mathsf{T}$ and $\mathsf{C}$ is fundamental in order to be able to freely choose the two sub–sets based on the Haar wavelet supports. In other words, the arbitrariness with which the two sub–sets are chosen forms the basis for the orientation selective property of the ranklet transform.

### Ranklet coefficients

Once the rank transform, the Mann–Whitney test and the Haar wavelet supports have been introduced, the definition of the *ranklet transform* is straightforward. In fact, given an image constituted by a set of $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N$ pixels, the horizontal, vertical and diagonal ranklet coefficients can be computed in the following way:

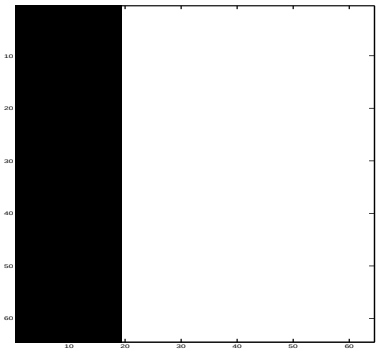
$$R_j = \frac{W_{XY}^j}{mn/2} - 1, \quad j = \mathsf{V}, \mathsf{H}, \mathsf{D} \tag{3.25}$$

Here $W_{XY}^j$ is computed by splitting the $N$ pixels into the two sub–sets $\mathsf{T}_j$ and $\mathsf{C}_j$ differently for each $j = \mathsf{V}, \mathsf{H}, \mathsf{D}$, as previously discussed for the Haar wavelet supports.

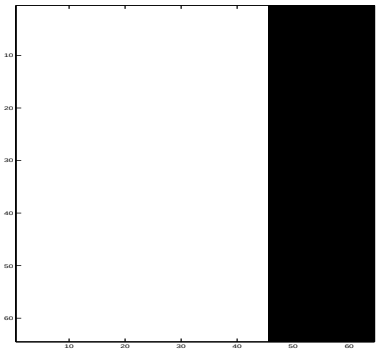
The geometric interpretation of the ranklet coefficients $R_j$—with $j = \mathsf{V}, \mathsf{H}, \mathsf{D}$—is quite simple, see Fig. 3.20. Suppose that the image we are dealing with is characterized by a vertical edge with the darker side on the left, where $\mathsf{C}_{\mathsf{V}}$ is located, and the brighter side on the right, where $\mathsf{T}_{\mathsf{V}}$ is located. Then $R_{\mathsf{V}}$ will be close to +1, as many pixels in $\mathsf{T}_{\mathsf{V}}$ will have higher intensity values than the

pixels in $\mathsf{C}_\mathsf{V}$. Conversely, $R_\mathsf{V}$ will be close to $-1$ if the dark and bright side are reversed. At the same time, horizontal edges or other patterns with no global left–right variation of intensity will give a value close to 0. Analogous considerations could be drawn for the other ranklet coefficients, namely $R_\mathsf{H}$ and $R_\mathsf{D}$.

Notice that the definition given for the ranklet coefficients in Eq. 3.25 clarifies the reasons for both the non–parametric and orientation selective properties of the ranklet transform. In particular, the computation of the ranklet coefficients by means of the Mann–Whitney test $W_{XY}$ determines the non–parametric properties of the ranklet transform. On the other hand, the possibility to calculate them at different orientations—namely vertical, horizontal and diagonal—by means of the Haar wavelet supports, determines its orientation selective property.



$$\Rightarrow \qquad R_{\mathsf{V,\,H,\,D}} = [+0.59, 0, 0]$$



$$\Rightarrow \qquad R_{\mathsf{V,\,H,\,D}} = [-0.59, 0, 0]$$

**Figure 3.20:** Ranklet transform applied to some synthetic images.

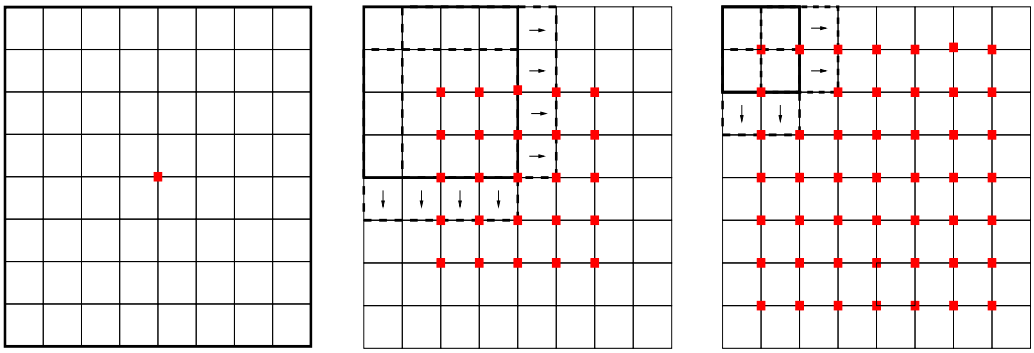


**Figure 3.21:** Multi–resolution ranklet transform at resolutions 16, 4 and 2 pixels, of an image with pixel size $16 \times 16$.

### 3.4.3 The multi–resolution approach

From the considerations drawn above, it is evident what are the causes determining the non–parametric and orientation selective properties of the ranklet transform. Analogously, it is possible to justify also its multi–resolution property. In fact, the close correspondence between the Haar wavelet transform and the ranklet transform leads directly to the extension of the latter to its multi–resolution formulation. Similarly to what is usually done for the Haar wavelet transform—therefore—the ranklet coefficients at different resolutions can be computed simply stretching and shifting the Haar wavelet supports. This means that the multi–resolution ranklet transform of an image is a set of triplets of vertical, horizontal and diagonal ranklet coefficients, each one corresponding to a specific resolution and shift of the Haar wavelet supports.

For example, suppose that the multi–resolution ranklet transform of an image with pixel size $16 \times 16$ is performed at resolutions 16, 4 and 2 pixels, namely using Haar wavelet supports with pixel size $16 \times 16$, $4 \times 4$ and $2 \times 2$, see Fig. 3.21. This actually means that the ranklet transform of the image is computed at resolution 16 pixels, by shifting the Haar wavelet support with linear dimensions 16 pixels, at resolution 4 pixels, by shifting that with linear dimensions 4 pixels and at resolution 2 pixels, by shifting that with linear dimensions 2 pixels. Suppose also that the horizontal and vertical shifts of the Haar wavelet supports along the horizontal and vertical dimensions of the image are of 1 pixel. Then the multi–resolution ranklet transform of the image is composed by 1 triplet $R_{\mathrm{V,H,D}}$ of ranklet coefficients deriving from the ranklet transform at resolution 16 pixels, 25 triplets $R_{\mathrm{V,H,D}}$ from that at resolution 4 pixels and 49 triplets $R_{\mathrm{V,H,D}}$ from that at 2 pixels.

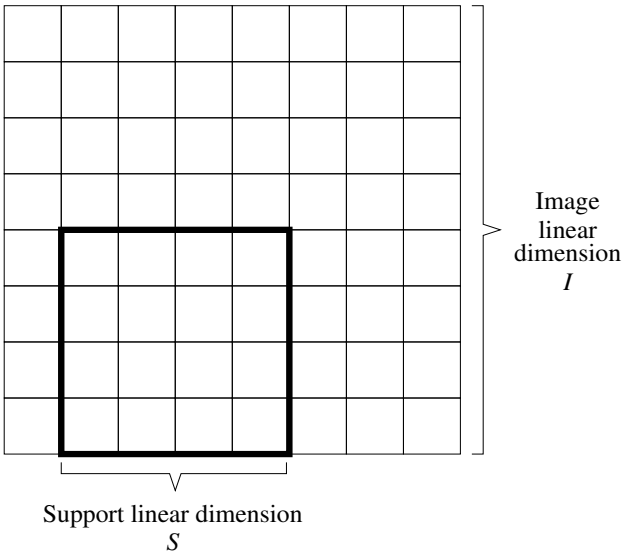



**Figure 3.22:** Linear dimensions $I$ and $S$ respectively of the image and of the Haar wavelet support.

Notice that—in order to generalize the above discussed calculation to whatever image size and resolution—the number $nT$ of triplets $R_{\mathsf{V,H,D}}$ at each resolution is computed as:

$$nT = (I + 1 - S)^2 \tag{3.26}$$

Here $I$ and $S$ represent respectively the linear dimension of the image and that of the Haar wavelet support, as shown in Fig. 3.22.

### 3.4.4 Completeness

Other than the above discussed orientation selectivity and multi–resolution, the ranklet transform shares with the wavelet transform also a notion of *completeness*. It is evident that, when dealing with ranklets, completeness does not refer to the ability of exact image reconstruction as for wavelets. This could not be an issue for the ranklet transform, since ranklets are defined in terms of the relative order of the pixel values and—in particular—their values are disregarded during the ranklet transform. In such a context—thus—completeness is rather intended in the sense that a set of ranklets is complete if their value is sufficient to unambiguously specify the order of the pixel values.

This concept can be formalized by following (Smeraldi, 2003b). Suppose that an image is constituted by a set of $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N$ pixels. As already discussed, the pixel values can be substituted by their relative order among all the others. This means that an image $(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N) \in \mathbb{R}^N$ can be identified with an element $(1, 2, \ldots, N) \in S_N$, namely the symmetric group of the permutations of $N$ points. Let now $C = \{\mathsf{W}_1, \mathsf{W}_2, \ldots, \mathsf{W}_m\}$ be a covering of $(1, 2, \ldots, N)$. The ranklet decomposition $\mathcal{R}$ is thus defined as the map:

$$\mathcal{R} : S_N \to \mathbb{R}^m \tag{3.27}$$

that sends a permutation $\pi \in S_N$ into the vector $\mathcal{R}(\pi) = (\mathcal{R}^i_{\mathsf{W}_j})_{\forall i, \forall j}$ obtained by computing each admissible ranklet $\mathcal{R}^i$ depending on the dimensionality of the image and on every window $\mathsf{W}_j \in C$. Now, the following definition can be stated:

**Definition 1** *The ranklet decomposition $\mathcal{R}$ is complete if it is invertible over its range, i. e. if it is one–to–one.*

In other words, the ranklet decomposition $\mathcal{R}$ is said complete if there exists a map:

$$\tilde{\mathcal{R}} : \mathcal{R}(S_N) \to S_N \tag{3.28}$$

such that for every image $X$:

$$\tilde{\mathcal{R}}\,(\mathcal{R}(\pi_X)) = \pi_X \tag{3.29}$$

where $\pi_X$ is the ranking of $X$. In this sense, the map $\tilde{\mathcal{R}}$ is the closest analogue for ranklets of the reconstruction operator previously discussed for the wavelet transform. In particular, it can be demonstrated that for the ranklet transform such a completeness holds. More details on the demonstrations proving the completeness for ranklets are given in (Smeraldi, 2003b).

Notice that proving the completeness for ranklets has not an operative importance. As already discussed, in fact, exact image reconstruction is not an issue for the ranklet transform, differently from the wavelet transform. However, the completeness of ranklets has a theoretical significance itself. First, it shows that the entire information available to rank features is effectively captured by ranklets. Second, it further clarifies the analogy between ranklet transform and Haar wavelet transform. Finally, it is of great relevance for some computational models of biological vision suggesting that temporal order retinal coding might form a rank–based image representation in the visual cortex.

### 3.4.5 Retinal coding toward the visual cortex

A reasonable explanation for the good results achieved by ranklets in visual pattern classification problems—such as the above discussed face detection or tumoral mass detection in digital mammograms—can be traced in their being very close to the real way in which retinal ganglion cells encode the visual informations sent to the brain. Several works, in fact, suggest that retinal ganglion cells encode the information sent to the visual cortex in the brain by mainly using a rank–based coding scheme. In the following, some very general concepts on this topic will be discussed.

It is generally assumed that the information transmitted from the retina to the brain codes the intensity of the visual stimulus at every location in the visual field. Although this strong statement can certainly be a simplification, it is clear that the aim of retinal coding is to transmit enough information about the image on the retina to allow objects and events to be identified. Studies on the speed of visual processing demonstrate that the time available for information transmission through the visual system is severely limited. In particular, it appears that information processing and transfer should be less than 50 ms between the retina and the brain. Classically, ganglion cells are thought to encode their inputs in their output firing frequency. The process of retinal spike train generation is supposed to be stochastic, namely subject to a Poisson or pseudo–Poisson noise. In particular, two implementations of rate coding are usually considered. One supposing that significant informations are encoded on the number of spikes of ganglion cells, the other on the mean inter–spike interval. Nevertheless, the firing frequency is not the only option. In recent years, in fact, a strong debate has opposed works suggesting that codes are embedded in the neurons mean firing rates and works in favor of temporal codes, namely codes embedded in the precise temporal structure of the spike train. The literature in this field is divided as well. Some publications adopt the firing rate approach, such as (Warland *et al.*, 1997). Other approaches adopt the temporal approach, such as (Softky, 1995; Shadlen & Newsome, 1995, 1998; Gautrais & Thorpe, 1998).

However, another hypothesis concerning the temporal coding scheme suggests that the retinal encoding could be based on the order—namely on the rank—of firing over a population of ganglion cells. The idea is that the most strongly activated ganglion cells tend to fire first, whereas more weakly activated cells fire later or not at all. In such a context, the relative timing in which the ganglion cells fire the first spike of their spike train can be used as a rank code, see (Thorpe, 1990).

In (Van Rullen & Thorpe, 2001), for example, this rank order encoding scheme has been deeply tested and compared to a pair of encoding schemes respectively based on the spike count and on the mean inter–spike interval. In particular, it has been shown that this rank order coding outperforms the other two and that can lead to a very good stimulus reconstruction. It has appeared also that the very first spikes generated in the retina can carry sufficient information for further cortical processing, thus assuring relatively short time periods for information transmission through the visual system.

Chapter 4

# Exploring Image Representations Performance

*The following Chapter will be devoted to the discussion of the experiments performed in order to find the crop's image representation which provides the best classification performance. As already anticipated, in fact, the novel technique pursued in this work consists of classifying—by means of a Support Vector Machine (SVM)—the entire pixels of the crop, or at least a transformed version of them, where transforms tested are the wavelet transform, the steerable pyramid and the ranklet transform. To this aim, the Chapter will start with Section 4.1 by giving an overview of the research approach adopted and will continue by describing the data set and methods used. It will then go further by discussing the simplest case, namely the choice of a pure pixel–based image representation. In particular, Section 4.2 will review and discuss the results obtained for this case. Section 4.3 will describe the experiments performed with a wavelet–based image representation, namely the case in which crops are transformed by using the multi–resolution discrete Haar wavelet transform and its overcomplete version. Section 4.4 will discuss the results obtained by introducing a higher orientation selectivity in the image representation, namely transforming the crops by means of steerable pyramids. Finally, Section 4.5 will describe in detail a novel and very promising approach based on ranklets. In particular, the performances of this ranklet–based image representation will be explored by means of SVM Recursive Feature Elimination (SVM–RFE), namely recursively eliminating some of the less discriminant ranklets coefficients according to the cost function of SVM.*

## 4.1 The Research Approach Adopted

### 4.1.1 Overview

The primary objective of this work is to solve a two–class classification problem in which the two classes are represented respectively by crops of tumoral masses and crops of normal tissue, both collected from diagnosed digital mammograms. To this aim, SVM is chosen as classifier—see Section 2.2—whereas different crop's image representations are singularly evaluated as classification features. In particular, in order to find the optimal solution for this two–class classification problem, the different crop's image representations are evaluated while SVM's parameters are tuned as to achieve the best possible classification performance. Notice that this approach is *novel* as regards mammographic mass classification and can be viewed as a sort of *featureless* approach, since no a priori information is extracted from the crops themselves. The image representation is in fact submitted to the classifier as it is. In particular—differently from what it is classically done when dealing with such problems—no geometrical informations are extracted from the crops, such as for example circularity, gradient and so forth.

As already anticipated, the image representations explored in this work are traditional and variated versions of the pixel–based, wavelet–based, steer–based and ranklet–based image representations introduced in Chapter 3. In the evaluation of the *pixel–based image representation*, for example, the raw pixel values of each crop—see Section 3.1.1—are used to train and test SVM. Some variations on the pixel–based theme are also evaluated, namely resized and equalized crops, see respectively Sections 3.1.2 and 3.1.3.

In exploring the *wavelet–based image representation*—see Section 3.2.3—the wavelet coefficients obtained from the application of the multi–resolution Haar discrete wavelet transform to the crop are presented to the classifier. An analogous study is conducted also for the multi–resolution Haar overcomplete wavelet transform.

As regards the *steer–based image representation*, the steerable pyramid discussed in Section 3.3.3 is applied to the crop and the resulting coefficients are then considered as the classification features. A variation on the steer–based theme is also considered, namely the application of wedge filters—see Section 3.3.4—in order to find the angle at which the orientation energy is maximal. In this case, the classifier is trained and tested with the coefficients obtained by applying to the crop the steerable pyramid oriented at the maximal orientation energy angle.

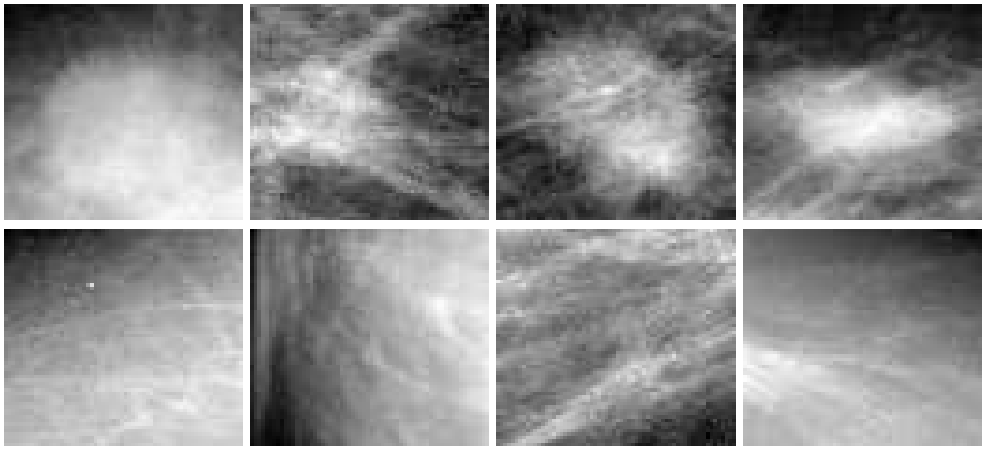


**Figure 4.1:** The two classes. Mass class (top row). Non–mass class (bottom row).

Finally, the *ranklet–based image representation*—see Section 3.4—is evaluated. Here, the triplets of ranklet coefficients obtained by applying the multi–resolution ranklet transform to the crop are presented to the classifier. Its classification performance is then further explored as the ranklet coefficients are eliminated by means of the recursive feature elimination technique discussed in Section 2.3, namely SVM–RFE.

Notice that the approach discussed above—and the results that will be shown in the following—walk mainly along the road traced by some recent work, namely (Angelini *et al.*, 2004) for the pixel–based and wavelet–based image representations, (Masotti, 2004) for the ranklet–based image representation and, finally, (Masotti, 2005) for the application of SVM–RFE to ranklet coefficients.

### 4.1.2 Data set

The data set used to evaluate the different image representations is comprised of 6000 crops with pixel size $64 \times 64$ representing the two classes, namely tumoral masses and normal tissue—or non–masses—as shown in Fig. 4.1. In particular, the number of crops representing the mass class is 1000, whereas that of crops representing the non–mass class is 5000. All the crops are extracted—and then resized to pixel size $64 \times 64$—from the diagnosed mammographic images belonging to the Digital Database for Screening Mammography (DDSM) collected by the University of South Florida (USF), see (Heath *et al.*, 2000).

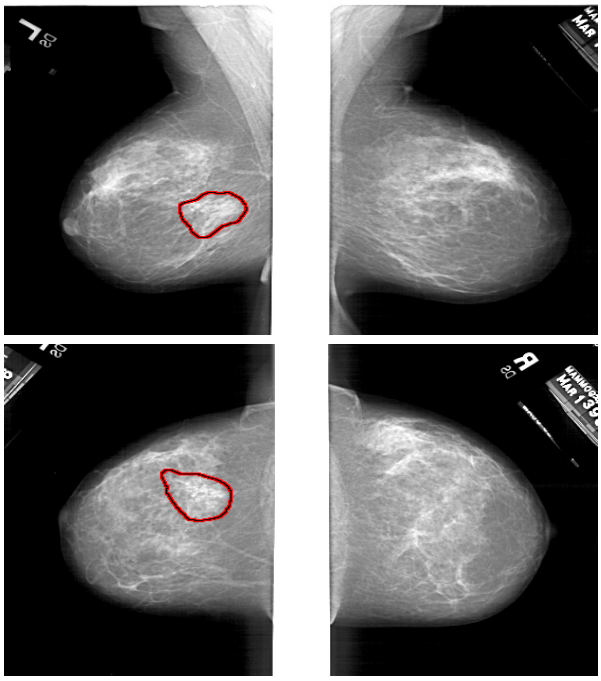


**Figure 4.2:** Four standard views of a mammographic case from the DDSM database. Left and right medio–lateral oblique views (top row). Left and right cranio–caudal views (bottom row). The suspicious region is marked in both the two views of left breast. In this specific case, the abnormality is a malignant spiculated mass with architectural distortions.

In particular, the DDSM is a database of digitized film–screen mammograms with associated ground truth and other information that was completed in the fall of 1999. The purpose of this resource is to provide an on–line large set of mammograms in a digital format usable by researchers in order to evaluate and compare the performance of CAD algorithms.

It contains 2620, four–view mammography screening exams—also referred to as *cases*—obtained from Massachusetts General Hospital, Wake Forest University School of Medicine, Sacred Heart Hospital and Washington University of St. Louis School of Medicine. The four standard views—namely *medio-lateral oblique* and *cranio–caudal*, one for each of the two breasts—are digitized with Lumisys scanner at 50 $\mu$m or Howtek scanner at 43.5 $\mu$m pixel size, both with a 12–bit gray–level resolution, as shown for example in Fig. 4.2.

Each case is diagnosed differently by the radiologist according to the severity of the finding. *Normal cases* contain mammograms which are read as normal from screening exams and have a normal screening exam four years later. *Benign cases* contain cases in which something suspicious is found and the patient is recalled for some additional work–up that resulted in a benign finding. Finally, *cancer cases* are those in which a histologically proven cancer is found.

### 4.1.3 Methods

One of the most common problems one has to face—when dealing with a two–class classification problem—is the lack of samples in order to train and test the classifier. As already discussed in Section 2.1.3, cross–validation is a common procedure used to handle classifiers when the dimensionality of the data set is limited. In particular, given a $n$–dimensional data set $\mathcal{D}$, first it is divided into $k$ homogeneous sub–sets $\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_k$, also known as *folds*. Then the classifier is trained with the collection of the first $k-1$ folds— $\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_{k-1}$—and tested on the fold left over, namely $\mathcal{F}_k$. The procedure is thus permuted for each $\mathcal{F}_i$, where $i = 1, \ldots, k-1$.

As it is evident from the dimensions cited for the training and test sets in Section 4.1.2, the data set used in this work does not represent an exception as regards the lack of samples. In order to overcome the difficulties arising from that restricted number of available crops, a 10–folds cross–validation procedure is thus implemented. As discussed in Section 2.1.3, in fact, this number of folds is the most reasonable choice for sparse data set. According to this choice, the data set is therefore divided into 10 folds, each one containing 100 mass crops and 500 non–mass crops. In this way, for each permutation of the cross–validation procedure, SVM is trained with 900 mass crops and 4500 non–mass crops, whereas it is tested on 100 mass crops and 500 non–mass crops.

As regards the classification performances of the different image representations, they are compared using ROC curves. Section 2.1.4—in fact—introduced ROC curve analysis as a widely employed method in order to evaluate the performance of a classifier used to separate two classes. Actually, they are particularly suited for that kind of problems being plots of the classifier's *True Positive Fraction* ($TPF$) versus its *False Positive Fraction* ($FPF$), where the quantity $TPF$ is also known as the system *sensitivity* and the quantity $1 - FPF$ as the system *specificity*. For more details, see Eq. 2.5, 2.6, 2.7 and 2.8.

In this work, the quantity $FPF$ is represented by the fraction of non–masses that have been incorrectly classified as belonging to the mass class, whereas the quantity $TPF$ by the fraction of masses that have been correctly classified as belonging to the mass class. Furthermore, it has been already pointed out in Section 2.1.4 that in order to generate a full ROC curve instead of just a single point, it will suffice varying the free parameters of the learning machine, thus altering the values of $TPF$ and $FPF$ on the same test set. In this way, it is possible to trade a lower—or higher—$FPF$ value for a higher—or lower—$TPF$ value by choosing appropriate values for the free parameters under study. In the present work, this is achieved by recursively changing the threshold $b$ which represents the position in the feature space of the Maximal Margin Hyperplane found by SVM, see Eq. 2.45. This actually corresponds to moving the hyperplane of the SVM solution in the feature space. In particular, the fraction of true positives and false negatives is then computed for each choice of the threshold $b$. Each single point of the ROC curves is therefore obtained by averaging the results of the 10–folds cross–validation technique applied to the entire data set.

In such a context, the purpose of this work can be—*very informally*—seen as finding image representations characterized by ROC curves which climb rapidly toward the upper–left corner of the graph. This means, in fact, a high number of masses that have been correctly classified as belonging to the mass class and a low number of non–masses that have been incorrectly classified as belonging to the mass class. In particular, this family of ROC curves are preferable to those which follows a diagonal path from the lower–left corner to the upper–right corner. The latter situation, in fact, represents the case in which every improvement in $FPF$ is matched by a corresponding decline in the $TPF$.

A final remark deserves to be pointed out. In this work, classification features are generally submitted to SVM after being processed by a technique—known as *scaling*—which re–maps correspondent features of the training and test sets in the range $[0, 1]$. Here, correspondent features are intended to be correspondent pixels when evaluating the pixel–based image representation, correspondent wavelet coefficients when evaluating the wavelet–based image representation and so forth. The scaling coefficients are calculated for each feature during the training phase, then are used to scale correspondent features both in the training and test set. This technique is very common in the pattern classification community, since it is useful in order to avoid that features of greater value dominate those of smaller value. Furthermore, since classification depends mainly on the inner products of feature vectors, the scaling technique is useful to avoid numerical difficulties.

## 4.2 Pixels Performance

The simplest way to code an image is by just concatenating all its intensity values, thus yielding a long vector with as many entries as the number of pixels in the image. This codification is usually referred to as the *pixel–based image representation* of the image under analysis.

In the specific context of image classification, adopting the pixel–based image representation actually forces the classifier to separate the images under exam into different classes by simply using the informations derived from the intensity values of their pixels. In some sense, it forces the classifier to learn the typical intensity content of images representing tumoral mass and that of images representing normal tissue.

In the following, few words will be used in order to describe the three main pixel–based image representation evaluated in this work. In particular, the classical pixel–based image representation will be briefly discussed, together with a few theme variations based on image resizing and histogram equalization. Notice that all these three featureless techniques based on the pixel–based image representation constitute a *novel* approach to the mammographic mass classification.

### 4.2.1 Original pixel–based image representation

Dealing with the specific case of mammographic tumoral masses and normal tissue classification, the two classes to separate look like as in Fig. 4.3, when characterized by their pixel–based image representation. In other words, their classification features are represented exactly by the $64 \times 64$ intensity values of their pixels.

Notice that—when choosing the pixel–based image representation—masses and normal tissue appear as they are in reality, namely masses appear as round–shape objects with defined edges, whereas non–masses appear as less defined and very heterogeneous objects. An exception to round–shape masses is represented by *spiculated* masses, namely objects having a star–shaped boundary or margin with sharp fingers pointing away from the center of the mass. Although these two families of masses are quite different, common characteristics which differentiate them from non–masses are the tendency to have a fairly sharp boundary and to appear brighter than the surrounding tissue. In particular, giving information on the shape of the boundary and contrast with the surrounding tissue, the pixel–based image representation emphasizes specifically these shared characteristics.

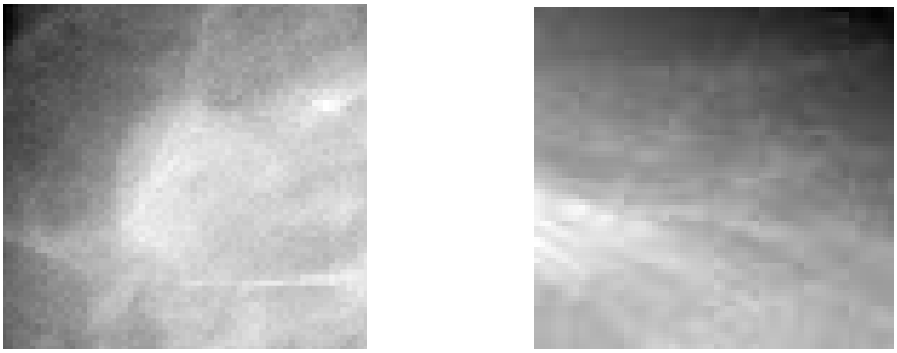


**Figure 4.3:** Original pixel–based image representation. Mass (left). Non–mass (right). Characteristics which differentiate masses from normal tissue are the tendency of the former to have a fairly sharp boundary and to appear brighter than the surrounding tissue.

In the rest of this work, the above discussed image representation—also referred to as *original pixel–based image representation*—will be indicated as **PixS**. In particular, the pre–fix **Pix** stands for being a pixel–based image representation, whereas the post–fix **S** stands for having correspondent classification features—namely pixels—scaled in the interval [0, 1], as discussed in Section 4.1.3.

### 4.2.2 Equalized pixel–based image representation

With the purpose of giving extra importance to the former characteristic differentiating masses from normal tissue—namely the sharpness of mass boundary—histogram equalization is applied to the mammographic crops. As already discussed in Section 3.1.3, in fact, the net effect of histogram equalization on images is to transform them into images having higher contrast and exhibiting a larger variety of gray tones. In the present case, this actually results in having crops in which edges and boundaries are enhanced, as shown in Fig. 4.4. In some sense, thus, the idea is to make more noticeable to SVM the differences existing between masses and normal tissue when looking specifically at their boundary. It is evident that the underlying hypothesis is that—since masses have generally sharp boundaries, whereas normal tissue has blunt ones, or at least none—this can help SVM in separating the two classes.

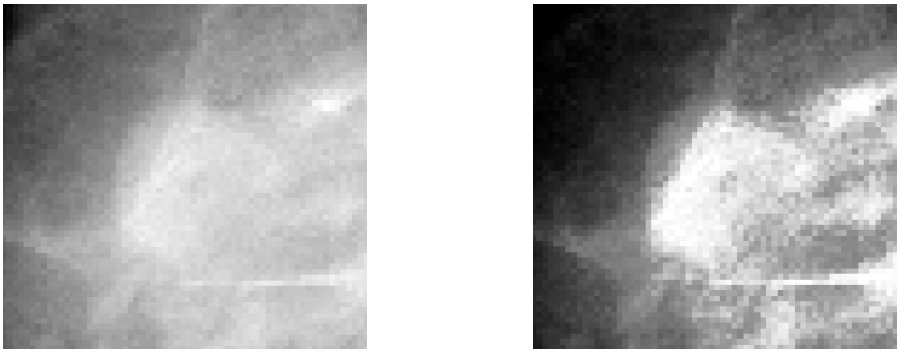


**Figure 4.4:** Equalized pixel–based image representation. Original mass (left). Equalized mass (right). The net effect of histogram equalization on crops is to transform them into images having higher contrast and exhibiting a larger variety of gray tones. This results in an enhancement of edges and boundaries.

In the rest of this work, the above discussed image representation—also referred to as *equalized pixel–based image representation*—will be indicated as **PixH**. As for the original pixel–based image representation, the pre–fix **Pix** stands for being a pixel–based image representation, whereas the post–fix **H** stands for being submitted to histogram equalization.

### 4.2.3 Resized pixel–based image representation

Going exactly into the opposite direction of the equalized pixel–based image representation—namely giving extra importance to the brightness of masses with respect to the surrounding tissue—bi–linear image resizing is applied to the crops. The resulting crops are characterized by a lower spatial resolution which supplies a very approximative idea about edges and boundaries, but which provides an effective picture of the brightness distribution of the pixels, see Fig. 4.5. Contrarily to what considered for the equalized case, the idea here is thus to make more noticeable to SVM the differences existing between masses and normal tissue when looking specifically at the brightness distribution of the pixels. The underlying hypothesis here is that—since masses are generally characterized by a central nucleus brighter that the surrounding tissue, whereas normal tissue has typically none—this can help SVM in separating the two classes.

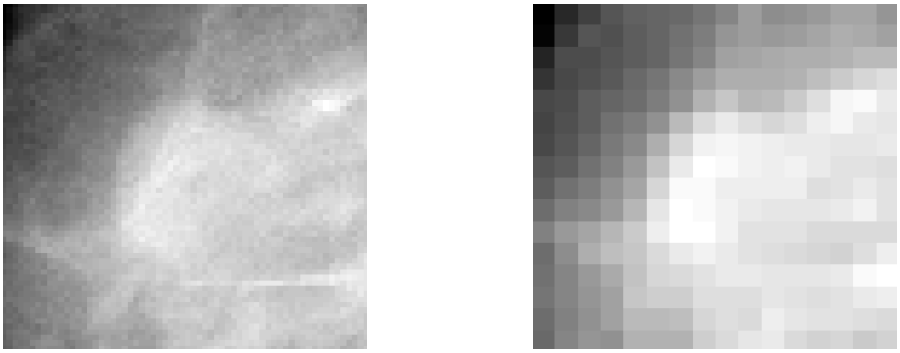


**Figure 4.5:** Resized pixel–based image representation. Original mass (left). Resized mass (right). The crops resulting from bi–linear resizing are characterized by a lower spatial resolution which provides an effective picture of the brightness distribution of the pixels.

In the rest of this work, the above discussed image representation—also referred to as *resized pixel–based image representation*—will be indicated as **PixR**. As for the pixel–based image representations discussed above, the pre–fix **Pix** stands for being a pixel–based image representation, whereas the post–fix **R** stands for being bi–linearly resized.

### 4.2.4 Results and discussion

In order to evaluate in detail the pixel–based image representations discussed above, several tests are performed.

First, the original pixel–based image representation—**PixS**—is evaluated for different SVM's kernels, namely linear and polynomial with degree 2 and 3. See Section 2.2.4 for a precise definition of SVM's kernel and for more informations on the most typical kernels used in literature. With this image representation, in particular, SVM is in the situation of classifying images having $64 \times 64$ pixels size and whose correspondent pixels are scaled between $[0, 1]$.

Second, the influence of image resizing is tested by applying bi–linear resizing to the crops. In particular, the original crops having $64 \times 64$ pixel size are resized to $16 \times 16$ pixel size by means of bi–linear resizing. The resized crops are then scaled between $[0, 1]$ and finally classified by using the same SVM's kernels cited above. This image representation—characterized by both resizing and scaling—will be referred to as **PixRS** in the following.

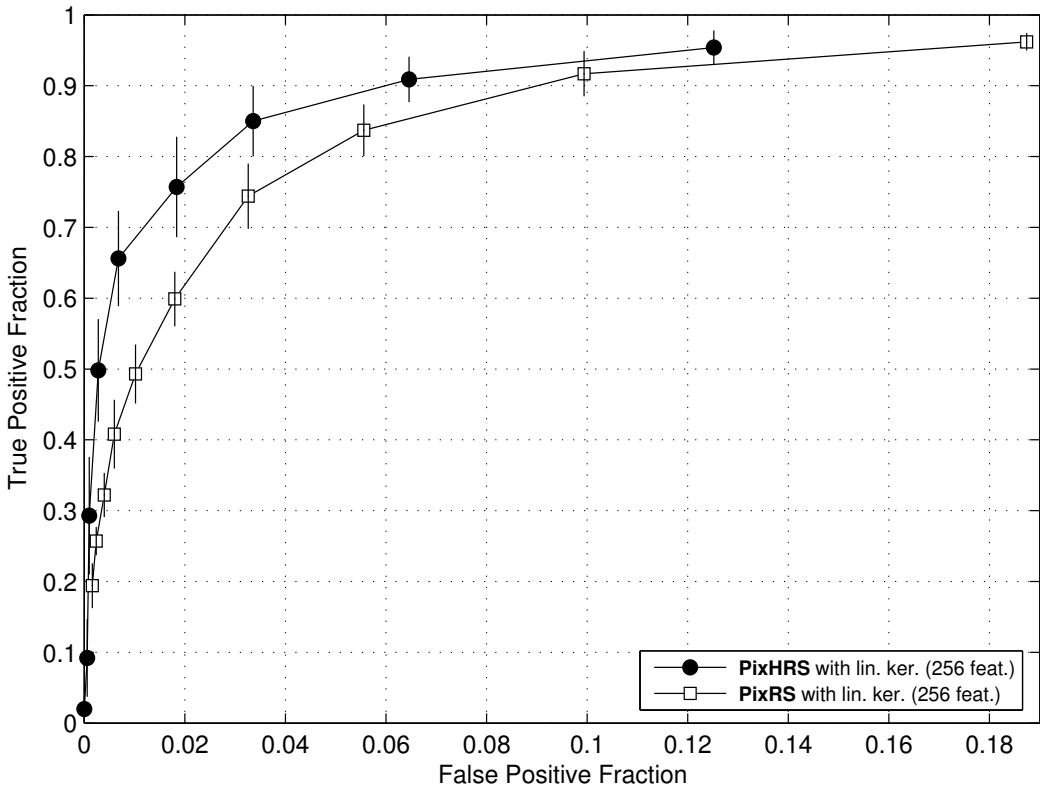



**Figure 4.6:** ROC curves obtained by using pixel–based image representations. The best performances are achieved by **PixHRS**, namely crops processed by means of histogram equalization, bi–linear resizing and scaling. Good performances are also achieved by **PixRS**, namely crops processed by means of bi–linear resizing and scaling. An SVM's linear kernel is used.

Third, histogram equalization is explored. In particular, the original crops having $64 \times 64$ pixel size are all processed by means of histogram equalization. The obtained crops are resized to $16 \times 16$ pixel size by means of bi–linear resizing, scaled between $[0, 1]$ and—finally—classified by using the same SVM's kernels cited for the two cases described above. This image representation—characterized by histogram equalization, resizing and scaling—will be referred to as **PixHRS** in the following.

The results obtained show several interesting aspects which deserve some discussion. Some of them—namely the best ones—are represented in Fig. 4.6.

| | $FPF \sim .01$ | $FPF \sim .02$ | $FPF \sim .03$ | $FPF \sim .04$ | $FPF \sim .05$ |
|---|---|---|---|---|---|
| **PixHRS** | .70 ± .06 | .77 ± .07 | .84 ± .05 | .86 ± .05 | .89 ± .03 |
| **PixRS** | .49 ± .04 | .63 ± .03 | .72 ± .05 | .78 ± .03 | .82 ± .04 |

**Table 4.1:** Classification results comparison. The $TPF$ values obtained by the best performing pixel–based image representations are shown, in particular for $FPF$ values approximately equal to .01, .02, .03, .04 and .05.

First, experiments show that crops resizing has not a tangible effect on the classification performances. This means that the classification results achieved by the original pixel–based image representation—**PixS**—and its correspondent bi–linear resized version—**PixRS**—are practically the same. This is an important result, since it demonstrates that similar results can be achieved by using $16 \times 16 = 256$ features instead of $64 \times 64 = 4096$, thus sensibly reducing the computational times. In particular—due to that similarity between the performance achieved by **PixS** and **PixRS**—only the ROC curve correspondent to the faster image representation is plotted in Fig. 4.6, namely the latter.

Second, the tests performed demonstrate that histogram equalization has a very positive effect on the classification performances. In particular, the original crops processed by means of histogram equalization, bi–linear resizing and scaling of correspondent pixels are those achieving the best classification results. It is evident from Fig. 4.6, in fact, that the ROC curve correspondent to this image representation—namely **PixHRS**—is significantly better than that correspondent to **PixRS**, particularly for $FPF$ values comprised between .01 and .04. Notice, furthermore, that—as for the above discussed case—here the number of features is equal to $16 \times 16 = 256$.

Third, the SVM's kernel which performs globally better is the linear. This is reasonable since—working with pixel–based image representations—correlations among correspondent pixels are much more reliable as features than correlations among distant pixels, see (Schölkopf *et al.*, 1998). In the case of linear kernel, in particular, the correlations considered are those among correspondent pixels, namely the inner products computed are $K(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} \cdot \boldsymbol{y})^1$, where $\boldsymbol{x}$ and $\boldsymbol{y}$ are two vectors containing the pixels of two images.

Finally, in order to to give also some quantitative results—other than a detailed ROC curve analysis—the $TPF$ values achieved by **PixHRS** and **PixRS** for FPF values close to .01, .02, .03, .04 and .05, are shown in Tab. 4.1.

## 4.3 Wavelets Performance

The main motivation for evaluating wavelet–based image representations is that they go—in some sense—in the direction of histogram equalization, namely in the direction of enhancing edges and boundaries of the image under study. As discussed in Section 3.2, in fact, wavelets are specifically suited to capture the shape and the interior structure of objects into images. The reason for that is their ability in encoding the difference in average intensity between local regions along different orientations in a multi–scale framework. In this sense, a strong response from a particular wavelet indicates the presence of an intensity difference at that location in the image—namely an edge or a boundary—whereas a weak response indicates a uniform area.

In the following, the two wavelet–based image representations evaluated in this work will be described. The multi–resolution discrete Haar wavelet transform will be first discussed. Its redundant version—namely multi–resolution overcomplete Haar wavelet transform—will be then considered. The motivation for testing both those two wavelet–based image representations is mainly related to the willingness of exploring the classification performances of SVM while the spatial resolution of the transformed crops is varied. It is well worth reminding, in fact, that—as discussed in Section 3.2.3—for the multi–resolution discrete Haar wavelet transform the number of pixels in the analyzed image is equal to that of the original image. On the contrary, for the multi–resolution overcomplete Haar wavelet transform the number of pixels in the analyzed image is redundant, typically twice as the number of pixels in the original image. As it is evident from Fig. 4.7 and Fig. 4.8, this different number of resulting pixels does not represent only a simple difference of dimensions, but it influences rather sensibly also the spatial resolution of the transformed image. In this sense, the evaluation of the two wavelet–based image representations presented here is motivated by the tentative of understanding how their different spatial resolution influence the performances.

Finally, it is well worth noticing that—as for the pixel–based—the wavelet–based image representations described here constitute a *novel* approach to mammographic mass classification. On the other hand, however, some past works have addressed different problems—such as pedestrian, car and face detection—by using a similar featureless approach based on redundant wavelet dictionaries. The most interesting works on that topic are probably those developed by the MIT Artificial Intelligence Laboratory, namely (Papageorgiou, 1997; Oren *et al.*, 1997; Papageorgiou *et al.*, 1998a,b; Papageorgiou & Poggio, 1999a,b).

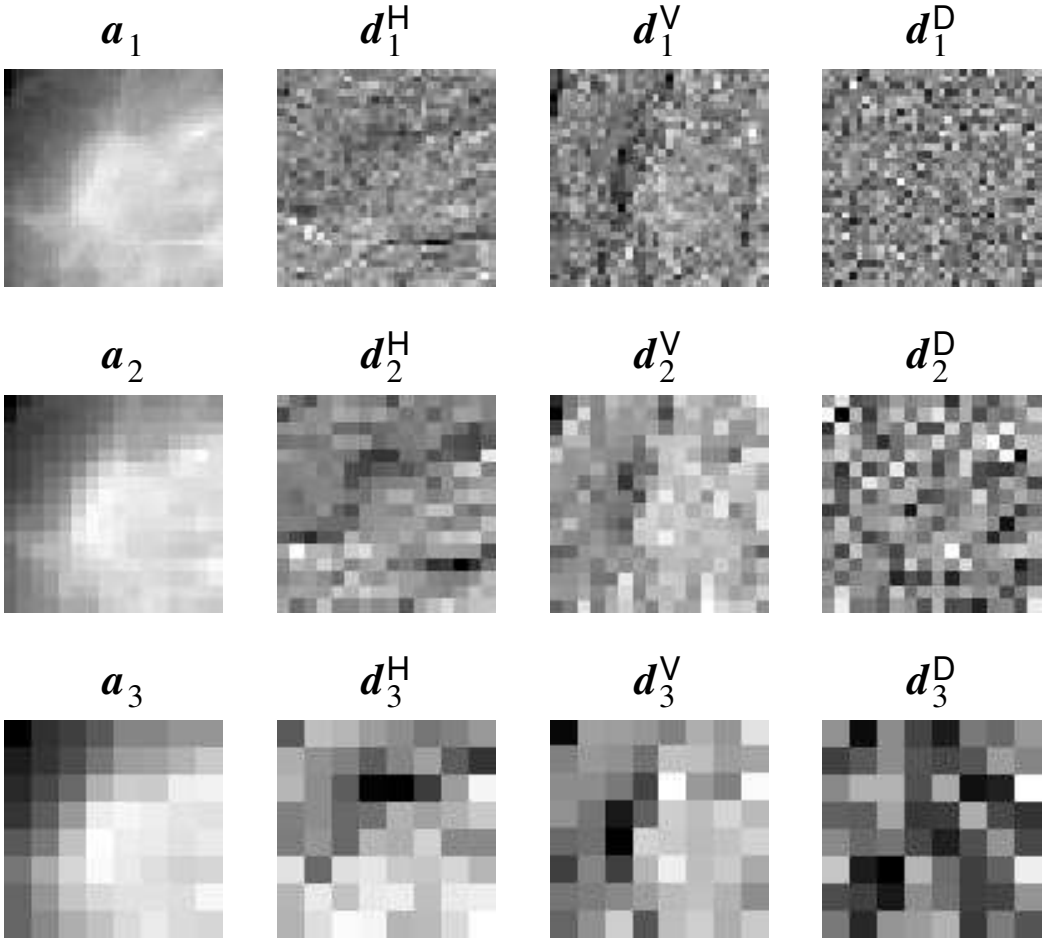



**Figure 4.7:** Multi–resolution discrete Haar wavelet transform. Three decomposition levels are shown, one for each row. In particular, for each level $j = 1, 2, 3$, the approximation component $\boldsymbol{a}_j$, together with the horizontal detail $\boldsymbol{d}_j^{\mathsf{H}}$, the vertical detail $\boldsymbol{d}_j^{\mathsf{V}}$ and the diagonal detail $\boldsymbol{d}_j^{\mathsf{D}}$ are depicted. Notice that all images have undergone *pixel replication*—as discussed in Section 3.1.2—for displaying purposes.

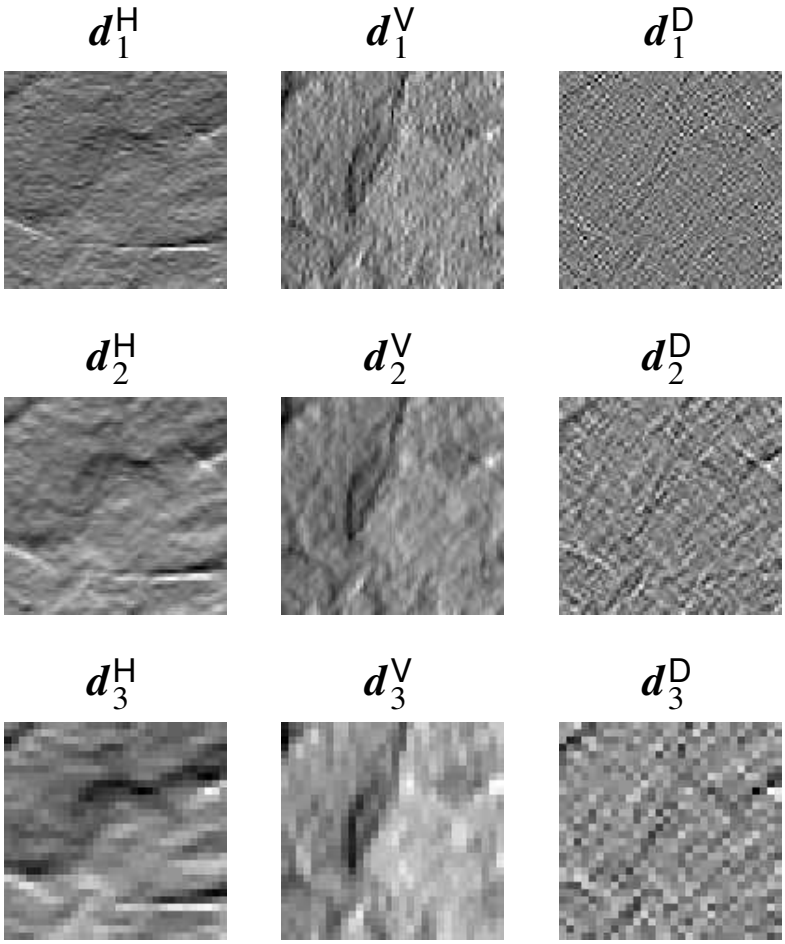



**Figure 4.8:** Multi–resolution overcomplete Haar wavelet transform. Three decomposition levels are shown, one for each row. In particular, for each level $j = 1, 2, 3$, the horizontal detail $\boldsymbol{d}_j^{\mathsf{H}}$, the vertical detail $\boldsymbol{d}_j^{\mathsf{V}}$ and the diagonal detail $\boldsymbol{d}_j^{\mathsf{D}}$ are depicted. Here the approximation components are not shown, since for the multi–resolution overcomplete wavelet transform they are generally characterized by visual artifacts, in particular for decomposition levels higher than one. For that reason in the rest of this work they will be ignored. Notice that all images have undergone *pixel replication*—as discussed in Section 3.1.2—for displaying purposes.

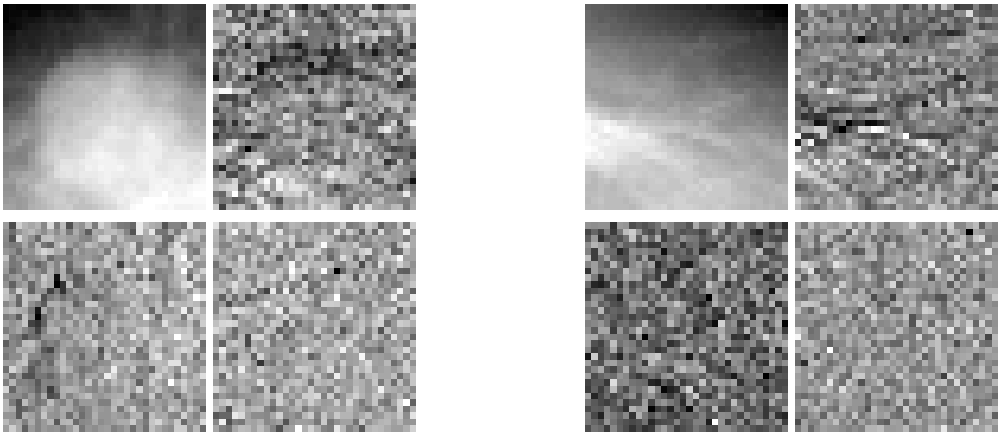


**Figure 4.9:** DWT–based image representation. Mass (left). Non–mass (right). The approximation component $\boldsymbol{a}_j$ (upper–left), together with the horizontal detail $\boldsymbol{d}_j^{\mathsf{H}}$ (upper–right), the vertical detail $\boldsymbol{d}_j^{\mathsf{V}}$ (lower–left) and the diagonal detail $\boldsymbol{d}_j^{\mathsf{D}}$ (lower–right) are depicted for both mass and non–mass. One–level decomposition.

### 4.3.1 DWT–based image representation

In the specific case of this work, when characterized by using an image representation based on multi–resolution discrete Haar wavelet transform, the crops representing tumoral masses and normal tissue will look like as in Fig. 4.9. There, in particular, the result of a multi–resolution discrete Haar wavelet transform is shown for one–level decomposition. Notice that each detail contains informations regarding its specific orientation. For example, by looking carefully at the mass case in Fig. 4.9, it is possible to notice that the horizontal detail gives information about the horizontal edge created by the mass and the surrounding tissue. Similarly behaves the vertical detail. In such a context, the classification features will thus be represented by the $64 \times 64 = 4096$ wavelet coefficients obtained by applying the multi–resolution discrete wavelet transform to the crops up to the first decomposition level.

In the rest of this work, the above discussed image representation—also referred to as *DWT–based image representation*—will be indicated as **DwtS**. In particular, the pre–fix **Dwt** stands for being a DWT–based image representation, whereas the post–fix **S** stands for having correspondent classification features—namely wavelet coefficients—scaled in the interval [0, 1], as previously discussed.

### 4.3.2 OWT–based image representation

Although very efficient from a computational point of view, one of the major drawback for the DWT–based image representation—in this specific problem—is its low spatial resolution. It is in fact evident—for example from the vertical details $\boldsymbol{d}_1^{\mathsf{V}}$, $\boldsymbol{d}_2^{\mathsf{V}}$, $\boldsymbol{d}_3^{\mathsf{V}}$ in Fig. 4.7 and $\boldsymbol{d}_1^{\mathsf{V}}$ in Fig. 4.9—that for high decomposition levels the wavelet details represent edges and boundaries as blunt objects. This is due both to the poor spatial resolution characterizing the original crops—namely their original $64 \times 64$ pixel size—and to the sub–sampling operations performed by the discrete wavelet transform. Notice, in particular, that this tendency to poor spatial resolution is the reason why in this work the wavelet decomposition will be performed only up to the first level for the DWT–based image representation. Higher levels are in fact too little informative to be taken into account.

On the contrary, since for the multi–resolution overcomplete Haar wavelet transform the sub–sampling operations are removed, the resulting image representation is characterized by a richer spatial resolution and the problem is somehow attenuated. Compare, for example, the discrete details $\boldsymbol{d}_1^{\mathsf{V}}$, $\boldsymbol{d}_2^{\mathsf{V}}$, $\boldsymbol{d}_3^{\mathsf{V}}$ in Fig. 4.7 with the correspondent overcomplete ones in Fig. 4.8. It is evident that—in the latter case—the removal of the sub–sampling operations proves to be very useful in order to obtain transformed crops with higher spatial resolution.

Dealing with this image representation, thus, the classification features will be represented by the wavelet coefficients obtained by applying the multi–resolution overcomplete Haar wavelet transform to the crops. In particular, by the selecting the decomposition levels which represent the best compromise between spatial resolution and noise level—namely the fourth and sixth—the crops representing tumoral masses and normal tissue will be characterized by approximately 3000 redundant wavelet coefficients as shown in Fig. 4.10. Notice in particular that—differently from the DWT–based image representation—here the approximation components are disregarded. The problem with the approximation components obtained by the application of the multi–resolution overcomplete Haar wavelet transform is—in fact—that they are affected by some evident visual artifacts that could influence negatively the classification performances.

In the rest of this work, the above discussed image representation—also referred to as *OWT–based image representation*—will be indicated as **OwtS**. As for the DWT–based image representations discussed above, the pre–fix **Owt** stands for its being an OWT–based image representation, whereas the post–fix **S** stands for its being scaled in the interval $[0, 1]$.

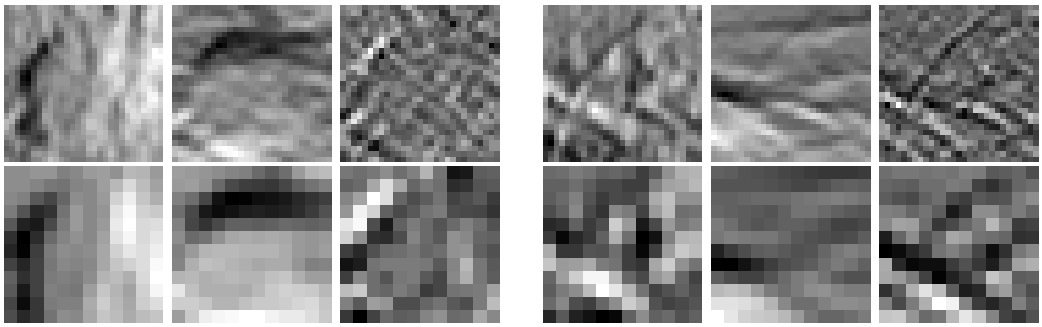


**Figure 4.10:** OWT–based image representation. Mass (left). Non–mass (right). The vertical detail $\boldsymbol{d}_j^{\mathsf{H}}$ (left), the horizontal detail $\boldsymbol{d}_j^{\mathsf{V}}$ (middle) and the diagonal detail $\boldsymbol{d}_j^{\mathsf{D}}$ (right) wavelet coefficients of level 4 (top row) and 6 (bottom row) are shown. Notice that the approximation components are disregarded. The reason is that for the multi–resolution overcomplete Haar wavelet transform they are generally affected by some evident visual artifacts that could influence negatively the classification performances.

### 4.3.3 Results and discussion

The two wavelet–based image representations discussed above are evaluated by performing several tests.

#### DWT–based image representation

As regards the tests performed for the DWT–based image representations, first the original representation—namely **DwtS**—is evaluated. With this image representation, in particular, SVM is asked to classify $64 \times 64 = 4096$ wavelet coefficients obtained by applying the multi–resolution discrete Haar wavelet transform to the crops and by scaling them in the interval $[0, 1]$. An SVM's linear kernel is used.

Second, the influence of histogram equalization is explored by equalizing the crops before transforming them with multi–resolution discrete Haar wavelet transform. In particular, the original crops having $64 \times 64$ pixel size are all processed by means of histogram equalization. The obtained crops are transformed by means of multi–resolution discrete Haar wavelet transform, scaled between $[0, 1]$ and—finally—classified by using an SVM's linear kernel as for the first test. This image representation—characterized by histogram equalization, multi–resolution discrete Haar wavelet transform and scaling—will be referred to as **DwtHS**.

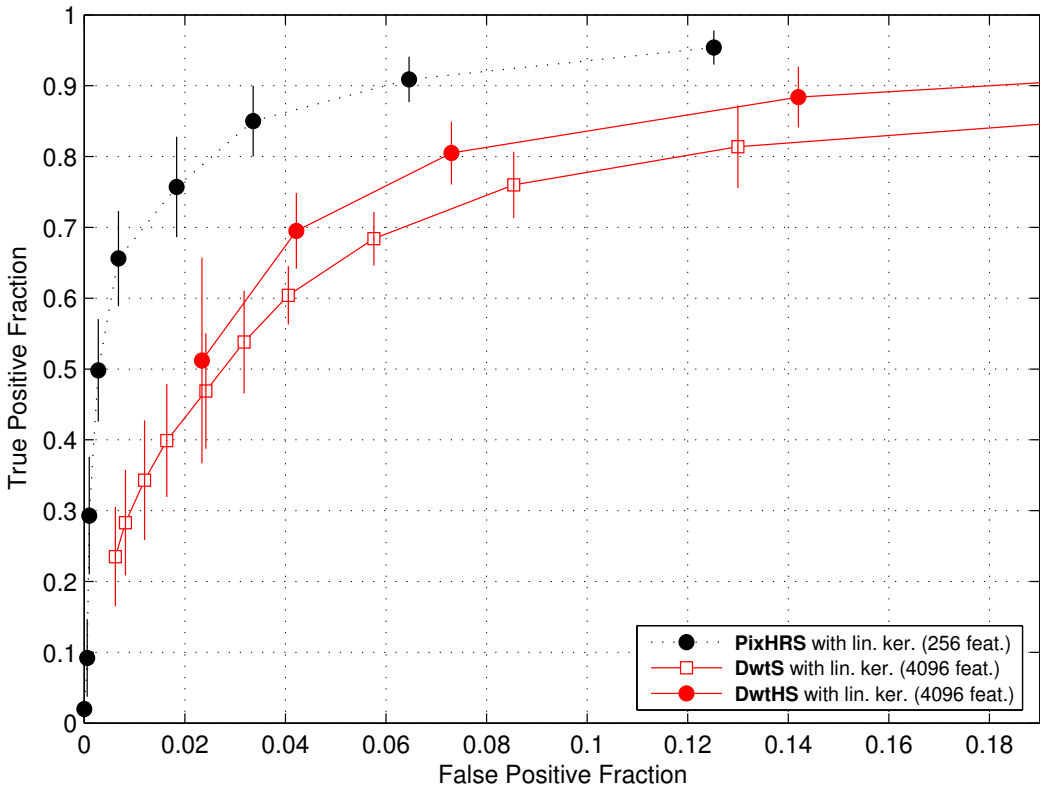



**Figure 4.11:** ROC curves obtained by using wavelet–based image representations, namely DWT–based. Poor performances—with respect to **PixHRS**—are achieved by **DwtHS**, namely crops processed by means of histogram equalization, multi–resolution discrete Haar wavelet transform and scaling. Poor performances are achieved also by **DwtS**, namely crops processed by means of multi–resolution discrete Haar wavelet transform and scaling. An SVM's linear kernel is used.

Third, the effect of a different choice for the SVM's kernel is tested. In particular, other than for the linear kernel, the image representations discussed above are tested for polynomial kernels with degree 2 and 3.

The results obtained with the DWT–based image representation show several interesting aspects. For the sake of clearness, only the best ROC curves are plotted. In particular, Fig. 4.11 shows some results correspondent to the first two tests, whereas Fig. 4.12 to the last one.

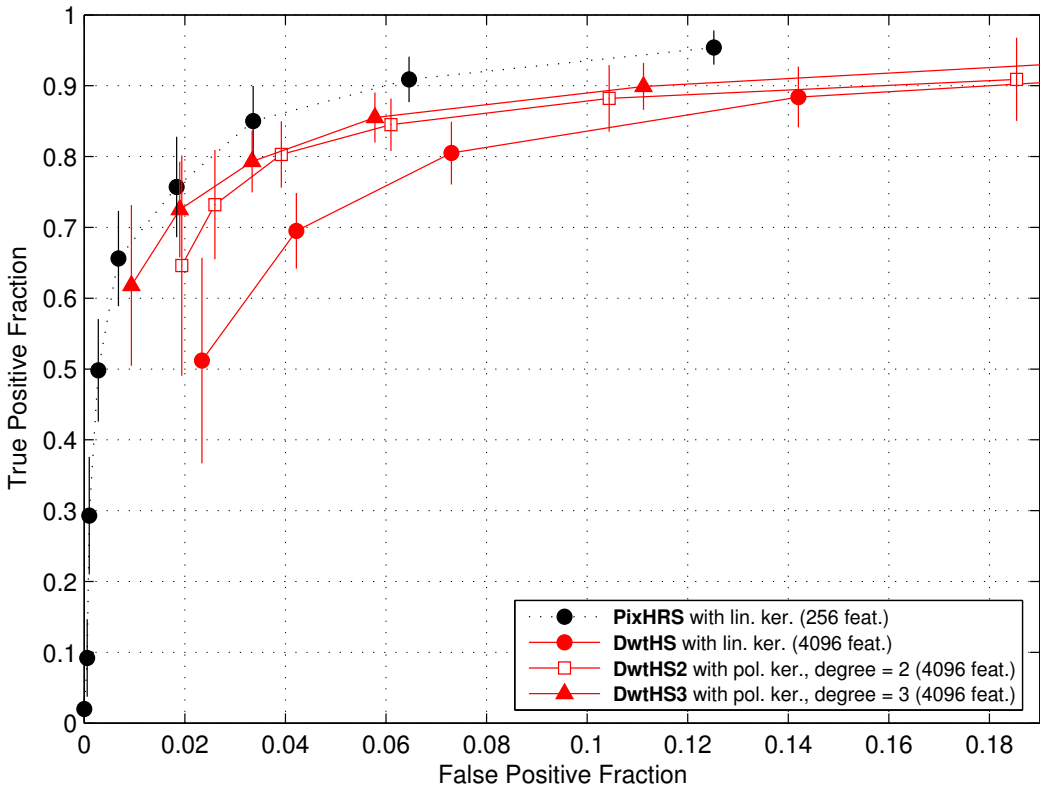



**Figure 4.12:** ROC curves obtained by using wavelet–based image representations, namely DWT–based. Discrete performances—with respect to **PixHRS**—are achieved by both **DwtHS2** and **DwtHS3**, namely crops processed by means of histogram equalization, multi–resolution discrete Haar wavelet transform and scaling. An SVM's polynomial kernel with degree 2 and 3 is respectively used.

First, experiments show that histogram equalization has a positive influence—although slight—on the classification performances. As for the pixel–based image representation, in fact, the original crops processed by means of histogram equalization, transformed by the multi–resolution discrete Haar wavelet transform and finally scaled are those achieving the best classification results. This is evident in Fig. 4.11, where the ROC curve correspondent to this image representation—namely **DwtHS**—proves to be slightly better than that correspondent to **DwtS**, although worse than the best one achieved by the pixel–based image representation. Notice that the number of features for both **DwtHS** and **DwtS** is 4096.

Second, tests demonstrate that the SVM's kernel which performs globally better is the polynomial with degree higher than one, see Fig. 4.12. This result seems to have a logic. Dealing with a wavelet–based image representation, in fact, the vector of features is a concatenation of the approximation and detail components. In particular, each pixel of the original crop is represented four times in the vector of features, namely by one wavelet coefficient in the approximation component and by one wavelet coefficient in each one of the three details. Contrarily to the pixel–based image representation—where correlations among correspondent features are the only important ones—here correlations among distant features are important as well due to the structure characterizing the vectors of features.

### OWT–based image representation

As regards the tests performed for the OWT–based image representations, first the original representation—namely **OwtS**—is evaluated. In this case, the classification features handled by SVM are the approximately 3000 wavelet coefficients obtained by applying the multi–resolution overcomplete Haar wavelet transform to the crops and by scaling them in the interval $[0, 1]$. Linear and polynomial with degree 2 and 3 SVM's kernels are tested.

Second, the influence of histogram equalization is explored by equalizing the crops as for the first test. In particular, the original crops having $64 \times 64$ pixel size are all processed by means of histogram equalization. The obtained crops are transformed by means of multi–resolution overcomplete Haar wavelet transform, scaled between $[0, 1]$ and—finally—classified by using linear and polynomial with degree 2 and 3 SVM's kernels. This image representation—characterized by histogram equalization, multi–resolution overcomplete Haar wavelet transform and scaling—will be referred to as **OwtHS** in the following.

The results achieved by the OWT–based image representations are shown in Fig. 4.13. As for the previous cases, in order to have a plot as clear as possible, only the best results are reported.

Experiments give first some confirmations about the importance of SVM's polynomial kernels with degree higher than one. In particular—evaluating different SVM's kernels—it results that also for the OWT–based image representation a vector of features in which a pixel of the original crop is represented more than once is best classified by means of polynomial kernels with degree higher than one. For that reason, in Fig. 4.13 the ROC curves plotted correspond to tests performed by using SVM's polynomial kernels with degree 2.

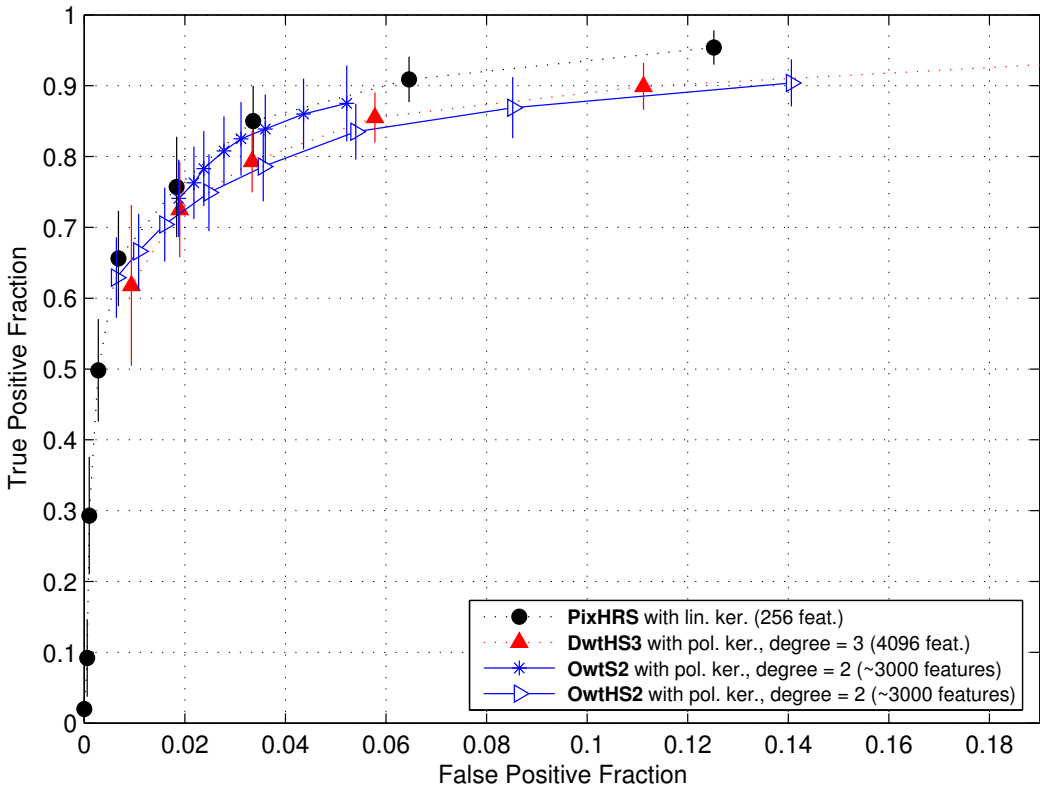



**Figure 4.13:** ROC curves obtained by using wavelet–based image representations, namely OWT–based. Discrete performances—with respect to **PixHRS**—are achieved by **OwtHS2**, namely crops processed by means of histogram equalization, multi–resolution overcomplete Haar wavelet transform and scaling. Good performances are achieved by **OwtS2**, namely crops processed by means of multi–resolution overcomplete Haar wavelet transform and scaling. An SVM's polynomial kernel with degree 2 is used.

Second, the results obtained with regard to the influence of histogram equalization on the classification performances contradict somehow what obtained for the DWT–based image representation. As it is evident from Fig. 4.13, in fact, the ROC curve which corresponds to the crops processed by means of histogram equalization, transformed by the multi–resolution overcomplete Haar wavelet transform and finally scaled—namely **OwtHS2**—proves to be worse with respect to that which corresponds to the crops simply transformed by the multi–resolution overcomplete Haar wavelet transform and finally scaled, in other words **OwtS2**.

| | $FPF \sim .01$ | $FPF \sim .02$ | $FPF \sim .03$ | $FPF \sim .04$ | $FPF \sim .05$ |
|---|---|---|---|---|---|
| **PixHRS** | .70 ± .06 | .77 ± .07 | .84 ± .05 | .86 ± .05 | .89 ± .03 |
| **OwtS2** | - | .75 ± .05 | .82 ± .05 | .85 ± .05 | .87 ± .05 |
| **DwtHS3** | .62 ± .11 | .73 ± .07 | .78 ± .04 | .82 ± .04 | .85 ± .03 |

**Table 4.2:** Classification results comparison. The $TPF$ values obtained by the best performing pixel–based, DWT–based and OWT–based image representations are shown, in particular for $FPF$ values approximately equal to .01, .02, .03, .04 and .05.

The reason is probably that the combined effect of histogram equalization together with a redundant wavelet analysis enhances too much the crops, thus encoding in the wavelet coefficients unnecessary and unimportant image details, for instance noise. Notice, furthermore, that the ROC curve corresponding to **OwtHS2** is practically overlapped to that corresponding to **DwtHS3**. In some sense, histogram equalization has the same effect on the classification performances obtained by both using discrete and overcomplete wavelet transform. At the same time, the ROC curve corresponding to **OwtS2** is almost overlapped to that corresponding to **PixHRS**. In particular they represent the best classification performances obtained so far.

To compare quantitatively the best results obtained for the three main image representations tested—namely pixel–based, DWT–based and OWT–based—Tab. 4.2 is presented. Here, the $TPF$ values achieved by **PixHRS**, **DwtHS3** and **OwtS2** are shown for FPF values close to .01, .02, .03, .04 and .05.

The good performances of **PixHRS** are evident. This is quite expected, since as already discussed in detail this image representation in based on both histogram equalization and bi–linear resizing, techniques which are theoretically supposed to separate well tumoral masses from normal tissue. In particular, the combined effect of histogram equalization and bi–linear resizing is to enhance edges and boundaries separating tumoral masses from the surrounding tissue, but—at the same time—also to strongly characterize their central bright nucleus.

On the other hand, the performances of **OwtS2** are good as well. Also for that image representation positive results are quite expected. The wavelet representation allows in fact to capture both the detailed structures and the general shape of tumoral masses. The overcomplete wavelet transform, however, clearly leads to superior performances with respect to the discrete wavelet transform—**DwtHS3**—due to the richer spatial resolution which assures.

## 4.4 Steerable Filters Performance

Although in their *very* preliminary version, in the following Section the tests performed in order to evaluate the steer–based image representation will be briefly presented and discussed. In particular, the motivation for placing this Section between that dealing with the tests performed by using the wavelet–based image representation and that dealing with the tests performed by using the ranklet–based image representation is mainly due to logic and coherence rather than to chronological reasons. In fact, the steer–based image representation has been the last—from a chronological perspective—being implemented and it is currently under evaluation. In this sense, the results presented herein must be considered as a sort of anticipation of a more complete study which is—at the time—still under development.

As for the wavelet–based image representation, the reason for evaluating the steer–based image representation is that it goes in the direction of histogram equalization, thus in the direction of enhancing edges and boundaries of the image under study. Furthermore, due to its redundancy and steering properties, it assures a rich spatial resolution of the transformed image—as for the overcomplete wavelet transform—but with a higher orientation selectivity. Those properties proved to be fundamental in order to achieve good classification performances in the image representations previously discussed.

In the following, the two steer–based image representations evaluated in this work will be described. The classical multi–resolution steerable pyramid will be first discussed. As already described in detail, it mainly consists of a linear transform in which an image is decomposed into a collection of sub–bands localized at different resolutions and steered at several orientations. Fig. 4.14, for example, shows a tumoral mass decomposed by means of a steerable pyramid at three resolutions and six different *fixed* orientations. A further implementation using wedge filters will be also considered. Here, the steering and asymmetric properties of wedge filters are first used in order to localize the angles at which the filters response is maximal. The classical steerable filters are then used in order produce a steerable pyramid at different resolutions and steered at the angles found by the wedge filters, see Fig. 4.15.

Finally, although this approach represents the first example of a featureless technique based on steerable pyramids and SVM for mammographic mass classification, however a similar scheme has been implemented in (Sajda *et al.*, 2002) by using as classifier a Hierarchical Pyramid Neural Network (HPNN).

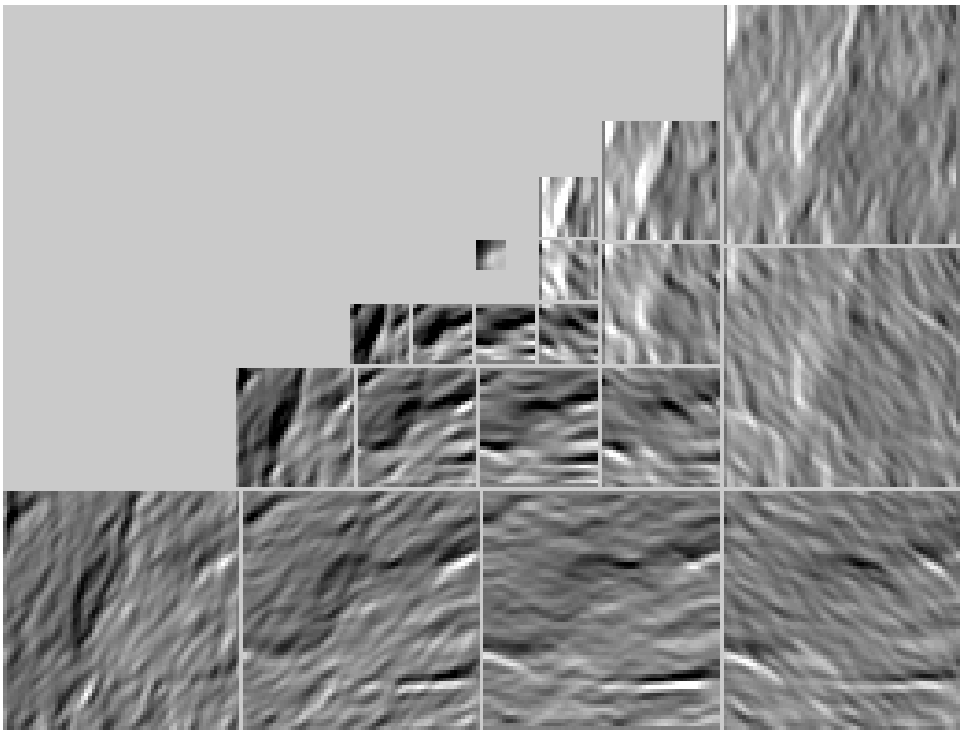


**Figure 4.14:** Steerable pyramid decomposition of a tumoral mass. Five order derivative steerable filters have been used. Shown are the resulting six orientations at three different resolutions and the final low–pass image.

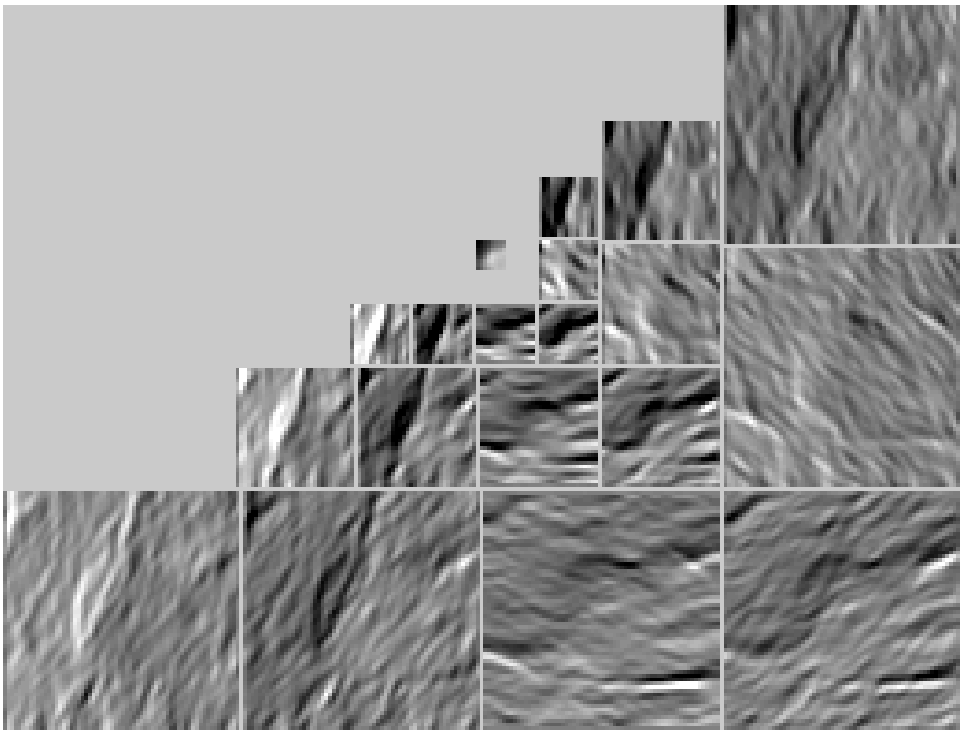


**Figure 4.15:** Steerable pyramid decomposition of a tumoral mass. Five order derivative steerable filters have been used. Shown are the orientations corresponding to the first six maximal responses found by the wedge filters, namely 185 °, 41 °, 117 °, 88 °, 151 ° and 344 ° (from upper–right to lower–left). Three different resolutions and the final low–pass image are also shown.

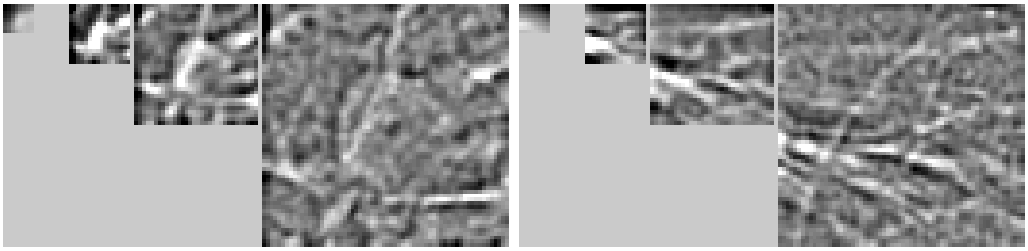


**Figure 4.16:** Steer–based image representation. Mass (left). Non–mass (right). This image representation corresponds to the multi–resolution steerable pyramid obtained by using as filters the zero order derivative of a Gaussian. Three–level, one–angle decomposition.

### 4.4.1 Steer–based image representation

When dealing with the steer–based image representation, the parameters which could influence the classification performances are mainly the number of decomposition levels and the number of fixed angles at which the pyramidal decomposition is performed. In particular, the number of angles is determined by the steerable filters used. As already discussed in Section 3.3, in fact, by changing the derivative order of the steerable filters used, the number of orientations may be adjusted, for example first derivatives yield two orientations, whereas second derivatives yield three orientations and so forth.

In this work—being the testing still under development—the only steerable filters evaluated are those correspondent to a zero order Gaussian derivative. This means a single orientation angle and a maximal number of decomposition levels equal to 3, when dealing with crops having pixels size $64 \times 64$. In particular, when characterized by such an image representation, the crops representing tumoral masses and normal tissue look like as in Fig. 4.16. Here, the first decomposition level is represented by a crop with pixel size $64 \times 64$, the second decomposition level by one with pixel size $32 \times 32$, the third decomposition level by one with pixel size $16 \times 16$ and finally by a low–pass residual with pixel size $8 \times 8$.

In the rest of this work, the above discussed image representation—also referred to as *steer–based image representation*—will be indicated as **SteerS**. In particular, the pre–fix **Steer** stands for being a steer–based image representation, whereas the post–fix **S** stands for having correspondent classification features—namely the coefficients obtained by the steerable pyramid—scaled in the interval $[0, 1]$ as previously discussed.

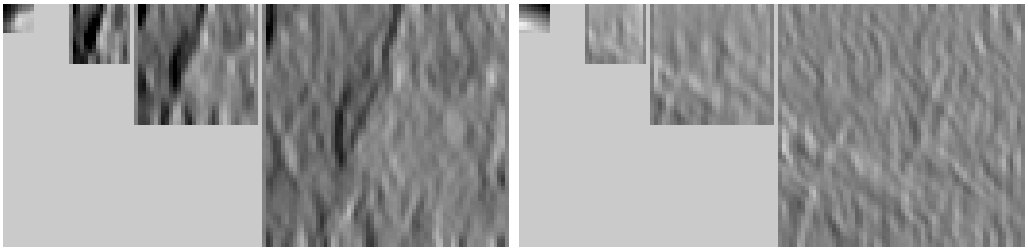


**Figure 4.17:** Steer–based image representation at maximal energy. Mass (left). Non–mass (right). This image representation corresponds to the multi–resolution steerable pyramid obtained by using as filters the five order derivative of a Gaussian oriented at the first maximal response angle found by wedge filters. Three–level decomposition.

### 4.4.2 Steer–based image representation at maximal energy

In the image representation which corresponds to the multi–resolution steerable pyramid at the maximal energy angle, the wedge filters are first used in order to individuate the angles at which their response is maximal. The filters obtained by the five order derivative of a Gaussian are then steered along the directions individuated by the wedge filters. Notice, in particular, that the number of angles for which the response is maximal could differ according to the crop. For this reason, since SVM deals with dimensionally homogeneous vectors, it is necessary to fix to one the number of angles at which the decomposition is performed. In this sense, the parameter which could influence the classification performances is only the number of decomposition levels. When characterized by such an image representation, the crops representing tumoral masses and normal tissue look as in Fig. 4.17. As for steer–based image representation, the first decomposition level is represented by a crop with pixel size $64 \times 64$, the second decomposition level by one with pixel size $32 \times 32$, the third decomposition level by one with pixel size $16 \times 16$ and finally by a low–pass residual with pixel size $8 \times 8$.

In the rest of this work, the above discussed image representation—also referred to as *steer–based image representation at maximal energy*—will be indicated as **SteerMaxS**. In particular, the pre–fix **SteerMax** stands for being a steer–based image representation at maximal energy, whereas the post–fix **S** stands for having correspondent classification features—namely the coefficients obtained by the steerable pyramid—scaled in the interval $[0, 1]$.

### 4.4.3 Results and discussion

In order to evaluate the steer–based image representation discussed above, some tests are performed. Again, being the evaluation in its very preliminary phase, the results presented herein should not be considered as an exhaustive picture of the classification performances of that image representation, but rather as an anticipation of the very first findings.

In the first test the steer–based image representation is evaluated. In particular, the original crops having $64 \times 64$ pixel size are decomposed by means of the multi–resolution steerable pyramid at different levels and steered at the single orientation corresponding to the zero order derivative filters used. The resulting coefficients are then classified by means of several SVM's kernels, namely linear and polynomial with degree 2 and 3.

Second, the steer–based image representation at the maximal energy is evaluated. In this case, the original crops are submitted for instance to the analysis of the wedge filters. Once the angle correspondent to the maximal energy is found for each crop, they are decomposed by means of the multi–resolution steerable pyramid at different levels and steered at each correspondent angle found. Also for this case, the resulting coefficients are then classified by means of several SVM's kernels, namely linear and polynomial with degree 2 and 3.

Experiments confirm that the SVM's kernels which performs globally better are the polynomial ones with degree higher than one, in particular with degree equal to three, see Fig. 4.18. As discussed for the wavelet–based image representation, this result is quite understandable. Dealing with feature vectors in which each original pixel of the crop is represented more than once, in fact, correlations among distant features are important.

The tests performed seem also to demonstrate that the use of the multi–resolution steerable pyramid oriented at the maximal energy angle results in an improvement of the classification performances. Looking at Fig. 4.18, in fact, it is evident that the steer–based image representation at maximal energy performs slightly better than the steer–based one. Nevertheless, the results achieved are sensibly worse than those correspondent to the best image representations found so far. Due to the incompleteness of the tests performed, a precise motivation for that is difficult to find. However, one possible explanation is that the number of features for the steer–based image representations—namely 5440—is much higher than that for the pixel–based and the OWT–based image representations, respectively 256 and 3000. This could result in a harder problem for SVM.

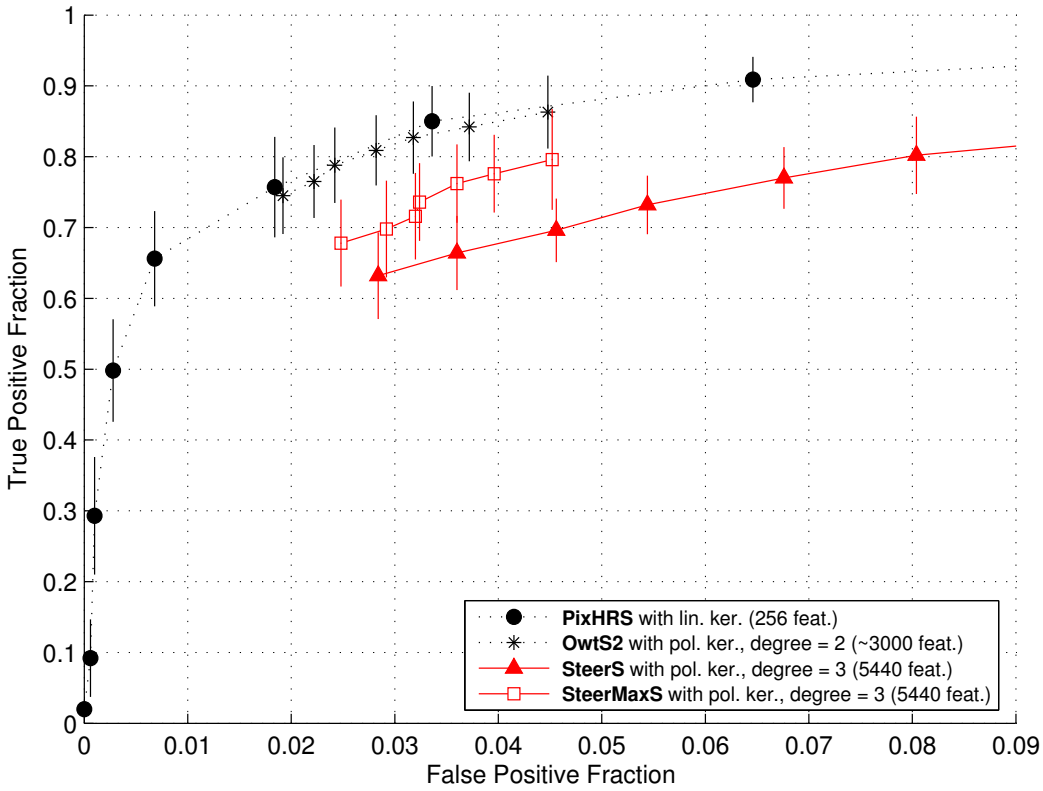



**Figure 4.18:** ROC curves obtained by using steer–based image representations. Poor performances—with respect to both **PixHRS** and **OwtS2**—are achieved by the ROC curve which corresponds to the coefficients first obtained by applying the multi–resolution steerable pyramid and then classified by means of SVM's polynomial kernel with degree 3. Poor performances, even though slightly better, are obtained by the ROC curve which corresponds to the coefficients first obtained by applying the multi–resolution steerable pyramid at maximal energy angle and then classified by means of SVM's polynomial kernel with degree 3.

## 4.5 Ranklets Performance

The main purpose of this Section is to discuss the tests performed by using the ranklet–based image representation. The aim is to understand whether the non–parametric, multi–resolution and orientation selective properties of the multi–resolution ranklet transform can be exploited in order to improve the performance obtained so far for the two–class classification problem under study. To this purpose, it is well worth reminding that—as discussed in Section 3.4—the non–parametric property of the multi–resolution ranklet transform derives from its being mainly based on the rank transform, a transform that—given $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N$ pixels—replaces the value of each $\boldsymbol{p}_i$ with the value of its order among all the other pixels. At the same time, the multi–resolution and orientation selective properties derive from its being mainly modeled on the multi–resolution over-complete Haar wavelet transform in two dimensions. This means that—as for the wavelet transform—the ranklet transform of each crop can be computed at different positions and resolutions by means of a suitable shift and stretch of the Haar wavelet supports. This clearly permits to analyze the crop at several different resolutions, thus allowing the multi–resolution ranklet transform to represent coarse scale features all the way down to fine scale features. Furthermore—for each resolution—the vertical, the horizontal and the diagonal ranklet coefficients can be computed. This clearly allows to analyze the crop at different orientations.

The approach adopted here is once more a featureless approach. In other words, the ranklet coefficients derived from the application of the multi–resolution ranklet transform to the mass crops and to the non–mass crops are directly used as classification features. To this purpose, the multi–resolution ranklet transform of each crop is first performed at different resolutions by shifting and stretching the Haar wavelet supports, see Fig. 4.19. Each crop is then presented to SVM as a collection of several ranklet triplets $R_{\mathrm{V,H,D}}$, each one corresponding to a specific shift and stretch of the Haar wavelet supports. Notice, in particular, that the ranklet–based image representation described here constitutes a *twofold novelty* for mammographic mass classification. The first reason is that featureless approaches have never been applied to tumoral mass classification, as previously discussed for the pixel–based, wavelet–based and steer–based approaches. The second reason is that ranklets have never been applied to medical image processing. As already discussed in Section 3.4, in fact, ranklets have been applied—up to now—almost exclusively to face detection problems, see (Smeraldi, 2002, 2003a; Smeraldi & Rob, 2003; Smeraldi, 2003b).

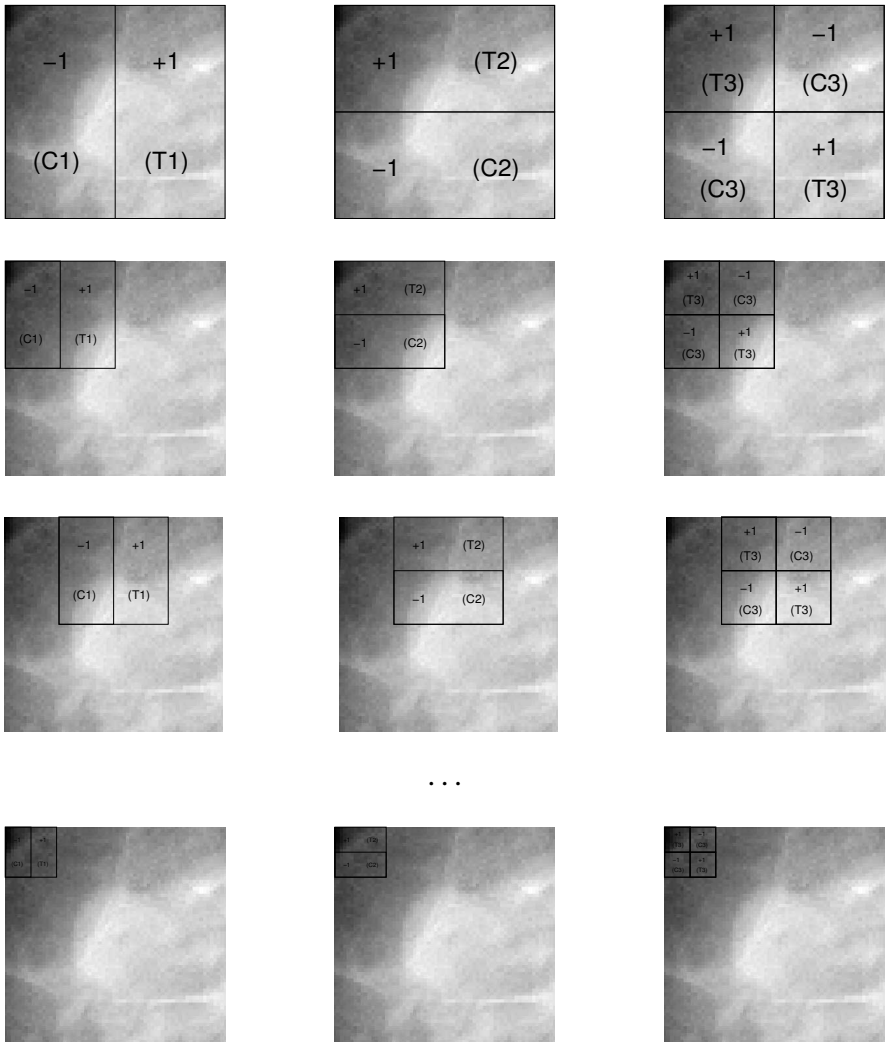



**Figure 4.19:** Multi–resolution ranklet transform. Left, middle and right columns represent respectively how vertical, horizontal and diagonal ranklet coefficients are calculated at different positions and resolutions. In this sense, from each row, a triplet $R_{\mathrm{V,H,D}}$ of ranklet coefficients is computed and presented to SVM.

| Resolutions | Number of ranklet coefficients |
|---|---|
| [16, 14, 12, 10, 8, 6, 4, 2] | 2040 |
| [16, 8, 4, 2] | 1428 |
| [16, 8, 2] | 921 |
| [16, 2] | 678 |
| [16, 4] | 510 |
| [16, 14, 12, 10] | 252 |
| [16, 8] | 246 |

**Table 4.3:** Number of resulting ranklet coefficients obtained by applying the multi–resolution ranklet transform to a crop with pixel size $16 \times 16$. Different combinations of resolutions are shown.

### 4.5.1 Ranklet–based image representation

In order to compute in reasonable times the multi–resolution ranklet transform, crops are for instance required to be resized from the original $64 \times 64$ pixel size to $16 \times 16$ by means of bi–linear resizing.

The number of classification features—ranklet coefficients—passed to SVM then strongly depends on the resolutions at which the multi–resolution ranklet transform is performed. Tab. 4.3 shows the correspondence among the resolutions at which it is performed and the number of ranklet coefficients computed. For example, the multi–resolution ranklet transform of a crop with pixel size $16 \times 16$ at resolutions $[16, 8, 4, 2]$ pixels—namely by using Haar wavelet supports having respectively pixel size $16 \times 16$, $8 \times 8$, $4 \times 4$ and $2 \times 2$—results in 1 triplet $R_{\mathrm{V,H,D}}$ from the resolution at 16 pixels, 81 triplets $R_{\mathrm{V,H,D}}$ from the resolution at 8 pixels, 169 triplets $R_{\mathrm{V,H,D}}$ from the resolution at 4 pixels and 225 triplets $R_{\mathrm{V,H,D}}$ from the resolution at 2 pixels, thus for a total of $3 \times (1 + 81 + 169 + 225) = 1428$ ranklet coefficients. Notice that, the lower is the linear dimension of the Haar wavelet support, the higher is the resolution at which the multi–resolution ranklet transform is performed and so the number of ranklet coefficients produced. And vice versa. This is consistent with the expression discussed in Eq. 3.26.

In the following, the above discussed image representation will be indicated as **RankS**, regardless of the resolutions at which the multi–resolution ranklet transform is performed. Here, the pre–fix **Rank** stands for its being a ranklet–based image representation, whereas the post–fix **S** for its having features—namely ranklet coefficients—scaled in the interval $[-1, 1]$. Notice that this last property is automatically assured by the definition of ranklet coefficients given in Eq. 3.25.

### 4.5.2 Results and discussion

In order to evaluate the performances of the ranklet–based image representation, three main experiments are carried out.

The first test is intended to understand the influence of the SVM's kernel on the classification performances. To this aim, the original crops are for instance resized from their original $64\times64$ pixel size to $16\times16$ by means of bi–linear resizing. Using—as image representation—the ranklet coefficients resulting from the multi–resolution ranklet transform of the resized crops at resolutions $[16, 8, 4, 2]$ pixels, several SVM's kernels are then varied, namely linear and polynomial with degree 2 and 3. The resulting number of classification features here is 1428.

The second test is intended to comprehend the effects of the multi–resolution property of the ranklet transform on the classification performances. As for the previous test, the original crops are resized from their original $64 \times 64$ pixel size to $16 \times 16$ by means of bi–linear resizing. The multi–resolution ranklet transform is then applied to the resized crops by using several combinations of different resolutions, namely those shown in Tab. 4.3. The number of classification features here varies according to the resolutions at which the analysis is performed.

The last test is intended to investigate the influence of histogram equalization on the performances. This aspect is explored by processing the original crops having $64\times64$ pixel size by means of histogram equalization. The obtained crops are resized to $16\times16$ pixel size, transformed by means of the multi–resolution ranklet transform at resolutions $[16, 8, 4, 2]$ pixels and—finally—classified by SVM. As for the first test, the number of classification features here is 1428.

The results obtained for those three tests are reported in the following. In particular, Fig. 4.20 shows some results about the first test. Fig. 4.21, Fig. 4.22 and Fig. 4.23 are concerned with the second test. Finally, Fig. 4.24 is related to the last test. The ROC curves obtained definitely deserve some discussion.

First, looking at Fig. 4.20, the ranklet–based image representation seems to improve its classification performances in correspondence of increasing values for the polynomial degree of the SVM's kernel. In particular, while the linear SVM's kernel achieves discrete performances, the polynomial SVM's kernels with degree 2 and 3 achieve excellent results. What is particularly worth noticing is that the ROC curves which correspond to the ranklet coefficients obtained by applying the multi–resolution ranklet transform at resolutions $[16, 8, 4, 2]$ pixels—and classified by means of SVM's polynomial kernel with degree 2 and 3—perform better than **PixHRS** and **OwtS2**, namely the best image representations found so far.

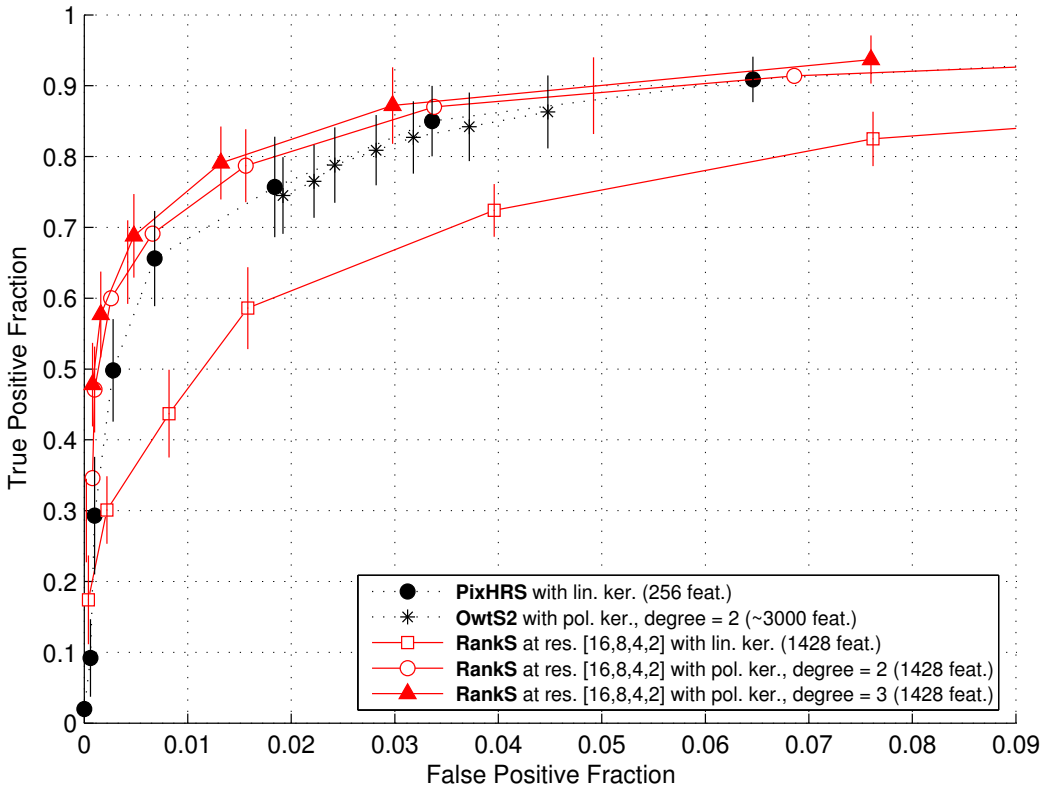



**Figure 4.20:** ROC curves obtained by using ranklet–based image representations. Excellent performances—with respect to both **PixHRS** and **OwtS2**—are achieved by ROC curves which correspond to the ranklet coefficients first obtained by applying the multi–resolution ranklet transform at resolutions $[16, 8, 4, 2]$ pixels and then classified by means of SVM's polynomial kernel with degree 2 and 3. Discrete performances are achieved by using an SVM's linear kernel.

This result is quite expected. As for the wavelet–based image representations, in fact, the vector containing the ranklet coefficients is actually a vector in which each pixel of the original crop—or better each region—is represented more than once. Each region of the original crop is, in fact, analyzed at different resolutions by the ranklet transform and accordingly encoded in the vector of features. As already discussed in Section 4.3.3, in such a situation correlations among distant features prove to be fundamental and—for this reason—SVM's polynomial kernels with degree higher than one perform better.

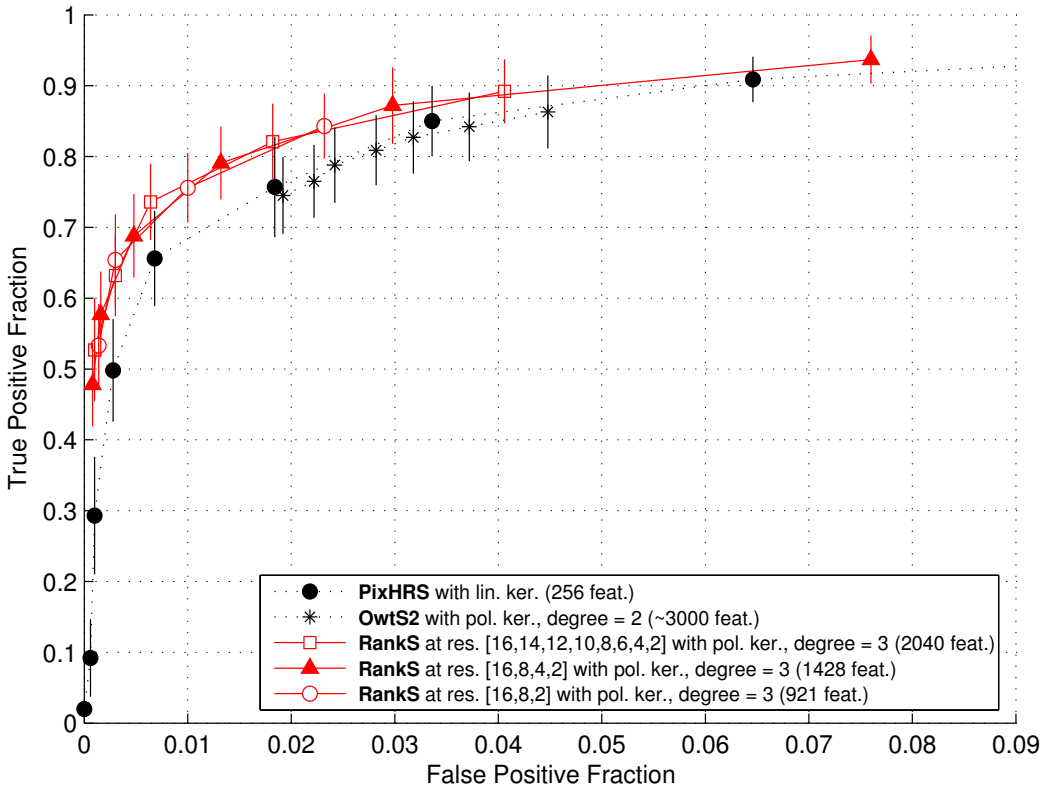



**Figure 4.21:** ROC curves obtained by using ranklet–based image representations. *Low*, *intermediate* and *high* resolutions are taken into account. Excellent performances—with respect to both **PixHRS** and **OwtS2**—are achieved by ROC curves corresponding to the ranklet coefficients obtained by applying the multi–resolution ranklet transform at resolutions $[16, 14, 12, 10, 8, 6, 4, 2]$, $[16, 8, 4, 2]$ and $[16, 8, 2]$ pixels. An SVM's polynomial kernel with degree 3 is used for them.

Second, Fig. 4.21 shows the results obtained employing as image representation the ranklet coefficients resulting from the multi–resolution ranklet transform at resolutions $[16, 14, 12, 10, 8, 6, 4, 2]$, $[16, 8, 4, 2]$ and $[16, 8, 2]$ pixels. It is evident from the ROC curve analysis that all these combinations perform almost identically and they all perform better than **PixHRS** and **OwtS2**. In particular—due to the previous considerations about the choice of SVM's kernel when dealing with ranklet–based features—an SVM's polynomial kernel with degree 3 is used.

This result is quite important, since it demonstrates that using 921 classification features—as for the case [16, 8, 2]—as well as 2040 classification features—as for the case [16, 14, 12, 10, 8, 6, 4, 2]—almost identical performances are achieved. This, in turn, means saving a lot of computational time. Notice, also, that in the tests discussed above all the resolutions are taken into account, as for the [16, 14, 12, 10, 8, 6, 4, 2] case. Or at least a sampled version of them is considered, as for the [16, 8, 4, 2] and [16, 8, 2] cases. In other words, *low*, *intermediate* and *high* resolutions are all *contemplated*.

Third, Fig. 4.22 shows the results obtained using as image representation the ranklet coefficients resulting from the multi–resolution ranklet transform at resolutions [16, 4] and [16, 2] pixels, thus *ignoring* the *intermediate* resolutions. An SVM's polynomial kernel with degree 3 is used as for the previous tests. Looking at the performances, it is evident that they are not essential for classification purposes. In fact, the results obtained for the [16, 4] and [16, 2] cases are only slightly different from those obtained for the [16, 8, 4, 2] case and they all perform better than the **PixHRS** and **Owt2** image representations. As for the tests discussed above, this result demonstrates that using 510 classification features—as for the case [16, 4]—as well as 1428 classification features—as for the case [16, 8, 4, 2]—it is possible to obtain almost identical performances. As discussed above, this result is worthy, since it means avoiding unnecessary waste of time by dealing with a redundant set of features.

Fourth, in Fig. 4.23 the results obtained by using as image representation the ranklet coefficients resulting from the multi–resolution ranklet transform at resolutions [16, 14, 12, 10] and [16, 8] pixels are shown. In this case, the *high* resolutions are *ignored*. Looking at the performances, it is evident that they are important for classification purposes. In fact, the results achieved by the [16, 14, 12, 10] and [16, 8] cases perform worse than those achieved by the [16, 8, 4, 2] case and by **PixHRS** and **OwtS2**. In particular, here—as for the tests discussed above—an SVM's polynomial kernel with degree 3 is used.

Finally, in Fig. 4.24 the results obtained with regard to the influence of histogram equalization on the classification performances are shown. It is quite evident that the ROC curve which corresponds to the crops processed by means of histogram equalization and that which corresponds to the crops non equalized are almost overlapping. This result is once again really important, since it demonstrates that a computational expensive procedure as histogram equalization is generally ineffective—when dealing with ranklet coefficients—in order to improve the classification results.

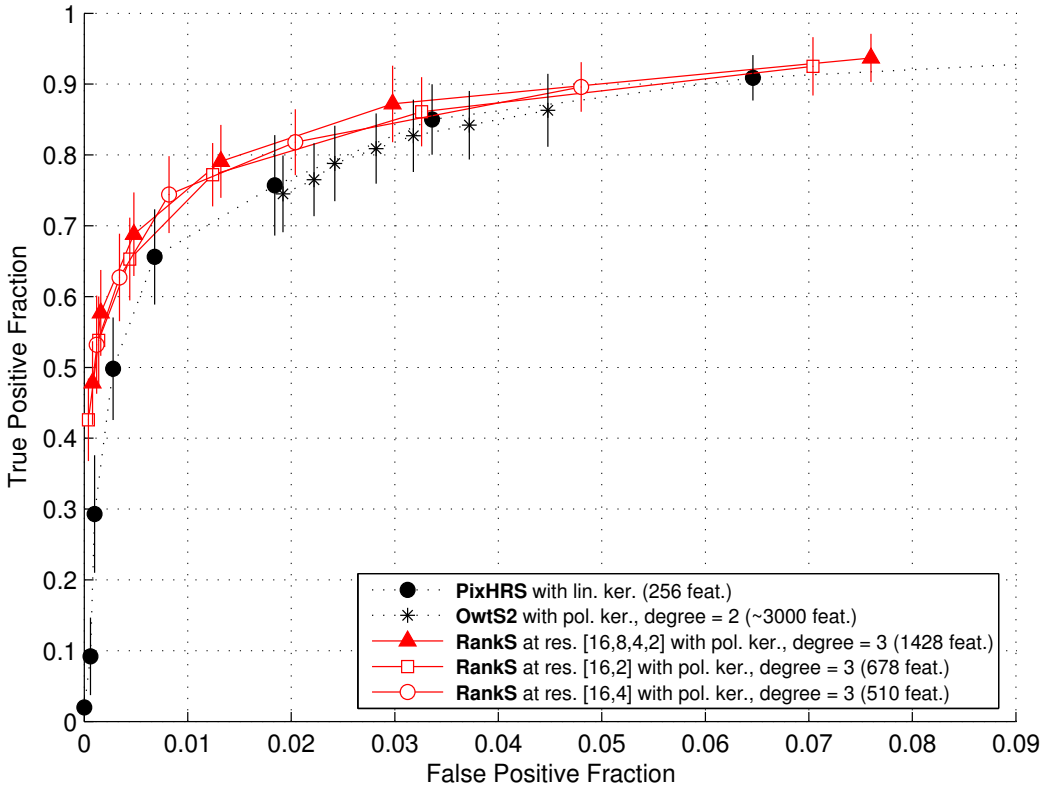



**Figure 4.22:** ROC curves obtained by using ranklet–based image representations. *Low* and *high* resolutions are taken into account. *Intermediate* resolutions are ignored. Excellent performances—with respect to both **PixHRS** and **OwtS2**—are achieved by ROC curves corresponding to the ranklet coefficients obtained applying the multi–resolution ranklet transform at resolutions $[16, 4]$ and $[16, 2]$ pixels. An SVM's polynomial kernel with degree 3 is used for them.

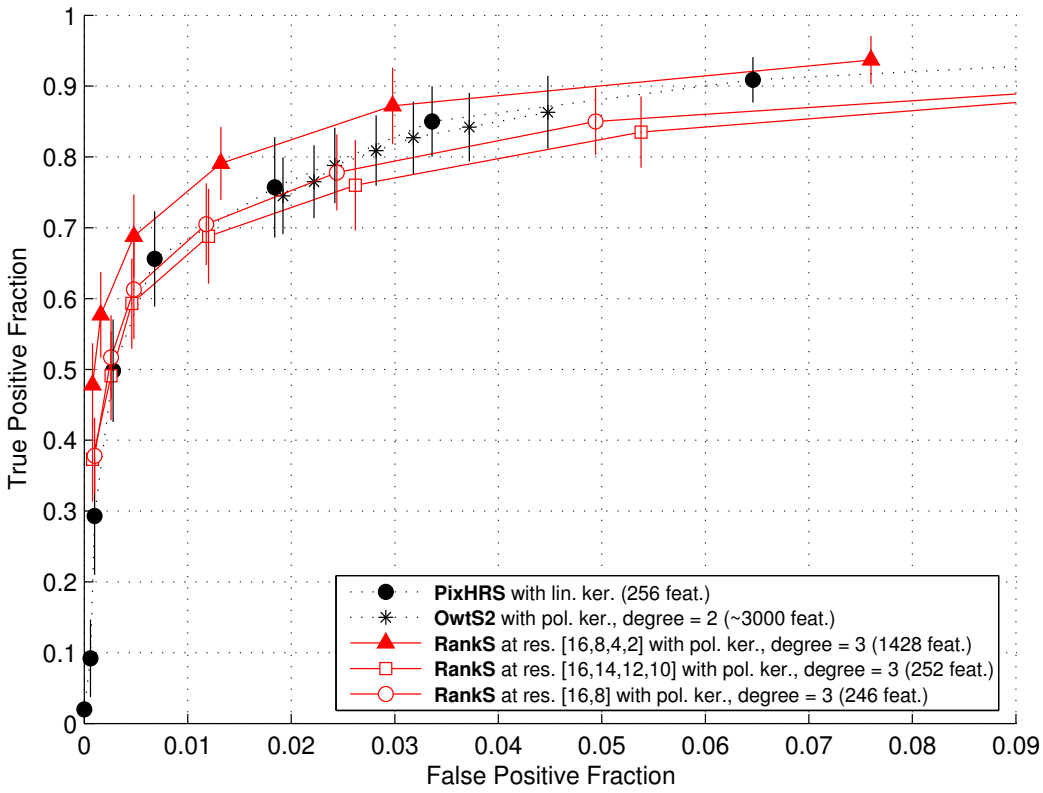



**Figure 4.23:** ROC curves obtained by using ranklet–based image representations. *Low* and *intermediate* resolutions are taken into account. *High* resolutions are ignored. Discrete performances—with respect to **PixHRS** and **OwtS2**—are achieved by ROC curves corresponding to the ranklet coefficients obtained by applying the multi–resolution ranklet transform at resolutions $[16, 14, 12, 10]$ and $[16, 8]$ pixels. An SVM's polynomial kernel with degree 3 is used for them.

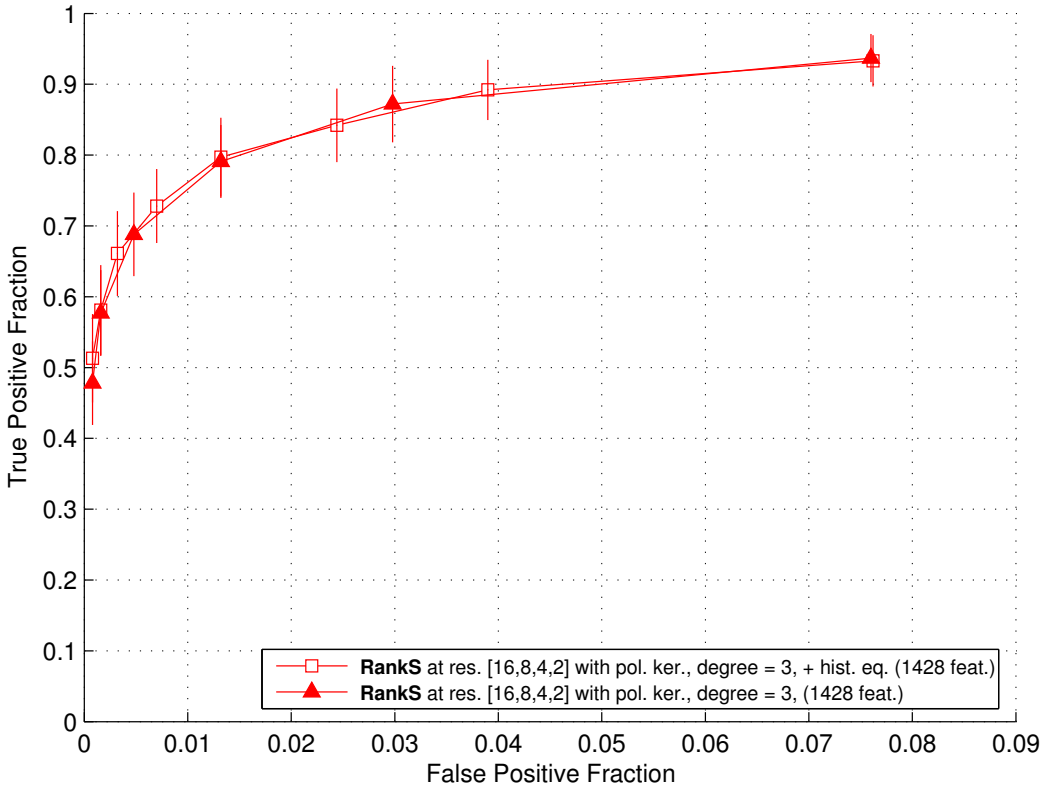



**Figure 4.24:** ROC curves obtained by using ranklet–based image representations. Histogram equalization is tested. Excellent performances are achieved by both ROC curves, namely that correspondent to the equalized crops and that correspondent to the non equalized crops. An SVM's polynomial kernel with degree 3 is used for them.

| | $FPF \sim .01$ | $FPF \sim .02$ | $FPF \sim .03$ | $FPF \sim .04$ | $FPF \sim .05$ |
|---|---|---|---|---|---|
| **RankS3** | .76 ± .05 | .82 ± .05 | .87 ± .05 | .89 ± .05 | .91 ± .04 |
| **PixHRS** | .70 ± .06 | .77 ± .07 | .84 ± .05 | .86 ± .05 | .89 ± .03 |
| **OwtS2** | - | .75 ± .05 | .82 ± .05 | .85 ± .05 | .87 ± .05 |
| **DwtHS3** | .62 ± .11 | .73 ± .07 | .78 ± .04 | .82 ± .04 | .85 ± .03 |

**Table 4.4:** Classification results comparison. The $TPF$ values obtained by the best performing pixel–based, DWT–based, OWT–based and ranklet–based image representations are shown, in particular for $FPF$ values approximately equal to .01, .02, .03, .04 and .05.

In order to give also some quantitative results, the $TPF$ values of the best performing image representations discussed so far are shown in Tab. 4.4, in particular for $FPF$ values close to .01, .02, .03, .04 and .05. The results obtained by the best pixel–based image representation **PixHRS**, OWT–based image representation **OwtS2** and DWT–based image representation **DwtHS3** are compared to the best ranklet–based image representation. In particular—as regards the best ranklet–based image representation—the results reported are those achieved by the ranklet coefficients produced by the multi–resolution ranklet transform at resolutions [16, 8, 4, 2] pixels and classified by means of an SVM's polynomial kernel with degree 3, for the sake of brevity **RankS3**.

The reasons for choosing **RankS3** have been somehow already anticipated when the tests performed with ranklets have been presented. However, it is well worth summarizing them.

First, the results achieved demonstrate that—when dealing with the ranklet–based image representation—the SVM's polynomial kernels with degree higher than one achieve excellent performances, for instance SVM's polynomial kernel with degree 3.

Furthermore, the *low* and *high* resolutions at which the multi–resolution ranklet transform is performed prove to be quite important in order to achieve good performances, whereas *intermediate* resolutions can be ignored without sensibly affecting the classification results. These considerations suggest to perform the multi–resolution ranklet transform at resolutions [16,4] or [16,2] pixels, thus ignoring the intermediate resolutions. Or at least to perform it at resolutions [16,8,4,2] or [16,8,2] pixels, thus using a sampled version of all the resolutions and achieving slightly better performances. In either cases the main idea is to use a reduced number of ranklet coefficients, namely only those influencing the classification performances.

### 4.5.3 Ranklet coefficients reduction by means of SVM–RFE

As already discussed in Section 2.3, one of the most challenging task—when facing a classification problem—is to reduce the dimensionality of the feature space by finding a restricted number of features which influence most the classification performances. The importance of that is twofold. First, finding a smaller sub–set of features which are particularly influent on the classification performances actually results in having smaller training and test sets, thus in lower computational times. This clearly proves to be fundamental when developing algorithms which must be suited for real–time working, as for instance medical applications. Second, the so–called *curse of dimensionality* from statistics theory asserts that the difficulty of an estimation problem increases drastically with the dimension of the space. In such a sense, it is not unusual that a classifier benefits from feature space dimensionality reduction.

In order to study whether and how it is possible to reduce the original 1428 ranklet coefficients of **RankS3** to a smaller sub–set of features, SVM–RFE is applied to each fold of the cross-validation procedure used for the previously discussed tests. The iterative procedure adopted is the following:

1. Train SVM for each fold
2. Test SVM for each fold
3. Compute the ranking criterion represented by Eq. 2.72 for each feature in each fold
4. Compute a ranking list, *common* to all folds, by averaging the ranking position of each feature in each fold
5. Remove the feature with the smallest rank in the ranking list

In particular, two aspects of this approach deserve some deeper and careful consideration. First, SVM must be re–trained after each feature elimination. This is reasonable, since the importance of a feature characterized by medium–low importance may be promoted by removing a correlated feature. Second, each fold of the cross–validation is characterized by a different training set. After each training phase, thus, the computation of the ranking criterion leads to a ranking list different for each fold. This in particular means that the feature having smallest ranking is different for each fold. In order to eliminate the same feature from all the training sets, it is thus necessary to compute a ranking list *common* to all folds.

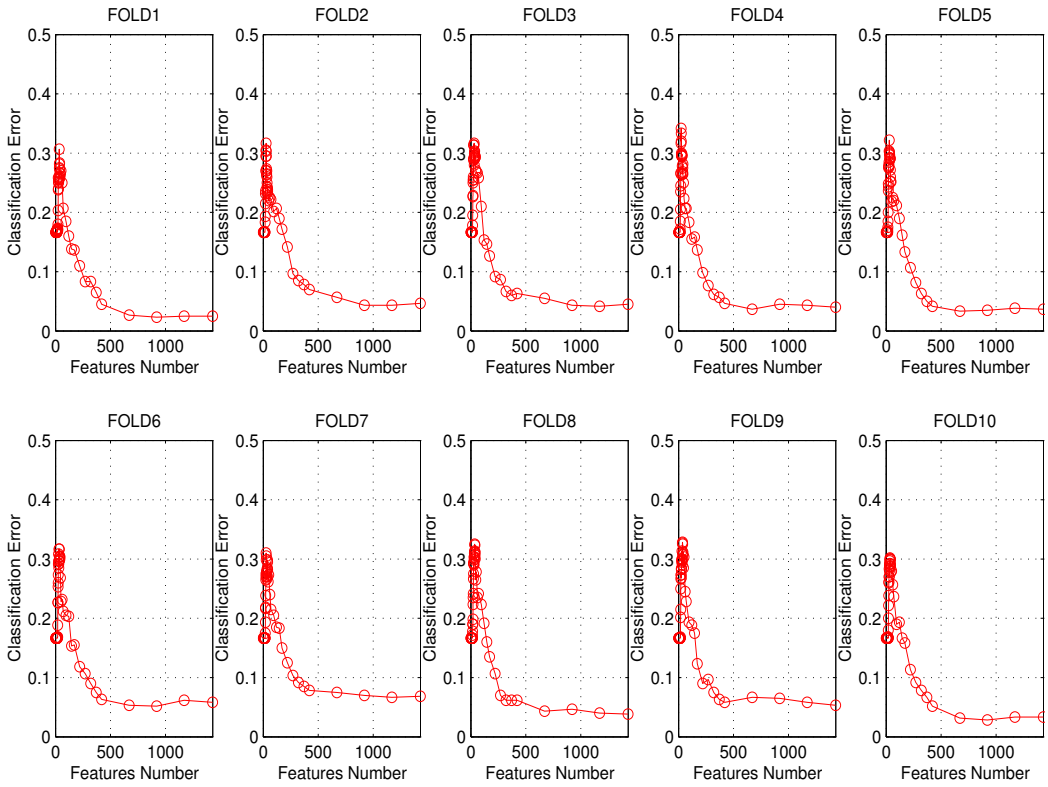



**Figure 4.25:** Application of SVM–RFE to the 1428 ranklet coefficients of the ranklet–based image representation **RankS3**. For each fold of the 10–fold cross validation procedure, the classification error versus the number of features selected by SVM–RFE is plotted. Notice that the number of ranklet coefficients can be sensibly reduced without affecting the classification performances.

This is achieved by averaging the ranking positions of each feature in each ranking list. The feature having the smallest rank in the common ranking list is thus eliminated from all the training sets and the procedure is iterated.

Experiments show that—with this technique—the number of ranklet coefficients can be significantly reduced without affecting the classification performances. It is evident from the results shown in Fig. 4.25, for example, that reducing the number of ranklet coefficients from 1428 down to 1000—or at least 500—the classification error remains practically unaffected.

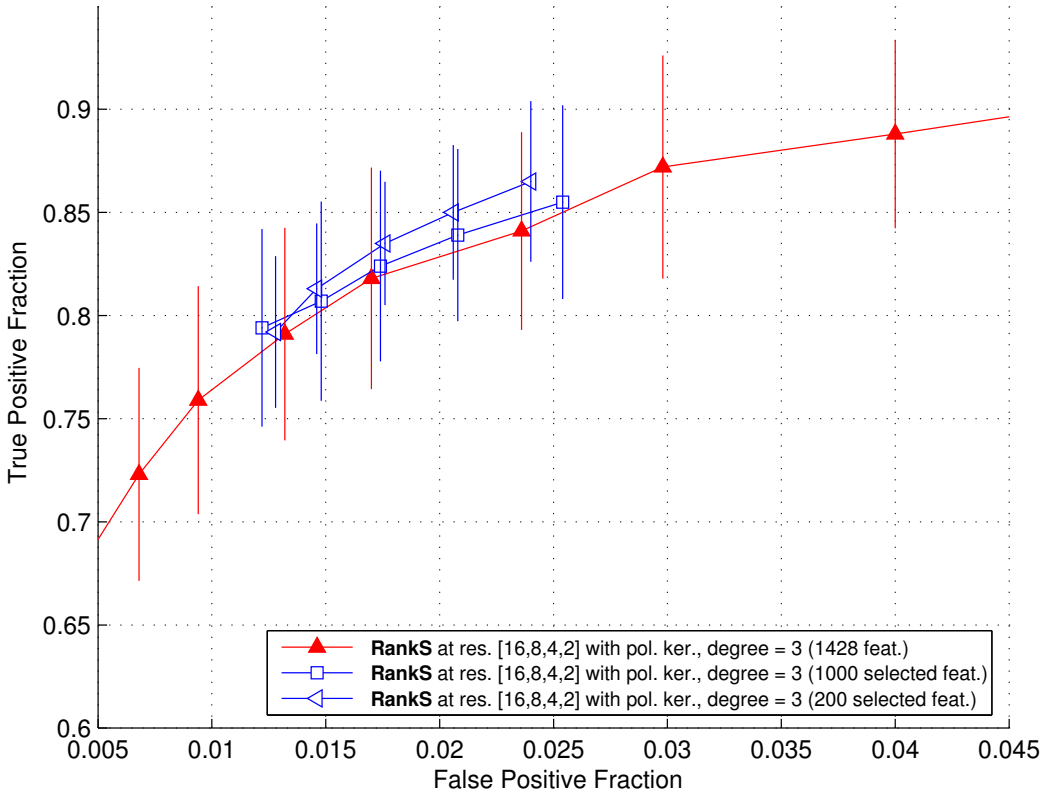



**Figure 4.26:** ROC curves obtained by using ranklet–based image representations in combination with SVM–RFE. Excellent performances—with respect to **RankS3**—are obtained by both ROC curves, namely that correspondent to a reduction of the number of ranklet coefficients from 1428 down to 1000 and that correspondent to a reduction from 1428 down to 200.

Similarly, reducing the number of ranklet coefficients of **RankS3** from 1428 down to 1000—and then down to 200—ROC curves can be generated for each specific *reduced* image representation. Here, a reduced image representation is characterized by a sub–set of features which results from the recursive application of SVM–RFE to the original 1428 ranklet coefficients of **RankS3**. In particular, the results shown in Fig. 4.26 seem to demonstrate that SVM takes some benefit from the reduction of the feature space dimensions, as anticipated in the introductory part of this Section. In fact, the results achieved by the reduced image representations are almost overlapped—or at least slightly better—with respect to those achieved by the original image representation.

Finally, some interesting considerations can be drawn about which ranklet coefficients are the most discriminating ones in this two–class classification problem. To this purpose, it is necessary to look carefully at the ranklet coefficients which survive after the various steps of SVM–RFE.

In Fig. 4.27, Fig. 4.28 and Fig. 4.29—for example—the ranklet coefficients produced by the multi–resolution ranklet transform at resolutions $16 \times 16$, $8 \times 8$, $4 \times 4$ and $2 \times 2$ pixels are shown. In particular, in Fig. 4.27 only the most discriminating 500 ranklet coefficients are shown, in Fig. 4.28 only the most discriminating 300, whereas in Fig. 4.29 only the most discriminating 200. Small green circles represent vertical ranklet coefficients, medium red circles represent horizontal ranklet coefficients and, finally, large blue circles represent diagonal ranklet coefficients. Furthermore, in order to give an idea of the resolutions involved, the gray dashed square represents the dimensions of the Haar wavelet supports.

By looking carefully at the ranklet coefficients calculated at resolutions $2 \times 2$ and $4 \times 4$ which survive after each cut, it is evident that the most discriminant ranklet coefficients are those near the borders of the image, thus those codifying the contour information of the image. That is reasonable, in fact, the main difference between the two classes at fine resolutions is that masses have sharp edges near the borders of the image, whereas normal tissue has not.

On the contrary, as the resolution decreases to $8 \times 8$ and $16 \times 16$, the most important ranklet coefficients are those near the center of the image, thus those codifying the symmetry information of the image, rather that its contour information. That seems to be reasonable too, since at coarse resolutions the main difference is that masses appear approximately as symmetric circular structures centered on the image, whereas normal tissue has a less definite structure.

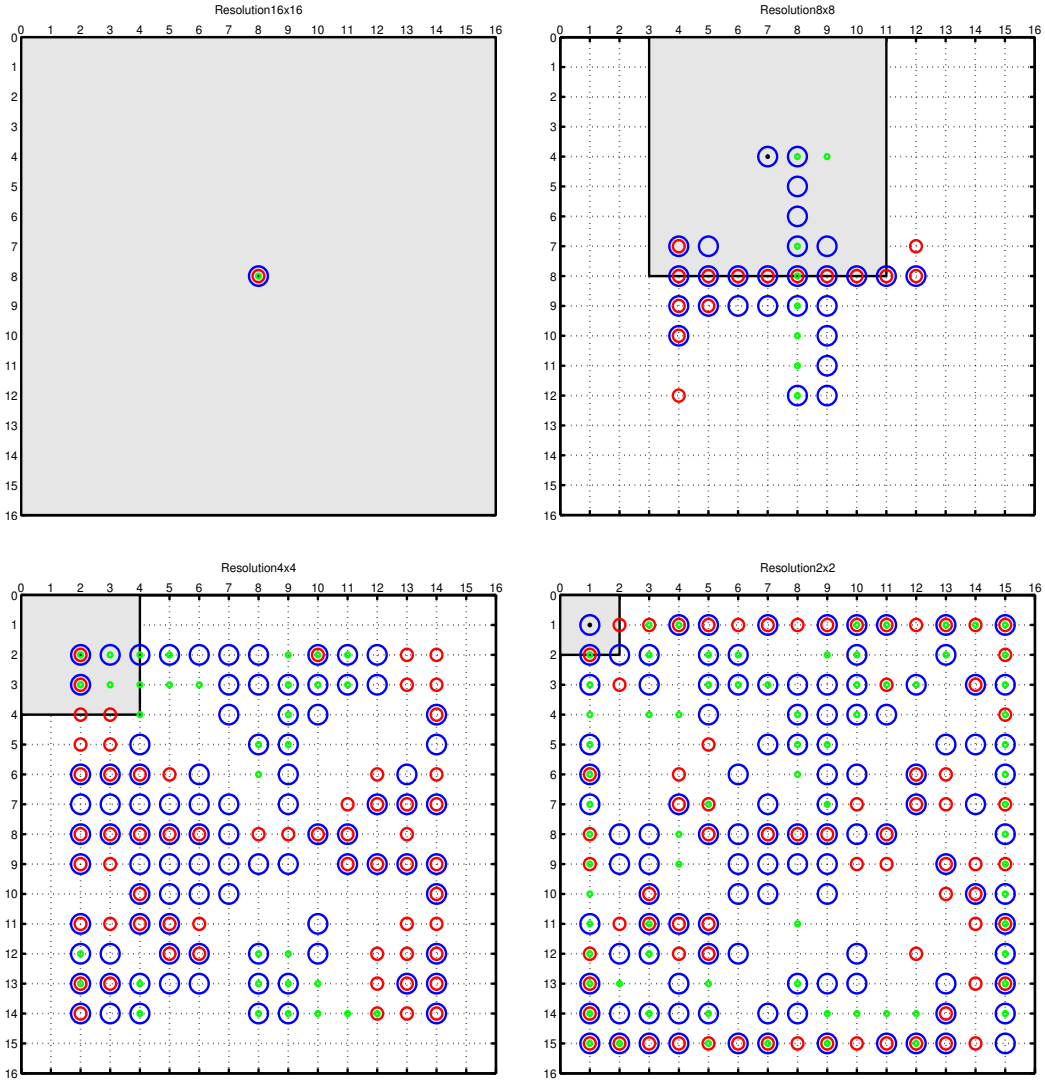



**Figure 4.27:** Ranklet coefficients after SVM–RFE has selected the 500 most relevant ones. Small green circles represent vertical ranklet coefficients, medium red circles represent horizontal ranklet coefficients, large blue circles represent diagonal ranklet coefficients. The gray dashed square represents the dimensions of the Haar wavelet supports. Resolution $16 \times 16$ (upper–left), $8 \times 8$ (upper–right), $4 \times 4$ (lower–left), $2 \times 2$ (lower–right) are represented.

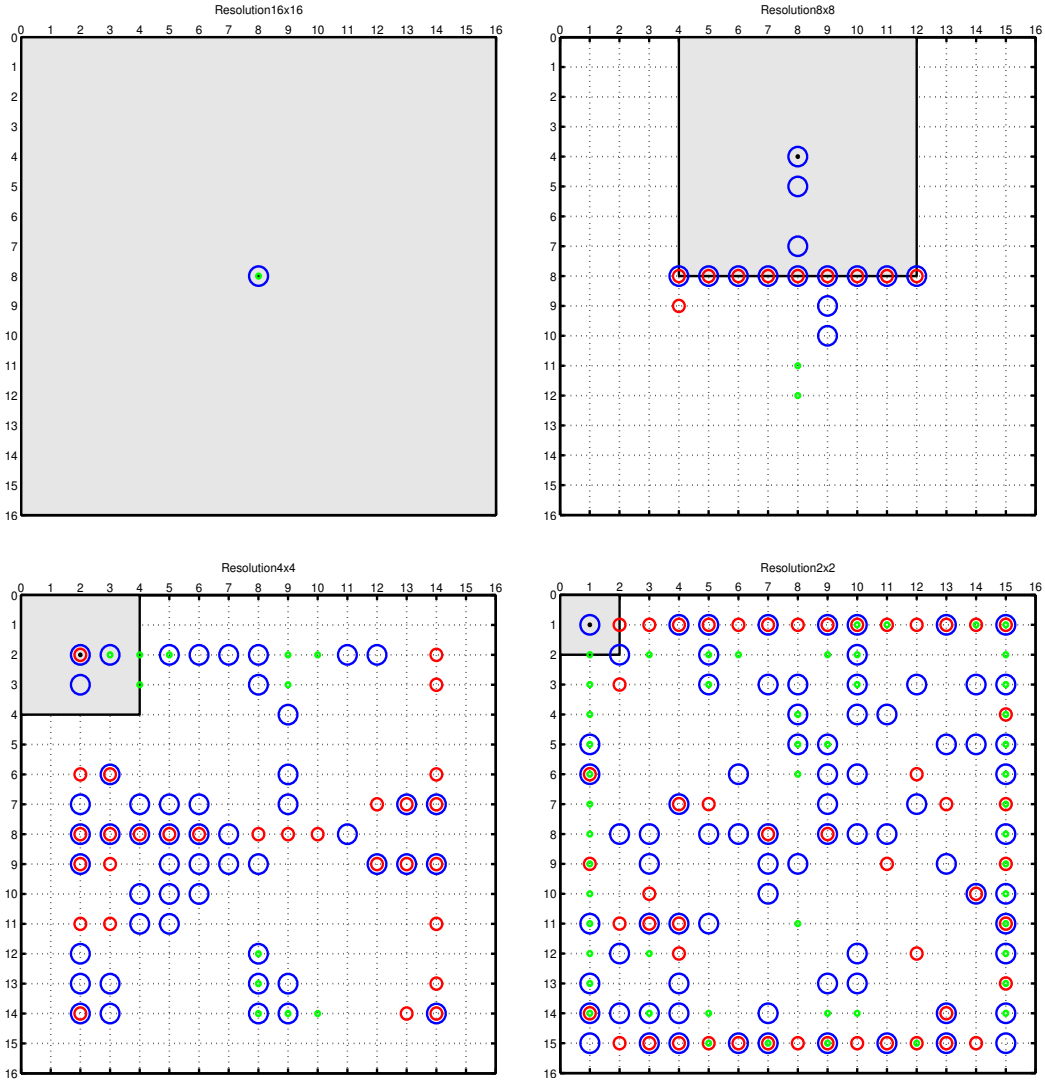



**Figure 4.28:** Ranklet coefficients after SVM–RFE has selected the 300 most relevant ones. Small green circles represent vertical ranklet coefficients, medium red circles represent horizontal ranklet coefficients, large blue circles represent diagonal ranklet coefficients. The gray dashed square represents the dimensions of the Haar wavelet supports. Resolution $16 \times 16$ (upper–left), $8 \times 8$ (upper–right), $4 \times 4$ (lower–left), $2 \times 2$ (lower–right) are represented.

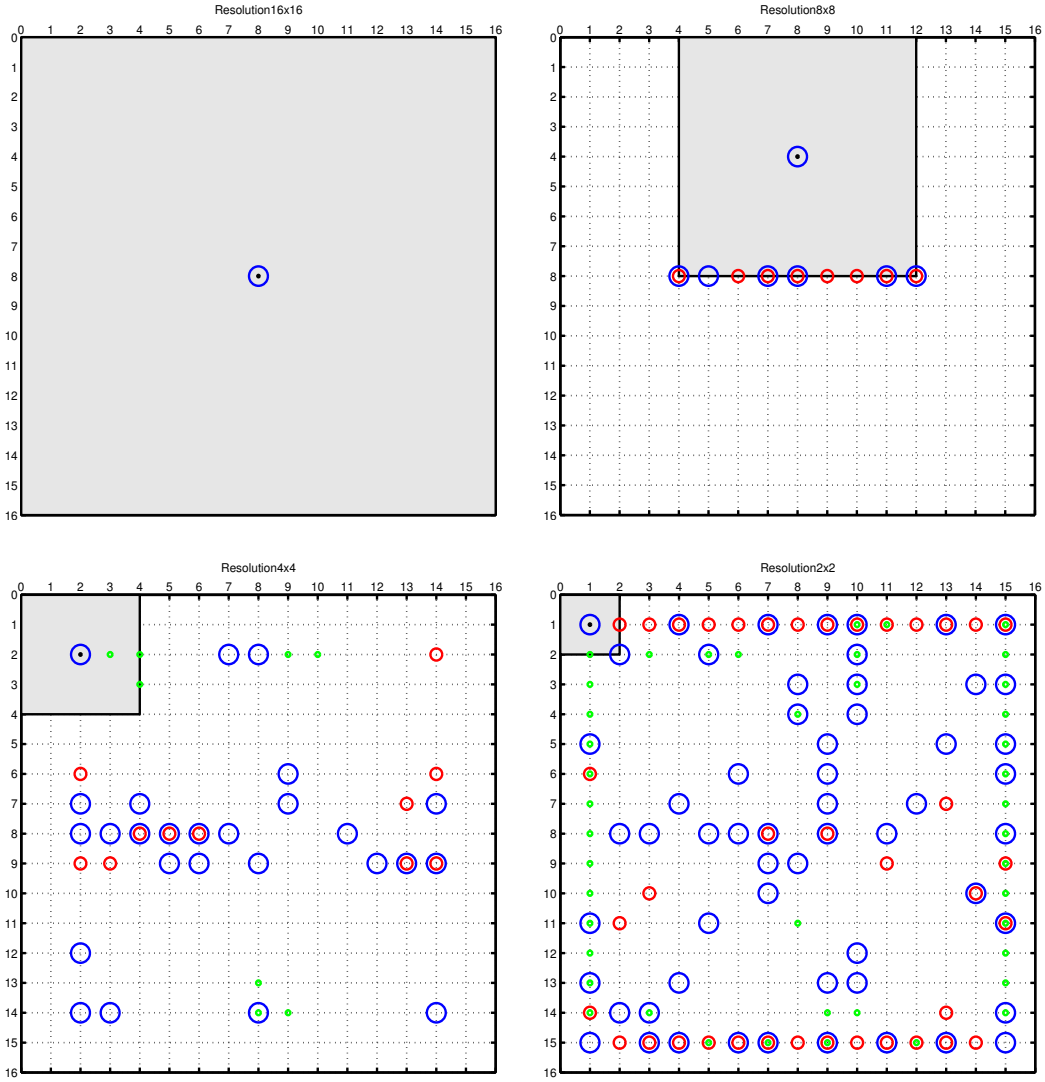



**Figure 4.29:** Ranklet coefficients after SVM–RFE has selected the 200 most relevant ones. Small green circles represent vertical ranklet coefficients, medium red circles represent horizontal ranklet coefficients, large blue circles represent diagonal ranklet coefficients. The gray dashed square represents the dimensions of the Haar wavelet supports. Resolution $16 \times 16$ (upper–left), $8 \times 8$ (upper–right), $4 \times 4$ (lower–left), $2 \times 2$ (lower–right) are represented.

Chapter 5

# CAD System Implementation

*In this Chapter, a practical application into a real–time working Computer–Aided Detection (CAD) system of some of the previously discussed image representations will be described. In particular, the wavelet–based and ranklet–based image representations will be used. The motivation for choosing those image representations is twofold. First, they prove to obtain excellent classification performances, in particular when compared to the others. Second, their implementation is almost straightforward and their computational times are definitely acceptable. Notice, specifically, that as regards the wavelet–based image representation, the overcomplete version will be considered. The reason is that—as previously evaluated—its richer spatial resolution allows for sensibly better classification performances. To this aim, Section 5.1 will introduce some of the main motivations for developing a featureless mass detection algorithm, together with some basic informations about the system. In Section 5.2, the mass detection scheme will be described in detail, with a specific attention to the practical implementation of the wavelet–based and ranklet–based image representations. Section 5.3 will give some details about the digital image database used in order to set up and evaluate the CAD system. Finally, in Section 5.4 some informations about the way in which performances are evaluated will be outlined, whereas in Section 5.5 the achieved results will be presented and discussed.*

## 5.1 System Motivation And Overview

As anticipated in Section 1.3, tumoral masses are thickenings of the breast tissue which appear on mammographic images as lesions with size ranging from 3 mm to 20–30 mm. Those lesions vary considerably in optical density, shape, position, size and characteristics of the boundary. In addition, their visual manifestation does not depend only upon the physical properties of the lesion itself, but it is also affected by the image acquisition technique and by the projection considered. It turns out that identifying morphological, directional or structural quantities that characterize them is very difficult.

The aspects outlined above make mass detection even more demanding for automatic CAD systems. In fact, automatic detection methods often rely on a feature extraction step in which masses are isolated by using a set of characteristics which describe them. Due to the great variety of masses, however, it proves to be extremely difficult to get a common set of features effective for every kind of masses. For that reason, many of the algorithms for mass detection so far developed have concentrated on the detection of a specific type of mass or—at least—on masses characterized by a particular size.

In order to deal with possibly every kind of masses, a detection system which does not rely on any feature extraction step is presented in this work. Considering the complexity of the class of objects to detect, considering that said objects frequently present characteristics similar to the environment which surround them and, finally, considering the objective difficulty of characterizing this class of objects with few measurable quantities, in the approach proposed herein no modeling is used. On the contrary, the algorithm automatically learns to detect masses by the examples presented to it, thus—as already discussed in Section 4.1—without any a priori knowledge provided by the trainer. Everything the system needs is a set of positive and negative examples, namely crops of tumoral masses and normal breast tissue. In particular, the detection scheme codifies the image with both a multi–resolution overcomplete Haar wavelet transform and a multi–resolution ranklet transform, as discussed in Sections 4.3.2 and 4.5.1. The amount of informations produced by each image representation is then separately classified by means of an SVM trained accordingly. Finally, a region is marked as a suspect mass according to a *combining strategy* applied to the results obtained by the two image representations. Notice, that the possibility of eliminating the feature extraction step is mainly due to the ability of SVM to handle multi–dimensional spaces and to maintain—at the same time—a good generalization capacity.

Several works, in the past, have used SVM in mammographic applications. As already discussed in Section 2.2, other than for its ability in handling multi–dimensional spaces without loosing in generalization capacity, this is mainly due to its advantages over other classifiers, namely an easier setting procedure and usually better performances on novel data. For example, in the past it has been used for reducing false positive signals, in the detection of mammographic micro–calcifications (Bazzani *et al.*, 2001) and in the diagnosis of ultra–sonography breast images (Chang *et al.*, 2003). Notice, in particular, that in both those cases SVM classifies signals by means of extracted image features. On the other hand, a featureless approach based on SVM for the detection of lesions in mammograms has been investigated for the first time by our group in (Campanini *et al.*, 2002, 2004c,a). In another study—see (El-Naqa *et al.*, 2002)—a similar approach has been used, but the class of object to detect—namely mammographic micro–calcifications—is much less heterogeneous in terms of size, shape and contrast.

## 5.2 Mass Detection Algorithm

The proposed mass detection algorithm is aimed at virtually detecting lesions whatever position they occupy and at whatever scales—or resolutions—they occur in the mammographic image. Roughly speaking, this is realized by scanning and classifying all the possible locations of the image—namely crops—with the passage of a window. By combining the scanning pass with an iterated resizing of the window, multi–scale detection is thus achieved. In such a context, each crop classified by SVM as belonging to the positive class of masses identifies an area judged as suspect by the CAD system.

Fig. 5.1 shows a detailed chart of the mass detection scheme presented herein. The first step consists of an external and internal breast *segmentation*, namely a pre–selection of the suspect regions within the breast. This is mainly achieved by means of a mammographic image resizing, a high–pass filtering, an adaptive local gray–level thresholding and, finally, the application of morphological operators. The aim of such a technique is basically to exclude the background area from further processing and to find out suspect regions within the breast. Due to such a segmentation, a significant reduction of both the number of false positives per image and of the computational times is achieved. For more details on that see (Campanini *et al.*, 2004b).

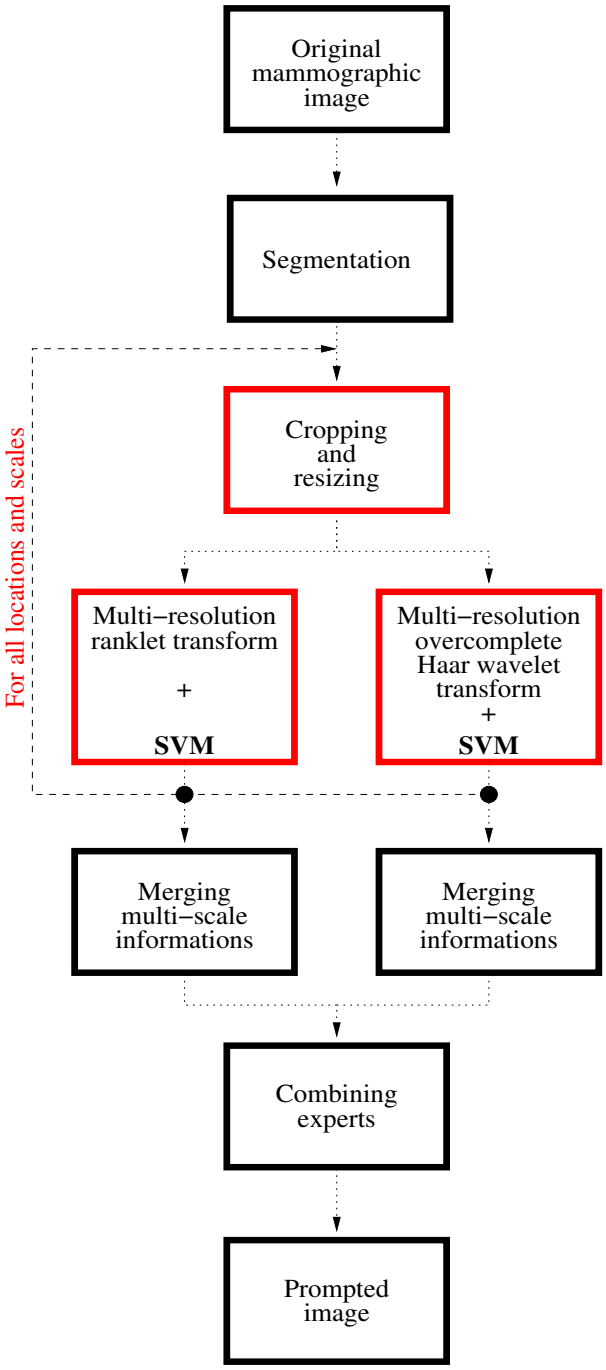



**Figure 5.1:** Mass detection algorithm.

Second, one of the main problems to address in mass detection is that lesions occur at different scales in the mammogram, typically in a range of dimensions from 3 mm to 20–30 mm. There thus emerges the necessity of scanning the mammographic image at different scales. On the other hand, the system needs a fixed size crop, since SVM deals with dimensionally homogeneous vectors. The solution implemented is consequently that of *cropping* the entire image by means of scanning windows with different dimensions and then *resizing* all the obtained crops to a pre–fixed pixel size $64 \times 64$.

For example, consider an input image with $4000 \times 3000$ pixel size—each pixel being 50 $\mu$m—and three scale targets of 32 mm (640 pixels), 16 mm (320 pixels), and 10 mm (213 pixels). The desired dimension of the crop is obtained resizing by means of bi–linear interpolation the windows of $640 \times 640$, $320 \times 320$ and $213 \times 213$ pixel size to 10%, 20% and 30% respectively. The analysis of the entire image is thus obtained by shifting the window with a scanning step fixed to approximately 10% of the linear dimensions of the window. In this way, there is a certain degree of superposition between contiguous squares. Without superposition, in fact, many lesions could fail to be detected because they are not centered on the scanning crop. This is consistent with the fact that during the training phase the positive examples are shown as crops centered on a mass. Notice, finally, that the number of analyzed scales is strictly related to the range size of the masses to detect.

Third, a multi–resolution analysis of each resized crop is performed by using two different approaches, namely the *multi–resolution overcomplete Haar wavelet transform* and the *multi–resolution ranklet transform* discussed in Section 4.3.2 and Section 4.5.1. The motivation for choosing those image representations is twofold. First—as discussed in detail in Section 4.3.3 and Section 4.5.2—they proved to obtain excellent classification performances, in particular when compared to the others. The reason is probably that their multi–resolution and orientation selective properties—together with redundancy for the multi–resolution overcomplete Haar wavelet transform and non–parametricity for the multi–resolution ranklet transform—make them particularly suitable for this kind of pattern classification problems. Second, their implementation is almost straightforward and their computational times are definitely acceptable. In such a way, the number of coefficients obtained for each image representation is quite high, approximately 3000 for the former, whereas 1428 for the latter. In other words, each resized crop is represented by a vector of approximately 3000 wavelet–based classification features and a vector of 1428 ranklet–based classification features.

Each one of the two feature vectors is thus used as input for one of two dedicated SVMs. Before the CAD system is applied in real–time modality, in fact, one SVM is trained by means of wavelet–based features, whereas the other by means of ranklet–based features. In such a way—once trained—each SVM is capable of classifying the correspondent input vector of wavelet–based or ranklet–based features. In particular, for each crop, SVM gives the distance from the separating Maximal Margin Hyperplane discussed in Eq. 2.45. This distance is used as an index of confidence on the correctness of the classification. In the past, in fact, some work has been done in order to extract a posterior probability from SVM outputs, see for example (Platt, 1999). With this in mind, a feature vector classified as positive with a large distance from the hyperplane will have a higher likelihood of being a true positive as compared to a vector very close to the hyperplane and hence close to the boundary area between the edges of the two classes. Following this approach, the scanning of all possible locations—at all analyzed scales—provides a list of suspect candidates, each candidate consisting of a crop with a distance from the hyperplane greater than a prefixed threshold.

The fourth step in the proposed detection scheme consists of *merging multi–scale informations*. The output of each SVM is in fact a set of candidates detected at either one of the scales. However, the same suspect region can be detected at several scales. In this case, the centers of the various candidates—representing that region at different scales—may not be the same, since the scanning step at one particular scale is different from the others. The candidates are then fused within a specified neighborhood into a single candidate. Therefore, the output of both the two detection methods—namely the wavelet–based and the ranklet–based—is a list of suspect regions, each one detected at least at one scale, see Fig. 5.2. Notice that, in literature, the output of a detection method—namely the specific image representation adopted, the classifier used, the classifier's settings and so forth—is usually referred to as the result achieved by a particular *expert*.

The final step consists of *combining* together the results obtained by the two experts in order to produce the final detection. The basic idea is that an ensemble of experts may improve the overall performance of each individual expert, provided that the individual experts are independent, namely they commit mistakes on different objects, see (Kuncheva *et al.*, 2000). In this specific case, the detection performance of the two experts is almost identical. However—due to the different image representations, kernels and training conditions used—they will often make different errors. Hence, one very efficient way to reduce false positives is to combine their outputs by performing a logical AND.

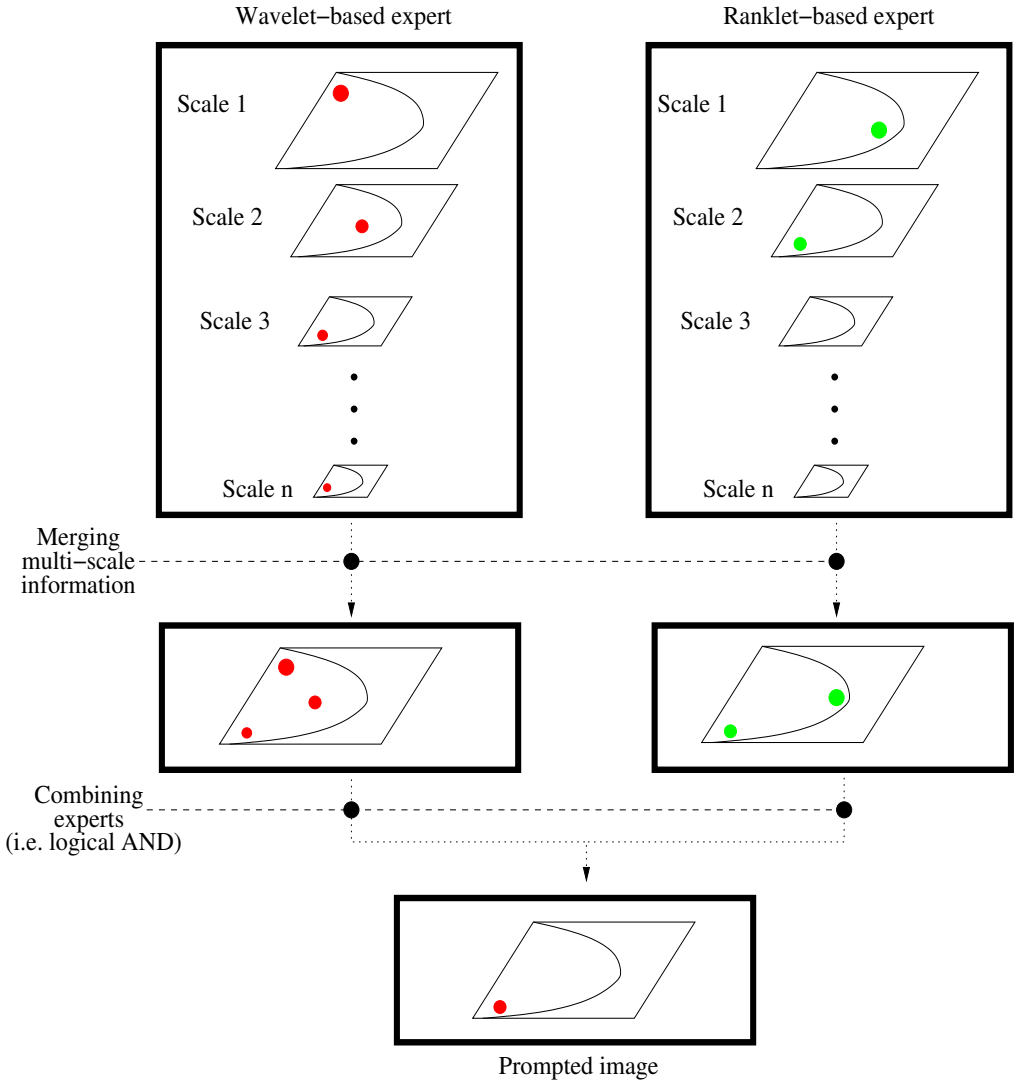



**Figure 5.2:** Merging multi–scale informations and combining the results of the wavelet–based and ranklet–based experts. Merging multi–scale informations consists of fusing into a single candidate all the candidates at all scales within a specified neighborhood. Combining the results consists of performing a logical AND of the results obtained after merging is completed.

## 5.3 Image Data Set

A former version of the proposed mass detection scheme has already been tested on digitized images of the DDSM database described in Section 4.1.2, see (Campanini *et al.*, 2004c). On that former version, however, the system was based on the combination of three experts all using the multi–resolution overcomplete Haar wavelet transform and differing mainly in the SVM's kernels used.

Further tests have been then performed on the FFDM database, namely digital images collected at two different sites, Maggiore Hospital in Bologna—Italy—and Triemli Hospital in Zurich—Switzerland—as described in (Campanini *et al.*, 2004a). Both those hospitals have a Giotto Image MD Full Field Digital Mammography (FFDM) system manufactured by Internazionale Medico Scientifica (IMS), Italy. Also in this case, however, the system was provided with three experts based exclusively on multi–resolution overcomplete Haar wavelet transform. Furthermore, its settings were optimized for the combined detection of both masses and micro–calcifications in mammograms.

In order to evaluate the mass detection performances of the CAD system by using both the wavelet–based and the ranklet–based image representations, the FFDM data set is used. This data set consists of about 750 images. They have gray–level resolution of 13 bits and linear pixel dimensions correspondent to 85 $\mu$m. They have been collected both in the course of the clinical evaluation of the FFDM system and subsequently during the regular clinical examinations. Each case is relative to one patient and is composed of four projections, namely two *cranio–caudal* and two *medio–lateral* views. In particular, digital mammograms are always available in four projections per patient. The database is comprised of 672 normal images without lesions and 88 images with at least one lesion, such as tumor opacities or clustered micro–calcifications. The locations of lesions have been marked by expert radiologists and collected together with the images.

## 5.4 Performance Evaluation

In order to set up and evaluate the mass detection algorithm, the data set is divided into two sub–sets, namely a training set—even though quite improperly, as it will be discussed in the following—and a test set. The former consists of 46 cancer images and 52 normal ones, whereas the latter of 42 cancer images and 620 without lesions. In the following the training and test procedures will be described.

### 5.4.1 Training procedure

The training of the CAD system is obtained by presenting a set of crops with pixel size 64×64 containing masses and a set of crops with pixel size 64×64 corresponding to normal tissue. Crops corresponding to masses represent positive examples for the classifier, whereas crops corresponding to normal tissue represent negative examples.

In such a context, each positive example is a portion of a mammographic image which contains completely a mass. In particular, the size of the positive crops is chosen so that the ratio between the crop area and the area of the mass core is nearly 1.3. In this way, all the positive examples are characterized by having about 30% of background and 70% of the area occupied by the mass. As a further consequence, the real size of masses is smaller than the size of the searching scale. For example, a scale with a 40 mm crop is appropriate for searching masses of 35 mm. Notice that, in this way, the classifier is specifically trained to recognize—as positives—feature vectors corresponding to squares centered on lesions.

As regards negative examples, they have no superposition with positive examples, since negative crops are extracted from normal cases, whereas positive crops from malignant ones. Furthermore, whilst the positive examples are quite well defined, there are no typical negative examples. To overcome the problem of defining this extremely large negative class, a *bootstrap* technique is used, see (Efron & Tibshirani, 1993). Namely, after the initial training, the system is re–trained by using a new set containing some mis–classified false positive examples. Those examples, in particular, are obtained from the detection of images which are not present in the initial training set. This procedure is thus iterated until an acceptable performance is achieved. In this way, the system is forced to learn by its own errors.

Due to the small number of FFDM images available for training the CAD system, the training set previously accomplished with the digitized images coming from the DDSM database is used in place of the former. This training set is comprised of few hundreds crops with pixel size $64 \times 64$ containing masses and few thousands crops with pixel size $64 \times 64$ corresponding to normal tissue. In particular, the difference between the number of images available to train the CAD system in the DDSM database—800 malignant, 600 normal—and in the FFDM database—46 malignant, 52 normal—is evident by noticing that they differ by one order of magnitude. Some crops used in the training procedure has been already shown in Fig. 4.1.

### 5.4.2 Test procedure

In order to evaluate the performance of the CAD system on the 42 cancer and 620 normal images of the FFDM test set, a sigmoidal Look–Up–Table (LUT) needs to be first applied for transforming the histogram of those FFDM images. The main objective is to find the best LUT mapping the FFDM images histogram into the histogram relative to the DDSM images used for the training step. In particular, in order to close the optimal LUT, the system is trained with 44 positive crops of lesions and 4000 negative crops of normal tissue taken from the FFDM images of the—so–called—training set. This approach is fundamental, since it allows to exploit the very large number of images from the DDSM database to train the system, whereas the detection performances can be evaluated on the still small FFDM dataset available.

## 5.5 Results

The proposed CAD system searches for masses with a size smaller than 35 mm. Therefore the multi–scale detection is performed by using as scales 8, 10, 13, 17, 22, 27, 33 and 40 mm. A region is defined as a true positive if its center falls within the ground–truth annotations, otherwise it is considered as a false positive. The number of false positives is computed using normal cases only.

The detection performances of the system are evaluated by means of FROC curves. As already discussed in Section 2.1.4, an FROC curve is a plot of the detection rate versus the average number of false positive marks per image. It provides a summary of the trade–off between the sensitivity and the specificity of the system.

In particular, the performance results are presented on a *per–mammogram* and on a *per–case* basis. In the former, the cranio–caudal and medio–lateral oblique views are considered independently. In the latter, a mass is considered discovered if it is detected in either one of the views. Notice, in particular, that the per–case evaluation takes into consideration that, in clinical practice, once the CAD alerts the radiologist to a cancer on one view, it is unlikely that the radiologist will miss the cancer. In this way, the scoring method considers all the malignant masses on a mammogram—or in a case—as a single true positive finding. The rationale is that a radiologist may not need to be alerted to all malignant lesions in a mammogram or case before taking action.

| $n_{Wav}$ | 2 | 2 | 3 | 3 | 5 | 10 |
|---|---|---|---|---|---|---|
| $n_{Rank}$ | 1 | 2 | 3 | 10 | 10 | 10 |
| *Mean number of false positives per–image* | 0.35 | 0.50 | 0.73 | 1.08 | 1.47 | 2.10 |
| *True positive fraction per–mammogram* | 0.52 | 0.59 | 0.72 | 0.78 | 0.80 | 0.84 |
| *True positive fraction per–case* | 0.73 | 0.77 | 0.86 | 0.91 | 0.95 | 0.95 |

**Table 5.1:** Performance of the proposed mass detection algorithm evaluated on 42 cancer and 620 normal images taken from the FFDM database. Results are given on a per–mammogram and on a per–case basis.

The CAD system performances are evaluated by putting a threshold on the maximum number of signals which the wavelet–based and the ranklet–based experts prompt before the logical AND. Basically, the suspect candidates are ranked according to their distance from the SVM's hyperplane. Only the best $n_{Wav}$ signals are then kept for the wavelet–based expert, whereas only the best $n_{Rank}$ are kept for the ranklet–based one. In particular, the different points of the FROC curve are obtained by varying the thresholds $n_{Wav}$ and $n_{Rank}$ for both the experts.

Tab. 5.1 shows the performances achieved by the proposed CAD system when evaluated on 42 cancer and 620 normal images taken from the test set of the FFDM database. Furthermore, a plot of the FROC curve correspondent to those performances is shown in Fig. 5.3. Notice, in particular, that results are shown on both a per–mammogram and a per–case basis. The results achieved are definitely promising and clearly indicate the suitability of the presented CAD system in detecting breast masses. Furthermore, they improve the performances obtained by our group in (Campanini *et al.*, 2004c,a) with a former version of the system provided with exclusively wavelet–based experts. It is evident that a direct comparison of the results is impossible, since in one case the image databases used are different, whereas in the other the performances are evaluated on the detection of both masses and micro–calcifications. However, the clear improvement in the results obtained by the CAD system proposed herein seems to confirm the effectiveness of combining the discussed wavelet–based and ranklet–based experts.

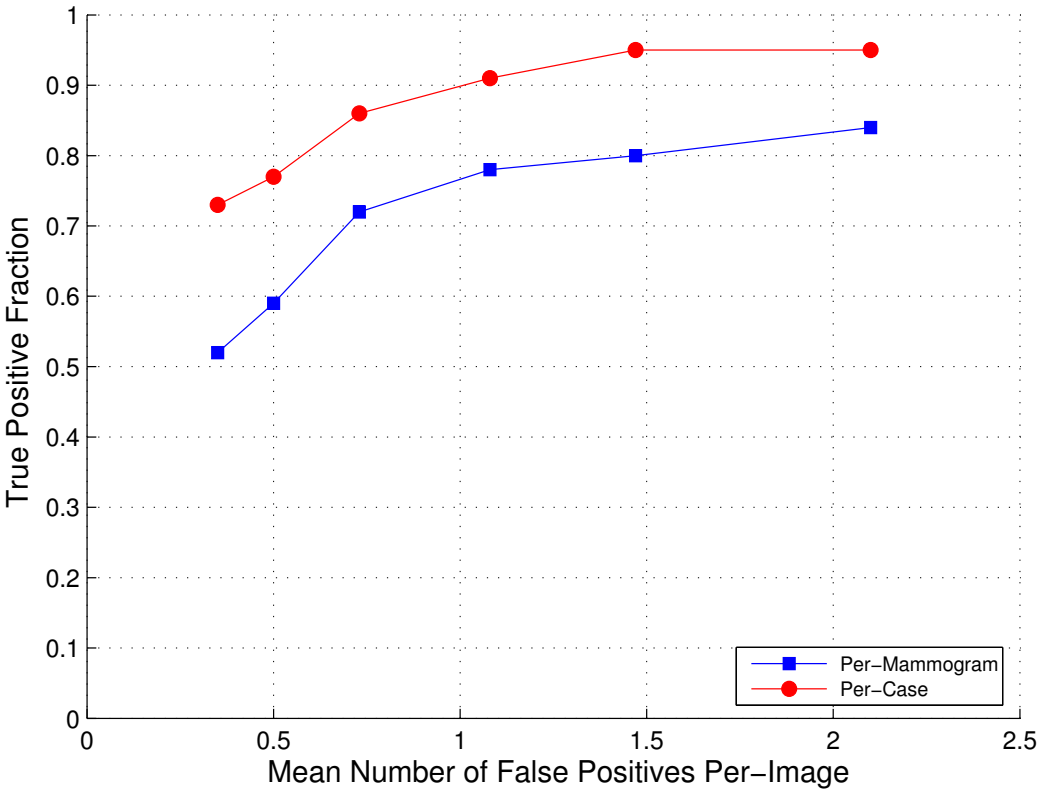



**Figure 5.3:** FROC curve correspondent to the proposed mass detection algorithm evaluated on 42 cancer and 620 normal images taken from the FFDM database. Results are given on a per–mammogram and on a per–case basis.

The reason for those successful results is probably twofold. First, the multi–resolution overcomplete Haar wavelet transform and the multi–resolution ranklet transform achieve excellent classification performances, as discussed in detail in Section 4.3.3 and Section 4.5.2. In other words, they are particularly suited to achieve high sensitivity values—namely high true positive values—when classifying tumoral masses. Second—by looking carefully at their marks on the mammograms under exam—they prove to act as two almost completely independent experts. In fact, they typically commit mistakes on different regions of the mammograms. Given those premises, performing a logical AND of their outputs is an approach which walks in the direction of maintaining a high specificity, while reducing the number of false positives per–image, see for example Fig. 5.4.

Future work has still to be done, for instance in order to enlarge the FFDM image database. In this way it will be possible to train new experts directly on digital images, allowing an improvement of the CAD results and a more precise determination of its performance. Further improvements could then be achieved by implementing other independent experts, namely the pixel–based image representation discussed in Section 4.2. This image representation, in fact, proved to achieve classification performances very close to that obtained by means of the wavelet–based and ranklet–based image representations.

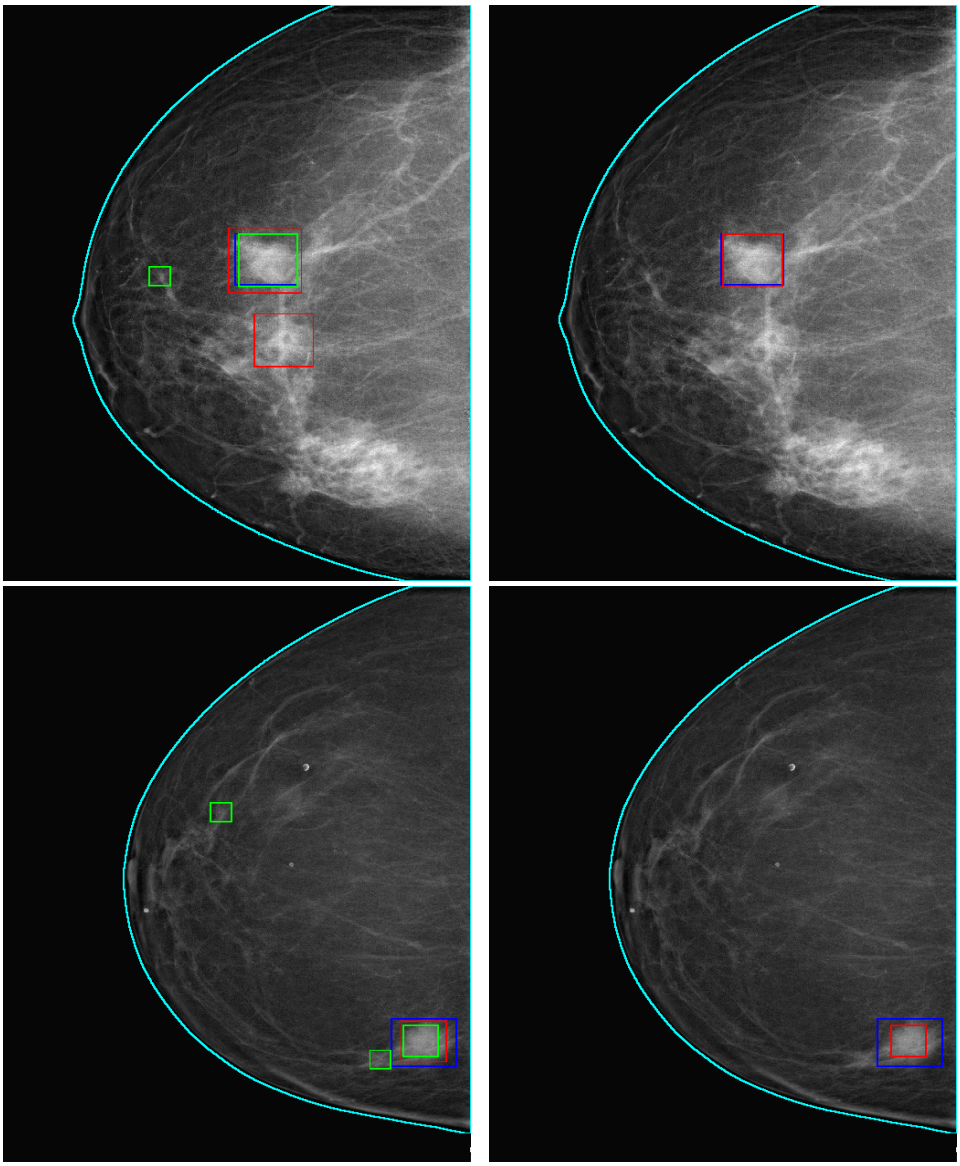


**Figure 5.4:** False positive reduction by combining both the wavelet–based and ranklet–based experts. Left column: mammographic images before the wavelet–based (red marks) and ranklet–based (green marks) outputs are combined by logical AND. Right column: the logical AND between the two experts is performed so that the true diagnosed mass (blue marks) survives as marked, whereas all false positives are rejected.

# Conclusions

In this work, a two–class classification problem is faced. In particular, the two classes to separate are tumoral masses—namely thickenings of the breast tissue with size ranging from 3 mm to 30 mm—and normal breast tissue. In order to do that, each X–ray image under study is scanned at all the possible locations with the passage of a window. A corresponding crop of the mammographic image is then extracted and classified by means of a Support Vector Machine (SVM) classifier as belonging to the class of tumoral masses or to the class of normal breast tissue. Differently from the most part of the mass detection algorithms developed up to now, the approach proposed herein does not rely on any feature extraction step aimed at individuating some measurable quantities characterizing masses. On the contrary, it is rather a *featureless* approach in which crops are passed to the classifier in their raw form, namely as vectors of gray–level values. The reason for this choice is that—due to the great variety of masses—it is extremely difficult to get a set of common features effective for all kind of masses. Thus, in order to deal with all their different kinds, a possible choice is the featureless approach, in which no a priori information is extracted from the crops.

The first experimental part of this work is aimed at evaluating the classification performances of different image representations—such as image representations based on pixels, wavelets, steerable pyramids and ranklets—by means of ROC curve analysis. This literally means that the features used to classify each crop are respectively the gray–level values of the crop, the coefficients obtained after applying the wavelet transform to the crop, the coefficients obtained after applying the steerable pyramid to the crop and, finally, the coefficients obtained after applying the ranklet transform to the crop. In this sense, each specific image representation selected for the crop embodies itself the features to classify.

Experiments show some very interesting results. The *pixel–based* image representation achieve very good classification performances, in particular when the crops are processed by means of histogram equalization and bi–linear resizing. Good performances are achieved as well by the *wavelet–based* image representations, in particular by its overcomplete version. The reason is probably that a richer spatial resolution allows for better classification performances. Very preliminary results show also that the image representation based on steerable pyramid—namely *steer–based* image representation—performs quite well. Nevertheless, the last results must be considered as a sort of anticipation of a more complete study which is—at the time—still under development.

The best classification performances are achieved by the *ranklet–based* image representation. In some sense, it could be considered as the *optimal image representation* for the image classification problem under analysis. In particular, due to its interesting results, further investigations are carried out by applying SVM Recursive Feature Elimination (SVM–RFE), namely by recursively eliminating some of the ranklet coefficients and—contemporary—monitoring the classification performances. Tests show that it is possible to sensibly reduce the number of ranklet coefficients—namely from 1428 down to 200—without affecting the classification performances. Furthermore, they show that at fine resolutions the most discriminant ranklet coefficients are those near the borders of the image, thus those codifying the contour information of the image. On the other hand, at coarse resolutions, the most important ranklet coefficients are those near the center of the image, thus those codifying the symmetry information of the image, rather that its contour information. This results seems reasonable, since the main difference between the two classes at fine resolutions is that masses have sharp edges near the borders of the image, whereas normal breast tissue has not. At the same time, at coarse resolutions the main difference is that masses appear approximately as symmetric circular structures centered on the image, whereas normal tissue has a less definite structure.

The second experimental part of this work deals with the application of two of the best image representations found into a real–time working Computer–Aided Detection (CAD) system. In particular—due to their excellent classification performances and almost straightforward implementation—the two image representations implemented are the ranklet–based and the wavelet–based. The approach proposed is that of first considering the two image representations as two separated mass detectors which commit different errors on the mammographic images under study. Then, to combine their responses by performing a logical AND.

This strategy is particularly effective, since those image representations prove to be well suited to achieve high sensitivity values—namely high true positive values—when classifying tumoral masses. At the same time, they act as two—almost completely—independent experts, since they commit mistakes on different regions of the mammograms. It follows that the logical AND of their responses maintains a high specificity, while reducing the number of false positives per–image. In order to give some quantitative result, the mass detection scheme proposed marks per–case 77% of cancers with a false–positive rate of 0.5 marks per–image. Due to those good classification results, the system proposed herein is currently deployed at three hospitals worldwide in its prototype version.

# Bibliography


Aldroubi, A. & Unser, M. (1996). *Wavelets in Medicine and biology*. CRC Press, Boca Raton, FL. 69

American Cancer Society (2005). Cancer facts and figures 2005. 8, 9

Angelini, E., Campanini, R., Iampieri, E., Lanconelli, N., Masotti, M. & Roffilli, M. (2004). Testing the performances of different image representations for mass classification in digital mammograms. *Image and Vision Computing*, Submitted, Pre–print available at: http://www.bo.infn.it/~masotti/publications.html. 3, 88, 101

Bazzani, A., Bevilacqua, A., Bollini, D., Brancaccio, R., Campanini, R., Lanconelli, N., Riccardi, A. & Romani, D. (2001). An SVM classifier to separate false signals from microcalcifications in digital mammograms. *Physics in Medicine and Biology*, **46**, 1651–1663. 149

Bishop, C.M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press. 39

Boser, B., Guyon, I. & Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, 144–152. 42

Burges, C.J.C. (1988). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, **2**, 121–167. 42

Campanini, R., Bazzani, A., Bevilacqua, A., Bollini, D., Dongiovanni, D.N., Iampieri, E., Lanconelli, N., Riccardi, A., Roffilli, M. & Tazzoli, R. (2002).

A novel approach to mass detection in digital mammography based on support vector machines (SVM). In *International Workshop on Digital Mammography 2002 Proc.*, 399–401. 3, 149

Campanini, R., Angelini, E., Dongiovanni, D.N., Iampieri, E., Lanconelli, N., Mair-Noack, C., Masotti, M., Palermo, G., Roffilli, M., Saguatti, G. & Schiaratura, O. (2004a). Preliminary results of a featureless CAD system on ffdm images. In *International Workshop on Digital Mammography 2004 Proc.*, Available at: http://www.bo.infn.it/~masotti/publications.html. 3, 149, 154, 157

Campanini, R., Angelini, E., Iampieri, E., Lanconelli, N., Masotti, M., Roffilli, M., Schiaratura, O. & Zanoni, M. (2004b). A fast algorithm for intra-breast segmentation of digital mammograms for CAD systems. In *International Workshop on Digital Mammography 2004 Proc.*, Available at: http://www.bo.infn.it/~masotti/publications.html. 149

Campanini, R., Dongiovanni, D., Iampieri, E., Lanconelli, N., Masotti, M., Palermo, G., Riccardi, A. & Roffilli, M. (2004c). A novel featureless approach to mass detection in digital mammograms based on support vector machines. *Physics in Medicine and Biology*, **49**, 961–975, Available at: http://www.bo.infn.it/~masotti/publications.html. 3, 149, 154, 157

Castleman, K.R. (1996). *Digital Image Processing*. Prentice Hall, Englewood Cliffs, NJ, USA. 64

Chang, R.F., Wu, W.J., Moon, W.K., Chou, Y.H. & Chen, D.R. (2003). Support vector machines for diagnosis of breast tumors on us images. *Academic Radiology*, **10**, 189–97. 149

Cortes, C. & Vapnik, V. (1995). Support vector networks. *Machine Learning*, **20**, 273–297. 42, 49

Croisier, A., Esteban, D. & Galand, C. (1976). Perfect channel splitting by use of interpolation/decimation/tree decomposition techniques. In *International Conference Information Science and Systems*, 443–446. 70

Daubechies, I. (1988). Orthonormal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, **41**, 909–996. 67, 70

Daubechies, I. (1992). *Ten lectures on wavelets*. SIAM, Philadelphia, PA. 69

Day, W.H.E. & Edelsbrunner, H. (1984). Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of Classification*, **1**, 7–24. 26

Del Bue, A., Smeraldi, F. & Agapito, L. (2004). Non-rigid structure from motion using non–parametric tracking and non–linear optimization. In *Proceedings of the IEEE Workshop on Articulated and Nonrigid Motion, Washington DC*. 88

Duda, R.O., Hart, P.E. & Stork, D.G. (2000). *Pattern Classification*. John Wiley & Sons, Inc., New York, 2nd edn. 23, 27

Efron, B. & Tibshirani, R.J. (1993). *An Introduction to the Bootstrap*. Chapman and Hall. 155

Egan, J.P. (1975). Signal detection theory and ROC analysis. *Series in Cognitition and Perception*. 35

El-Naqa, I., Yang, Y., Wernick, M.N., Galatsanos, N.P. & Nishikawa, R.M. (2002). Support vector machine approach for detection of microcalcifications. *IEEE Transactions on Medical Imaging*, **21**, 1552–1563. 149

Fawcett, T. (2004). ROC graphs: Notes and practical considerations for researchers. Tech. rep., HP Laboratories, Palo Alto, CA, USA. 35

Ferlay, J., Bray, F., Pisani, P. & Parkin, D.M. (2004). Globocan 2002: Cancer incidence, mortality and prevalence worldwide. 9

Franceschi, E., Odone, F., Smeraldi, F. & Verri, A. (2004). Finding objects with hypothesis testing. In *Proceedings of the Workshop on Learning for Adaptable Visual Systems, in conjunction with ICPR'04, Cambridge, UK*. 88

Freeman, W. (1992). Steerable filters and the local analysis of image structure. 78

Freeman, W.T. & Adelson, E.H. (1991). The design and use of steerable filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **13**, 891–906. xv, 78, 79, 80, 82, 83, 86

Furey, T.S., Christianini, N., Duffy, N., Bednarski, D.W., Schummer, M. & Hauessler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, **16**, 906–914. 54

Gautrais, J. & Thorpe, S.J. (1998). Rate coding vs temporal order coding: A theoretical approach. *Biosystems*, **48**, 5765. 96

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D. & Lander, E.S. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–537. 54

Gonzalez, R.C. & Woods, R.E. (1992). *Digital Image Processing*. Addison-Wesley, Reading, MA, USA, 3rd edn. 60, 64

Grossmann, A. & Morlet, J. (1984). Decomposition of hardy functions into square integrable wavelets of constant shape. *SIMAT*, **15**, 723–736. 67

Guyon, I., Boser, B. & Vapnik, V. (1993). Automatic capacity tuning of very large vc–dimension classifiers. *Advances in Neural Information Processing Systems*, **5**, 147–155. 42

Guyon, I., Weston, J., Barnhill, S. & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, **46**, 389–422. 53

Haar, A. (1910). Zur Theorie der orthogonalen Funktionensysteme. *Mathematische Annalen*, **69**, 331–371. 67, 74

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice Hall. 39

Heath, M., Bowyer, K.W., Copans, D., Moore, R. & Kegelmeyer, P. (2000). The digital database for screening mammography. *Digital Mammography: IWDW2000 5th International Workshop on Digital Mammography*, 212–218. 101

Hubbard, B.B. (1996). *The World According to Wavelets: the Story of a Mathematical Technique in the Making*. A. K. Peters, Ltd. 69

Jacob, M. & Unser, M. (2004). Design of steerable filters for feature detection using Canny-like criteria. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**, 1007–1019. 78

KARASARIDIS, A. & SIMONCELLI, E.P. (1996). A filter design technique for steerable pyramid image transforms. In *ICASSP*, vol. IV, 2389–2392, IEEE Sig Proc Society, Atlanta, GA. 78, 85

KAUFMAN, L. & ROUSSEEUW, P. (1990). *Finding Groups in Data: An introduction To Cluster Analysis*. J. Wiley, New York. 26

KEARNS, M., MANSOUR, Y. & RON, D. (1997). An experimental and theoretical comparison of model selection methods. *Machine Learning*, **27**, 7–50. 53

KOHAVI, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence*, 1137–1145. 30

KOHAVI, R. & JOHN, G. (1997). Wrappers for feature subset selection. *Artificial Intelligence Journal*, **97**, 273–324. 53, 54

KUNCHEVA, L.I., WHITAKER, C.J., SHIPP, C.A. & DUIN, R.P.W. (2000). Is independence good for combining classifiers? In *Proceedings of the 15th International Conference on Pattern Recognition*, 168–171. 152

LASDON, L.S. (1970). *Optimization Theory for Large Systems*. MacMillan. 50

LECUN, Y. (1986). Learning processes in an asymmetric threshold network. In *Disordered Systems and Biological Organizations*, 233–240, Springer-Verlag, Les Houches, France. 39

LECUN, Y., DENKER, J.S. & SOLLA, S.A. (1990). Optimum brain damage. In *Advances in Neural Information Processing Systems 2*, 598–605. 55

LEHMANN, E.L. (1995). *Nonparametrics: Statistical Methods Based on Ranks*. Holden–Day. 89, 90

LLOYD, S.P. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, **2**, 129–137. 26

MALLAT, S. (1989). A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**, 674–693. 67, 76

MALLAT, S. (1998). *A Wavelet Tour of Signal Processing*. Academic Press. 69

Mao, J. & Jain, A.K. (1996). A self–organizing network for hyperellipsoidal clustering (HEC). *IEEE Transactions on Neural Networks*, **7**, 16–29. 26

Marsteller, L.P. & Shaw de Paredes, E. (1989). Well defined masses in the breast. *Radiographics*, **9**, 13–37. 13

Masotti, M. (2004). A ranklet–based image representation for mass classification in digital mammograms. *Pattern Recognition*, Submitted, Pre–print available at: http://www.bo.infn.it/~masotti/publications.html. 3, 88, 101

Masotti, M. (2005). Exploring ranklets performances in mammographic mass classification using recursive feature elimination. In *The IEEE International Conference On Image Processing, Genova, Italy*, Submitted, Pre–print available at: http://www.bo.infn.it/~masotti/publications.html. 3, 88, 101

Mercer, J. (1909). Functions of positive and negative type and their connection with the theory of integral equations. *Transactions of the London Philosophical Society*, **209**, 415–446. 52

Meyer, Y. (1987). Les ondelettes. In *Contributions to nonlinear partial differential equations, Vol. II (Paris, 1985)*, vol. 155 of *Pitman Res. Notes Math. Ser.*, 158–171, Longman Sci. Tech., Harlow. 67

Meyer, Y., ed. (1992). *Wavelets and applications*, vol. 20 of *RMA: Research Notes in Applied Mathematics*, Masson, Paris. 69

Müller, K.R., Mika, S., Rätsch, G. & Sch"olkopf, B. (2001). An introduction to kernel–based algorithms. *IEEE Transactions on Neural Networks*, **12**, 181–202. 43

Murphy, W.A. & DeSchryver-Kecskemeti, K. (1978). Isolated clustered microcalcifications in the breast: radiologic–pathologic correlation. *Radiology*, **127**, 335–341. 16

Nishikawa, R.M., Haldemann, R.C., Papaioannou, J., Giger, M.L., Lu, P., Schmidt, R.A., Wolverton, D.E., Bick, U. & Doi, K. (1995). Medical imaging 1995: Image processing. In M.H. Loew, ed., *Initial experience with a prototype clinical intelligent mammography workstation for computer-aided diagnosis*, vol. 2434, 65–71, SPIE. 17

Novikoff, A.B.J. (1962). On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, vol. XII, 615–622. 39

Oren, M., Papageorgiou, C., Sinha, P., Osuna, E. & Poggio, T. (1997). Pedestrian detection using wavelet templates. 3, 111

Papageorgiou, C. (1997). *Object and Pattern Detection in Video Sequences*. Master's thesis, MIT. 3, 111

Papageorgiou, C. & Poggio, T. (1999a). A pattern classification approach to dynamical object detection. In *International Conference on Computer Vision ICCV'99*, 1223–1228. 3, 111

Papageorgiou, C. & Poggio, T. (1999b). Trainable pedestrian detection. In *International Conference on Computer Vision ICIP'99*. 3, 111

Papageorgiou, C., Evgeniou, T. & Poggio, T. (1998a). A trainable pedestrian detection system. In *IEEE Conference on Intelligent Vehicles, 1998*, 241–246. 3, 111

Papageorgiou, C., Oren, M. & Poggio, T. (1998b). A general framework for object detection. In *International Conference on Computer Vision ICCV'98*. 3, 111

Pavlidis, P., Weston, J., Cai, J. & Grundy, W.N. (2001). Gene functional classification from heterogeneous data. In *RECOMB '01: Proceedings of the fifth annual international conference on Computational biology*, 249–255. 54

Perona, P. (1992). Steerable–scalable kernels for edge detection and junction analysis. *Image and Vision Computing*, **10**, 663–672. 85

Platt, J.C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In B.S. A. Smola P. Bartlett & D. Schuurmans, eds., *Advances in Large Margin Classifiers*, vol. 10, 61–74, MIT Press. 152

Portilla, J. & Simoncelli, E.P. (2000). A parametric texture model based on joint statistics of complex wavelet coefficients. *Int'l Journal of Computer Vision*, **40**, 49–71. 78

Portilla, J., Strela, V., Wainwright, M. & Simoncelli, E.P. (2003). Image denoising using a scale mixture of Gaussians in the wavelet domain. *IEEE Trans Image Processing*, **12**, 1338–1351. 78

Roehrig, J., Doi, T., Hasegawa, A., Hunt, B., Marshall, J., Romsdahl, H., Schneider, A., Sharbaugh, R. & Zhang, W. (1998). Digital mammography. In N. Karssemeijer, M. Thijssen, J. Hendriks & L. van Erning L. Dordrecht: Kluwer Academic Publishers, eds., *Clinical results with R2 ImageChecker system*, 395–400. 17

Rosenblatt, F. (1962). *Principles of Neurodynamics*. Spartan Books, Washington, DC. 36

Rumelhart, D.E., Hinton, G.E. & Williams, R.J. (1986). Learning internal representations by error propagation. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1: foundations*, 318–362. 39

Sajda, P., Spence, C. & Pearson, J. (2002). Learning contextual relationships in mammograms using a hierarchical pyramid neural network. *IEEE Transactions on Medical Imaging*, **21**. 4, 122

Schölkopf, B. (1997). *Support Vector Learning*. Ph.D. thesis, Universität Berlin. 22, 23

Schölkopf, B., Simard, P.Y., Smola, A.J. & Vapnik, V.N. (1998). Prior knowledge in support vector kernels. In M.I. Jordan, M.J. Kearns & S.A. Solla, eds., *Advances in Neural information processings systems*, vol. 10, 640–646, MIT Press, Cambridge, MA. 110

Shadlen, M.N. & Newsome, W.T. (1995). Is there a signal in the noise? *Curr. Opin. Neurobiol.*, **5**, 248250. 96

Shadlen, M.N. & Newsome, W.T. (1998). The variable discharge of cortical neurons: Implications for connectivity, computation and information coding. *J. Neurosci.*, **18**, 38703896. 96

Shaw de Paredes, E. (1993). Radiographic breast anatomy: Radiologic signs of breast cancer. *RSNA Categorical Course in Physics*, 35–46. 12

Sickles, E.A. (1982). Mammographic detectability of breast microcalcifications. *American Journal of Roentgenology*, **139**, 913–918. 16

Sickles, E.A. (1986). Mammographic features of 300 consecutive nonpalpable breast cancers. *American Journal of Roentgenology*, **146**, 661–663. 16

Sickles, E.A. (1989). Breast masses: mammographic evaluation. *Radiology*, **173**, 297–303. 12

Simoncelli, E.P. & Adelson, E.H. (1996). Noise removal via Bayesian wavelet coring. In *Third Int'l Conf on Image Proc*, vol. I, 379–382, IEEE Sig Proc Society, Lausanne. 78

Simoncelli, E.P. & Farid, H. (1995). Steerable wedge filters. In *International Conference on Computer Vision*, Cambridge, MA. xv, xvi, 86, 87

Simoncelli, E.P. & Farid, H. (1996). Steerable wedge filters for local orientation analysis. *IEEE Trans Image Proc*, **5**, 1377–1382. xv, 78, 86, 87

Simoncelli, E.P. & Freeman, W.T. (1995). The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *Second Int'l Conf on Image Proc*, vol. III, 444–447, IEEE Sig Proc Society, Washington, DC. 78, 85

Simoncelli, E.P. & Portilla, J. (1998). Texture characterization via joint statistics of wavelet coefficient magnitudes. In *Proc 5th IEEE Int'l Conf on Image Proc*, vol. I, IEEE Computer Society, Chicago. 78

Simoncelli, E.P., Freeman, W.T., Adelson, E.H. & Heeger, D.J. (1992). Shiftable multi-scale transforms. *IEEE Trans Information Theory*, **38**, 587–607, special Issue on Wavelets. 78, 80

Smeraldi, F. (2002). Ranklets: Orientation selective non-parametric features applied to face detection. In *Proceedings of the 16th International Conference on Pattern Recognition, Quebec, QC*, vol. 3, 379–382. 4, 88, 129

Smeraldi, F. (2003a). A nonparametric approach to face detection using ranklets. In *Proceedings of the 4th International Conference on Audio and Video-based Biometric Person Authentication, Guildford, UK*, 351–359. 4, 88, 129

Smeraldi, F. (2003b). Ranklets: a complete family of multiscale, orientation selective rank features. Tech. Rep. RR0309–01, Department of Computer Science, Queen Mary, University of London, Mile End Road, London E1 4NS, UK. 4, 88, 95, 129

SMERALDI, F. & ROB, M.A. (2003). Ranklets on hexagonal pixel lattices. In *Proceedings of the British Machine Vision Conference, Norwich, UK*, vol. 1, 163–170. 4, 88, 129

SMITH, M.J. & BARNWELL, T.P. (1984). A procedure for designing exact reconstruction filter banks for tree–structured subband coders. In *Proceedings of the IEEE International Conference ASSP*, 27.1.1–27.1.4. 70

SMITH, M.J. & BARNWELL, T.P. (1986). Exact reconstruction techniques for tree–structured subband coders. *IEEE Transactions on ASSP*, **34**, 434–441. 70

SOFTKY, W.R. (1995). Simple codes versus efficient codes. *Curr. Opin. Neurobiol.*, **5**, 239247. 96

SPACKMAN, K.A. (1989). Signal detection theory: Valuable tools for evaluating inductive learning. In *Proceedings of the Sixth International Workshop on Machine Learning*, 160–163, Morgan Kaufman, San Mateo, CA. 35

STOLLNITZ, E.J., DEROSE, T.D. & SALESIN, D.H. (1996). *Wavelets for Computer Graphics: Theory and Applications*. Morgan Kaufmann Publishers, Inc. 69, 76

STRANG, G. & NGUYEN, T. (1996). *Wavelet and Filter Banks*. Wellesley-Cambridge Press. 69

SWETS, J. (1988). Measuring the accuracy of diagnostic systems. *Science*, **240**, 1285–1293. 35

SWETS, J., DAWES, R.M. & MONAHAN, J. (2000). Better decisions through science. *Scientific American*, **283**, 82–87. 35

TARASSENKO, L. (1998). *Guide to Neural Computing Applications*. Butterwoth–Heinemann. 21, 27

THORPE, S.J. (1990). Spike arrival times: a highly efficient coding scheme for neural networks. In *Parallel Processing in Neural Systems*, 91–94, Elsevier. 96

THURFJELL, E.L. & LINDGREN, J.A. (1996). Breast cancer survival rates with mammographic screening: similar favorable survival rates for women younger and those older than 50 years. *Radiology*, **201**, 421–426. 16

THURFJELL, E.L., LERNEVALL, K.A. & TAUBE, A.A. (1994). Benefit of independent double reading in a population-based mammography screening program. *Radiology*, **191**, 241–244. 16

VAN RULLEN, R. & THORPE, S.J. (2001). Rate coding versus temporal order coding: What the retinal ganglion cells tell the visual cortex. *Neural Computation*, **13**, 1255–1283. 97

VAPNIK, V. (1979). *Estimation of Dependences Based on Empirical Data [in Russian]*. Nauka, Moscow, (English translation: Springer Verlag, New York, 1982). 39, 44

VAPNIK, V. (1995). *The Nature of Statistical Learning Theory*. Springer Verlag. 42, 46

VAPNIK, V. (1998). *Statistical Learning Theory*. J. Wiley. 42

VAPNIK, V. & CHERVONENKIS, A.J. (1968). On the uniform convergence of relative frequencies of events to their probabilities. *Doklady Akademii Nauk USSR*, **181**. 39

VAPNIK, V. & CHERVONENKIS, A.J. (1974). *Theory of Pattern Recognition [in Russian]*. Nauka, Moscow, (German Translation: W. Wapnik & A. Tscherwonenkis, *Theorie der Zeichenerkennung*, Akademie–Verlag, Berlin, 1979). 39, 42, 46

VIDAKOVIC, B. (1996). *Statistical Modeling by Wavelets*. John Wiley & Sons. 69

WARLAND, D.K., REINAGEL, P. & MEISTER, M. (1997). Decoding visual information from a population of retinal ganglion cells. *J. Neurophysiol.*, **78**, 23362350. 96

WELLINGS, S.R., JENSEN, H.M. & MARCUM, R.G. (1975). An atlas of subgross pathology of the human breast with special reference to precancerous lesions. *Journal of the National Cancer Institution*, **55**, 231–273. 11

WOLFE, J. (1977). *Xeroradiography: Breast Calcifications*. Springfield, Ill. 14

ZABIH, R. & WOODFILL, J. (1994). Non-parametric local transforms for computing visual correspondence. In *Proceedings of the Third European Conference on Computer Vision (Vol. II)*, 151–158, Springer-Verlag New York, Inc. 89

# Index