TÉCNICO
LISBOA

Skin Cancer Detection Using Sparse Coding

Tomás Miguel Donga Cardoso

Thesis to obtain the Master of Science Degree in

Eletrical and Computing Engineering

Supervisor(s): Dr. Ana Catarina Fidalgo Barata
Prof. Jorge Dos Santos Salvador Marques.

Examination Committee

Chairperson: Prof. João Fernando Cardoso Silva Sequeira
Supervisor: Dr. Ana Catarina Fidalgo Barata
Member of the Committee: Prof. Mário Alexandre Teles de Figueiredo

June 2019

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

Acknowledgments

I would like to thank Prof. Jorge Marques and Dr. Ana Catarina Barata for supervising my thesis and for everything they taught me and helped me with during this process. I'd also like to thank my family for all their unwavering support, and also my friends and girlfriend, that made my journey easier, funnier, and above all, interesting and engaging. I would not change a second of it.

This work was partially supported by FCT under the project PTDC/EEIPRO/0426/2014.

Resumo

Melanoma é um dos tipos de cancro mais mortíferos. Tal como para outros tipos de cancro, a probabilidade de cura depende da precocidade do diagnóstico. Por este motivo, muitos esforços têm sido feitos no sentido de usar aprendizagem automática para automatizar o processo de deteção de melanoma. Esta tese integra-se neste grupo de trabalhos através do uso de técnicas de representação esparsa incluídas num sistema completo de diagnóstico de melanomas, que incorpora os passos de extracção de características e de classificação, com o intuito de fazer deteção de melanoma em imagens dermoscópicas.

Os métodos usados na tese baseiam-se na representação esparsa dos dados e no uso de dicionários aprendidos a partir dos dados. A eficácia de dicionários e códigos esparsos discriminativos é estudada, bem como a aplicação de agrupamento hierárquico dos átomos dos dicionários para eliminar quaisquer redundâncias. Finalmente o impacto de aprendizagem profunda no sistema descrito é testada através do uso de características extraidas de uma rede convolucional pré-treinada, nomeadamente a VGG19 [1]. O sistema final proposto nesta tese atinge uma sensibilidade de $56,41\%$ e uma especificidade de $71,43\%$ no conjunto de imagens dermoscópicas da competição de 2017 da International Skin Imaging Collaboration (ISIC) e uma sensibilidade de $64,84\%$ e uma especificidade de $88,82\%$ no conjunto de imagens do Interactive Atlas of Dermoscopy (EDRA) que, tendo em conta a simplicidade dos sistemas propostos, são resultados promissores.

Palavras-chave: Deteção de Melanoma, representações esparsas, SVM, redes convolucionais, clustering hierárquico, aprendizagem discriminativa

Abstract

Melanoma of the skin is one of the deadliest cancer types. As for most types of cancer, its chances of being cured greatly increase with the swiftness of its diagnosis. Due to this, great efforts have been put into using machine learning to automate the process of melanoma detection. This thesis joins that field of work by making use of sparse coding techniques embedded in a complete system for melanoma diagnosis that incorporates feature extraction and classification.

The methods used in this thesis are based on the sparse representation of data and dictionaries learned from data. The effectiveness of discriminative dictionaries and sparse codes is studied, as well as the application of hierarchical clustering to the dictionary atoms in order to further cut redundancies. Finally, the impact of deep learning in the aforementioned system is inspected through the use of deep features extracted from a pre-trained *convolutional neural network* (CNN), namely the VGG19 [1]. The systems proposed in this thesis achieve a sensitivity of $56,41\%$ and a specificity of $71,43\%$ for the image dataset from 2017 challenge from the International Skin Imaging Collaboration (ISIC) and a sensitivity of $64,84\%$ and a specificity of $88,82\%$ for the image dataset from the Interactive Atlas of Dermoscopy (EDRA) which, taking in account the simplicity of the systems, show promise.

Keywords: melanoma detection, sparse coding, svm, convolutional neural networks, hierarchical clustering, discriminative learning

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter motivates the use of sparse methods in skin cancer detection, presents the objective of this thesis and outlines its organization.

1.1 Motivation

Cancer is one of the most deadly diseases that currently afflicts the humankind. Amongst all types of cancer, skin cancer is the most common one and melanoma is the most aggressive and deadly type [2]. The American Cancer Intitute estimates that a total of $96,480$ new cases of melanoma skin cancer will be diagnosed in 2019 and $7,230$ people will die from it in the United States [2]. Considering stage-one melanoma patients, and comparing to those that get treatment within 30 days of being biopsied, the ones that get treatment within 30 to 59 days after the biopsy have 5% increased risk of dying and those that are treated more that 119 days after being biopsied have 41% risk of death [3]. This goes to show that early diagnosis is pivotal in order to reduce the mortality rate of this type of cancer.

Because of this, and with the uprise of machine learning and deep learning methods, increasing effort has been put into place in order to apply these methods to automate the process of skin cancer detection [4], [5].

1.2 Problem formulation

There are several types of skin lesions not all harmful to humans. Figure 1.1 shows the different types of skin lesions and their organization.

Melanocytes are the cells lodged in the deeper layer of the epidermis and are responsible for producing a brown pigment called melanin, which gives the skin its tan or brown color. Melanocytic skin lesions are those that originate from these cells. Most melanocytic skin lesions are benign, harmless, if the DNA is not damaged in such a way that the melanocytic cells have malignant proliferation and potentially metastasize [6]. This type of benign melanocytic lesions are called nevus, which in turn also encompasses several types, such as Clark nevus, Dermal Nevus or Spitz nevus, amongst others. How-

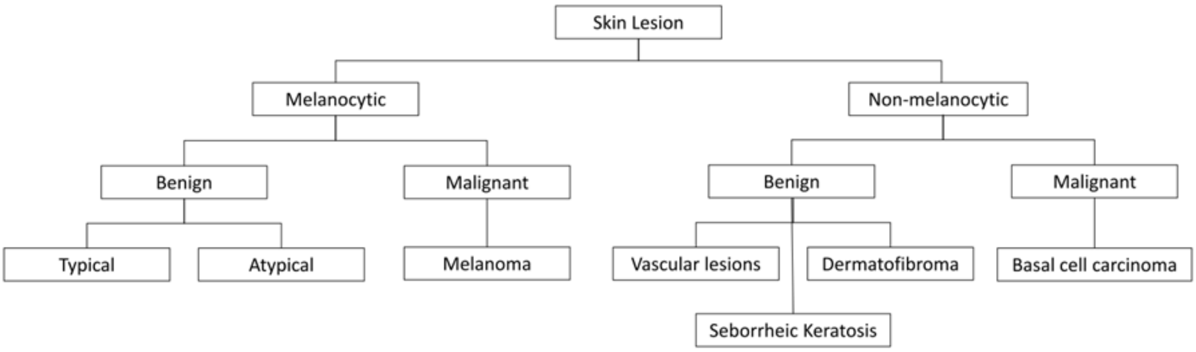


Figure 1.1: Skin lesions diagnosis tree

ever, when there is a malignant proliferation and the cell metastasizes, the skin lesion is then considered a melanoma [6].

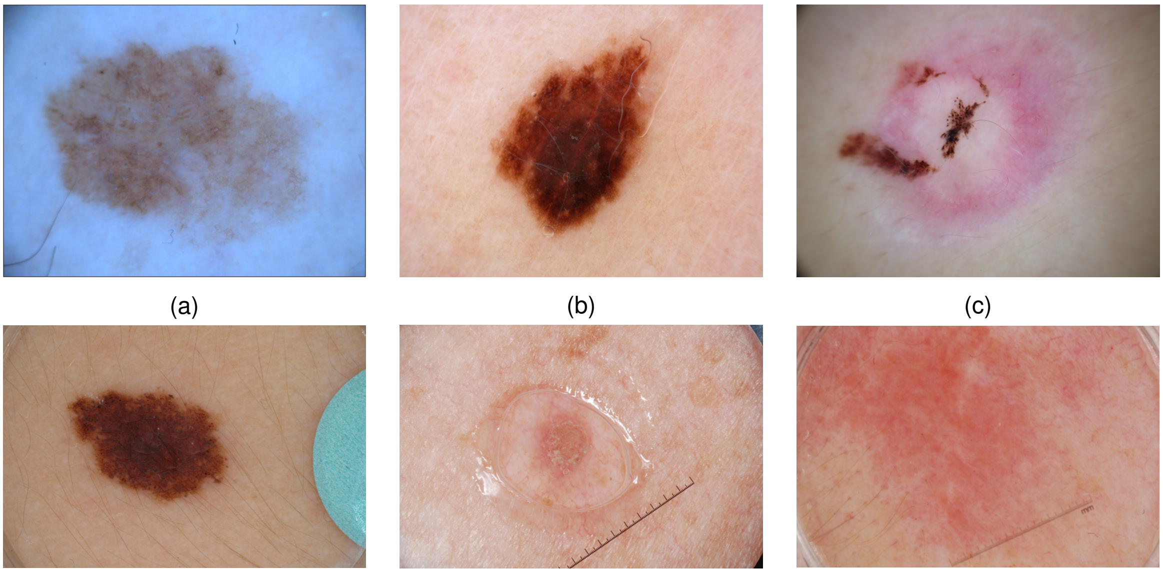
The remaining types of skin cancer originate from other types of cells in the intricate layers of the skin. These encompass benign lesions, such as vascular lesions, seborrheic keratosis, and dermatofibroma, and malignant lesions, such as basal cell carcinoma.

Even though basal cell carcinomas are malignant and harmful, they have a low level of metastasis, meaning the cancer cells do not spread fast, which makes them easier to treat. Melanoma skin lesions however have a very high level of metastasis, this means that the cancer cells will spread to other tissues at a quick rate, greatly hindering treatment, which is why an early diagnosis of melanoma is very important.

Even though some lesions can be diagnosed by a medical professional through naked eye inspection, the use of specific instruments is often required, such as the dermoscope, dermaphot or the stereomicroscope. The last two mentioned devices produce the so called dermoscopic images, some examples of which are presented in Figure 1.2, that are used by computer-aided diagnostic systems (CAD) to perform melanoma detection.

There are medical procedures that are used by medical experts to diagnose skin lesions, the most prominent ones being pattern analysis [8], ABCD rule [9], and seven-point checklist [10]. What all these have in common is that they focus on the analysis of dermoscopic features in the lesion, also called dermoscopic criteria. These include global features, such as shape and pattern of the lesion, and local features, which encompass pigment network, dots/globules, streaks, pigmentation, blue-whitish veil and vascular structures, to name but a few.

For example, the ABCD rule, which can only be applied to melanocytic lesions, assigns scores for four different criteria: asymmetry, border, color, and differential structure, are assigned a score, assessed semiquantitatively, which are then multiplied by given weight factors and a final dermatoscopy score (TDS) is obtained. Depending on this score, the lesion can then be classified as a benign lesion, a suspicious lesion, or highly suspicious for melanoma. The seven-point checklist algorithm is another important procedure and, as the name suggests, consists of observing in the lesion a few of 7 criteria (features normally associated with melanoma lesions). These 7 criteria are broken into major criteria (atypical pigment network, blue-whitish veil and atypical vascular pattern), which get 2 points each, and



(a)

(b)

(c)

(d)

(e)

(f)

Figure 1.2: Examples of dermoscopic images extracted from the ISIC dataset [7]

minor criteria (irregular streaks, irregular pigmentation, irregular dots/globules and regression structures) which get 1 point each. A skin lesion is classified as a melanoma if the total of points for that lesions is equal or higher than 3.

The presence of these features varies for every type of skin lesion, and, even within the same type, every lesion is unique, which is why it is sometimes so difficult to correctly diagnose them. Hence, CAD systems trained from thousands of images may play an important role in assisting medical doctors in skin cancer detection. The goal of this thesis is focused on the development of such computer-aided diagnostic systems.

1.3 Related work

For the past three decades, different approaches have been proposed to tackle the problem of skin cancer detection. The first aproaches took decisions based on global hand-crafted features such as color features (*e.g*, [11], [12]), texture features (*e.g*, [13], [14], [15]), border features (*e.g*, [16], [17], [18], [19]), and asymmetry features, including shape symmetry (*e.g*, [19], [20], [21]) and color and structure symmetry (*e.g*, [19], [22]). Classifiers trained on dictionary-based features, that allowed for information on local image features to be extracted, such as bag-of-words or sparse coding (*e.g*, [23], [15], [24]) were rarely used. Recently, deep learning started to be employed and is gradually becoming the standard in the area, making use of methods such as deep neural networks and transfer learning to achieve state-of-the-art results (*e.g*, [25], [26], [27]).

The performance of machine learning methods for image classification greatly depends on the chosen representation for the image itself and on the quality of the extracted features. Methods that use sparse representations have achieved good results in the past, in several image analysis and processing problems, such as image denoising, deblurring, inpainting, and super resolution [28]. Despite this, few works have employed sparse representation in the context of skin cancer detection. Notorious exceptions are [29], [23], which achieved promising results.

In [29], the effectiveness of different types of features is ascertained, including an ensemble of low-level features, deep features and sparse codes, all used to train separate independent classifiers which are combined through the fusion of classifier output (late fusion). Another comparision between types of local features is also made in [23], in this case between a bag-of-features method and sparse coding, where sparse coding achieved superior results. In both these works, as well as in most challenging machine learning problems, the features are of extreme importance and have a strong influence on the end result. Acknowledging this, [30] made a study of the several type of features used in computer-aided diagnosis systems for skin cancer in the past, which include hand-crafted features, dictionary-based features, deep learning features, and clinically inspired features. Because of this, this thesis is also focused on the type of features extracted, with emphasis on sparse coding.

Table 1.1 summarizes the performance of a few CAD systems. Some systems achieve very good scores but it should be noted that the performances are evaluated on datasets of different size and difficulty, which does not allow for a meaningful comparison.

Table 1.1: Results for some computer-aided diagnostic systems.

Paper	Date	Dataset	SE (%)	SP (%)	*BACC* (%)
[13]	2007	Images from universities	92.34	93.33	$92,84$
[12]	2007	subset of EDRA	87.0	77.3	82.15
[14]	2008	Images from universities	85.3	83.33	$84,315$
[42]	2013	Images from Hospital	98.0	79.0	88.5
[11]	2014	EDRA	61.63	75.83	$68,73$
[24]	2016	PH^2	100.0	90.3	$95,15$
[23]	2017	EDRA	85.5	75.1	$80,3$
[25]	2017	ISIC	$85,6$	$81,2$	$83,4$

1.4 Datasets

This thesis will make use of mainly two different datasets: the dataset associated with the interactive atlas of dermoscopy (EDRA) [31] and the recently-proposed dataset from the International Skin Imaging Collaboration (ISIC) 2017 [7].

ISIC holds a competition every year, centered on the topic of skin cancer. The dataset used in this thesis is the one from the 2017 competition, which has around 2000 images for training, of which less than 400 are labeled as melanoma. It also includes a separate validation set with 150 images and a set of 600 images that serve as test set. ISIC also provides segmentation masks for every image in the dataset, which classifies each pixel as belonging to the lesion or healthy skin.

The EDRA dataset is smaller in comparison to ISIC; a subset of around 800 images will be used, of which around 240 correspond to melanomas.

The exact numbers of images and their division between melanoma and non-melanoma are presented in tables 1.2 and 1.3.

Both datasets will be used to train and test every classification system introduced in this thesis which will also allow for a comparison to be made between both datasets.

Table 1.2: Distribution of images in the ISIC dataset

Subset	Melanoma	Non-melanoma	Total
Train	374	1626	2000
Validation	30	120	150
Test	117	483	600

Table 1.3: Distribution of images in the EDRA dataset

Melanoma	Non-melanoma	Total
241	563	804

1.5 Goals and Outline

This thesis aims to study the effectiveness of sparse coding techniques applied to the problem of melanoma skin cancer detection. This will be done through a number of experiments with different methods and algorithms in order to achieve the best possible result, while also comparing it to a common baseline system.

This thesis will thus be organized as follows:

1. Chapter 1: Introduction - This chapter motivates the work described in the thesis, formulates the problem, and presents a brief discussion of related works.

2. Chapter 2: Sparse Coding - In this chapter, the problem of sparse coding and dictionary learning are discussed and formulated as optimization problems.

3. Chapter 3: Baseline System - This chapter presents the baseline system with sparse features. It will serve as a comparison to the more sophisticated approaches that follow.

4. Chapter 4: Discriminative Dictionary Learning - This chapter introduces a collection of systems focused on the use of discriminative dictionaries for sparse coding.

5. Chapter 5: Deep features - Here, systems that make use of deep learning methods such as convolutional neural networks are introduced for feature extraction, where sparse coding will afterwards be applied.

6. Chapter 6: Comparison and Assessment of the Proposed System - The final system is selected, its performance is compared to other contestants in the ISIC challenge of 2017. A few considerations on the influence of data and dictionary initialization are also made.

7. Chapter 7: Conclusion - This chapter will wrap up the thesis and present the conclusion obtained on the effectiveness of the methods presented for melanoma skin cancer detection.

Chapter 2

Sparse Coding

2.1 Sparse Coding

Methods based on sparse representations of signals have achieved excellent results in a wide range of signal processing problems *e.g*, in image [28], [32], [33] and audio [34], [35] processing. At the core of these methods is sparse coding, which aims to represent an input vector $\mathbf{x} \in \mathbb{R}^n$ through a sparse linear combination of an over-complete set of basis vectors,

$$\mathbf{x} \approx \sum_{j=1}^{k} \mathbf{d}_j \alpha_j, \tag{2.1}$$

where $\mathbf{d}_j \in \mathbb{R}^n, j = 1, ..., k$ are the basis vectors, commonly known as **atoms**, and $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_k]$ is the vector of coefficients that will be called the **sparse code** of $\mathbf{x}$. A sparse linear combination is one in which only a small subset of the coefficients are different from zero, which means that only a few atoms contribute to represent $\mathbf{x}$. The set of atoms $\mathbf{D} = [\mathbf{d}_1, ..., \mathbf{d}_k]$ is called a **dictionary**. By having an over-complete set of atoms, that is, the dimension of the dictionary, k, is greater than the dimension of the input, n, a better representation and characterization of the input $\mathbf{x}$ can be achieved.

Sparsity is a desirable property for a representation model, as it means that the input can be represented using less information and maintain high fidelity. This means that sparsity may eliminate redundancies in the input, while maintaining its important features.

Equation (2.1) represents a single input vector. This idea can be expanded to include several inputs, with equal dimensions, as follows

$$\begin{array}{ccccc} \mathbf{X} & \approx & \mathbf{D} & \times & \boldsymbol{A} \\ n \times p & & n \times k & & k \times p \end{array} \tag{2.2}$$

where $\mathbf{X}$ is a matrix whose columns are the input vectors, n is the dimension of each vector and p the number of vectors. $\mathbf{D}$ is the dictionary with k atoms of dimension n, and $\boldsymbol{A}$ is the matrix with the **sparse codes**, each with dimension k and one for every vector in $\mathbf{X}$.

In order to apply sparse coding, both the dictionary and the sparse codes must be computed and that is achieved by solving two optimization problems.

2.2 Two Optimization Problems

2.2.1 Sparse code computation

The goal of sparse representations is to replicate the input as closely as possible through a linear combination of atoms, as done in (2.1), while enforcing sparsity of the vector of coefficients. How close the sparse representation is to the input can be expressed as $||\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}||_2^2$, which measures the reconstruction error, where $||\cdot||_2^2$ represents the square of the Euclidean norm (l_2 norm). A sparse vector is one such that most of its elements are zero. Thus, a good measure of sparsity is the "l_0 norm" of the vector of coefficients, which actually counts the number elements different from zero. But since it is a cardinal function, it is non-differentiable and difficult to optimize over. Therefore it is often replaced by the l_1 norm, which is applied to the estimation of the sparse codes:

$$||\boldsymbol{\alpha}||_1 = \sum_{j=1}^{k} |\alpha^j|. \tag{2.3}$$

The l_1 norm sums the absolute value of the elements of a vector and makes the problem of obtaining α convex, which facilitates the computation [36]. So, given an input $\mathbf{x}$, and a dictionary $\mathbf{D}$, the sparse code is obtained by solving the optimization problem

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}||\mathbf{x} - \mathbf{D}\alpha||_2^2 + \lambda||\boldsymbol{\alpha}||_1. \tag{2.4}$$

The trade-off between the two terms in the optimization problem is controlled by the variable λ. A high λ would put too much emphasis on the sparsity of $\boldsymbol{\alpha_i}$, but this could mean that the reconstruction error would be bigger. While a very low λ puts emphasis on the reconstruction error in detriment of sparsity, which is not desirable. An balance between both effects should be achieved [37], [38].

2.2.2 Dictionary learning

In most applications, the dictionary needs to be estimated from a set of training data. This is done by encapsulating the optimization over the sparse codes (2.4) with another optimization over the dictionary, which will use the input data and the estimate of the sparse codes to learn a dictionary. Assuming that instead of a single input vector $\mathbf{x}$, there is a set of input vectors $\mathbf{X} = [\mathbf{x_1}, ..., \mathbf{x_p}]$, dictionary learning involves solving the optimization problem

$$\min_{\mathbf{D}} \frac{1}{p} \sum_{i=1}^{p} \min_{\boldsymbol{\alpha_i}} \frac{1}{2}||\mathbf{x_i} - \mathbf{D}\boldsymbol{\alpha_i}||_2^2 + \lambda||\boldsymbol{\alpha_i}||_1, \tag{2.5}$$

where, for every training sample, there is a minimization with respect to the sparse code $\boldsymbol{\alpha_i}$, which involves a trade-off between minimizing the reconstructing error and enforcing the sparsity of the sparse code. This optimization problem translates into solving two alternating, iterative optimization problems, one over the dictionary and the other over the sparse codes, until convergence.

A precaution must be taken, however. Since sparsity is desired, the algorithm will try to lower the l_1 norm of the sparse code, reducing its values. Since the sparse code $\boldsymbol{\alpha_i}$ is reduced, in order to compensate this, the norms of the atoms will increase and this may lead to bad results. Therefore, the atoms must be constrained so that their norm is lower than a given constant $\|\mathbf{d}_j\|_2 < \epsilon$.

2.3 Solving the Optimization problems

Optimizing the cost function (2.5), as mentioned earlier, is commonly known as dictionary learning. A commonly used method for dictionary learning is *Online Dictionary Training* (ODT) [39], which is adopted in this thesis. This algorithm breaks the dictionary learning problem into two optimization problems, as mentioned before: sparse code update and dictionary update. It iterates throughout the training data, alternatingly solving these problems for each training instance. The two steps will be briefly described next.

2.3.1 Sparse code update

In the sparse code update, the dictionary is fixed and the cost functional is minimized with respect to each sparse code. There are a few different algorithms to obtain the sparse code given a dictionary $\mathbf{D} = [\mathbf{d_1}, ..., \mathbf{d_k}]$ and input vectors $\mathbf{X} = [\mathbf{x_1}, ..., \mathbf{x_p}]$, such as *orthogonal matching pursuit* (OMP) [40] and *least angle regression* (LARS) [41]. In this thesis, the LARS algorithm is used. Let $\hat{\mathbf{X}}$ be the sparse representation of $\mathbf{X}$, $\mathbf{r_i} = \mathbf{x_i} - \hat{\mathbf{x}}_i$ the residual of the i-th vector, and α_i^j the coefficient that multiplies atom $\mathbf{d_j}$ in the sparse representation of $\mathbf{x_i}$. The LARS algorithm is as follows:

1. for a sparse code $\boldsymbol{\alpha}_i$, it starts by making all its entries equal to zero.

2. finds the atom $\mathbf{d}_j$ that is the most correlated to the residual $\mathbf{r}_i$

3. increases the coefficient α_i^j in the direction of the sign of this correlation. Do this until some other atom $\mathbf{d}_q$ is just as correlated with $\mathbf{r}_i$.

4. increase α_i^j and α_i^q so that their atoms remain equally correlated with $\mathbf{r}_i$ (increase in their joint least squares direction), until another atom $\mathbf{d}_m$ has as much correlation with $\mathbf{r}_i$

5. continue until some stopping condition.

2.3.2 Dictionary update

The overall dictionary learning algorithm presented in [39] goes through all the training data and updates the dictionary at every iteration. It starts by initializing the dictionary and creating two matrices **A**$_0$ and **B**$_0$ and making them equal to zero. Then, for iteration i, it takes $\mathbf{x_i}$ and the dictionary built thus far to solve the LARS and compute the sparse code for $\mathbf{x_i}$. It then updates the **A**$_i$ and **B**$_i$ matrices as follows:

$$\mathbf{A}_i = \mathbf{A}_{i-1} + \alpha_i \alpha_i^T \;\; and \;\; \mathbf{B}_i = \mathbf{B}_{i-1} + \mathbf{x}_i \alpha_i^T. \tag{2.6}$$

Then, with $\mathbf{D}$ denoting the dictionary built thus far, the algorithm iterates through the n atoms in the dictionary and updates them individually. This pass through the atoms is done as many times as necessary until convergence of the reconstruction error $||\mathbf{x}_i - \mathbf{D}\alpha_i||$. With $\mathbf{A}_i = [\mathbf{a}_1, ..., \mathbf{a}_k]$ and $\mathbf{B}_i = [\mathbf{b}_1, ..., \mathbf{b}_k]$, for iteration j, it computes an intermediary variable $\mathbf{u}_j$:

$$\mathbf{u}_j = \frac{1}{\mathbf{A}_{jj}}(\mathbf{b}_j - \mathbf{D}\mathbf{a}_j) + \mathbf{d}_j, \tag{2.7}$$

and uses it to update the atom $\mathbf{d}_j$:

$$\mathbf{d}_j = \frac{1}{\max(||\mathbf{u}_j||_2, 1)}\mathbf{u}_j \tag{2.8}$$

Chapter 3

Baseline Architecture

This chapter describes a basic CAD system, based on sparse representation of data.

3.1 System overview

This thesis aims to a develop computer-aided diagnosis system for melanoma detection on dermoscopic images. In this chapter an initial baseline system is proposed and described. The problem addressed is a binary classification problem: given a dermoscopic image of a skin lesion, the system should be able to distinguish between melanoma and benign skin lesions.

The baseline system will serve as a stepping stone for improvement and comparison troughout this thesis. The architecture of the baseline system is shown in figure 3.1 and was inspired by the system proposed in [42].

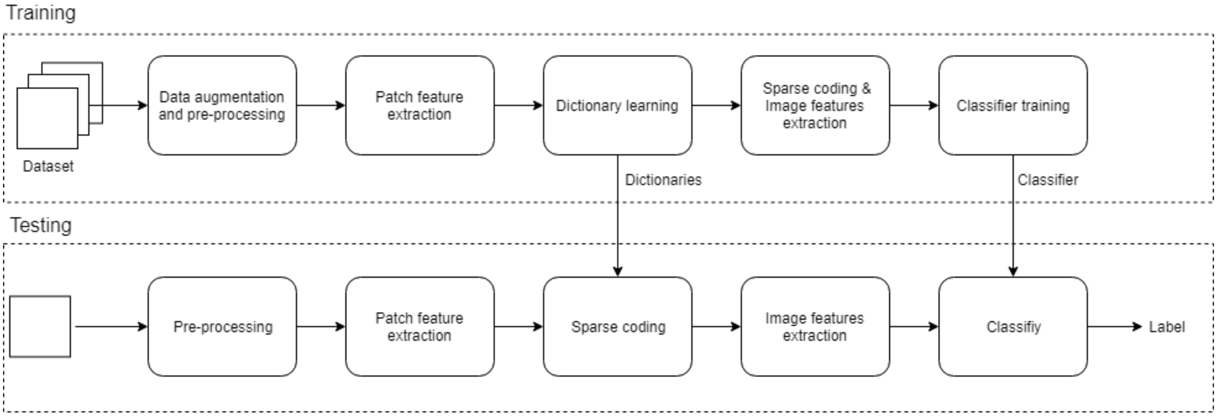


Figure 3.1: Block diagram of baseline system

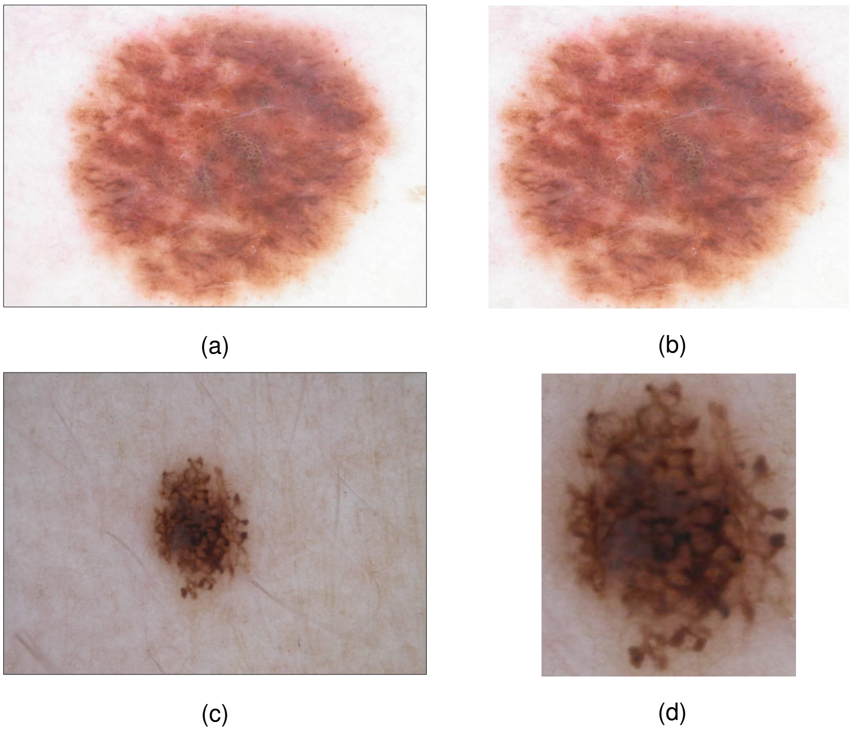
The CAD system involves two modes (training and testing), each of them comprising several tasks. In training, a dataset of dermoscopic images is received, these images are processed in order to normalize their size, as some images are much larger than others, and have different magnifications, and also to normalize their color spectrum, thus standardizing them. Data augmentation is also performed to ensure both classes are approximately equally represented in the training dataset. Two elements need to be trained, the dictionary and the classifier. The dictionary is trained with local image features extracted from image patches and the classifier is then trained with global image features (histograms of sparse codes computed using the previously learned dictionaries).

In testing, the learned dictionaries and classifier obtained in the training phase are used and a label for every image is produced as output, guessing whether the given image is a melanoma or not. The blocks of which the baseline system is composed will be further discussed in detail, in this section.

3.2 Data augmentation and pre-processing

Before feature extraction and classification is done, the images must be loaded and pre-processed. Both datasets are unbalanced class-wise, especially the ISIC dataset, where melanoma skin lesions account for only roughly 20% of the full dataset. The small presence of one of the classes hampers the training of the classifier which may produce subpar results. To tackle this problem, data augmentation techniques were used. For every melanoma image in the training set, three additional images were generated by rotating the image by multiples of 90º. Therefore, for every melanoma image, three additional images were produced, thus balancing the dataset.

Both datasets provide binary masks for the skin lesion in the images that can be used for segmentation. Segmentation is important in this problem, since the size of the skin lesion and the percentage of the image it occupies varies greatly among the dataset, which means some images show a lot of the neighboring skin while others do not, as is depicted in Figure 3.2.


(a)

(b)

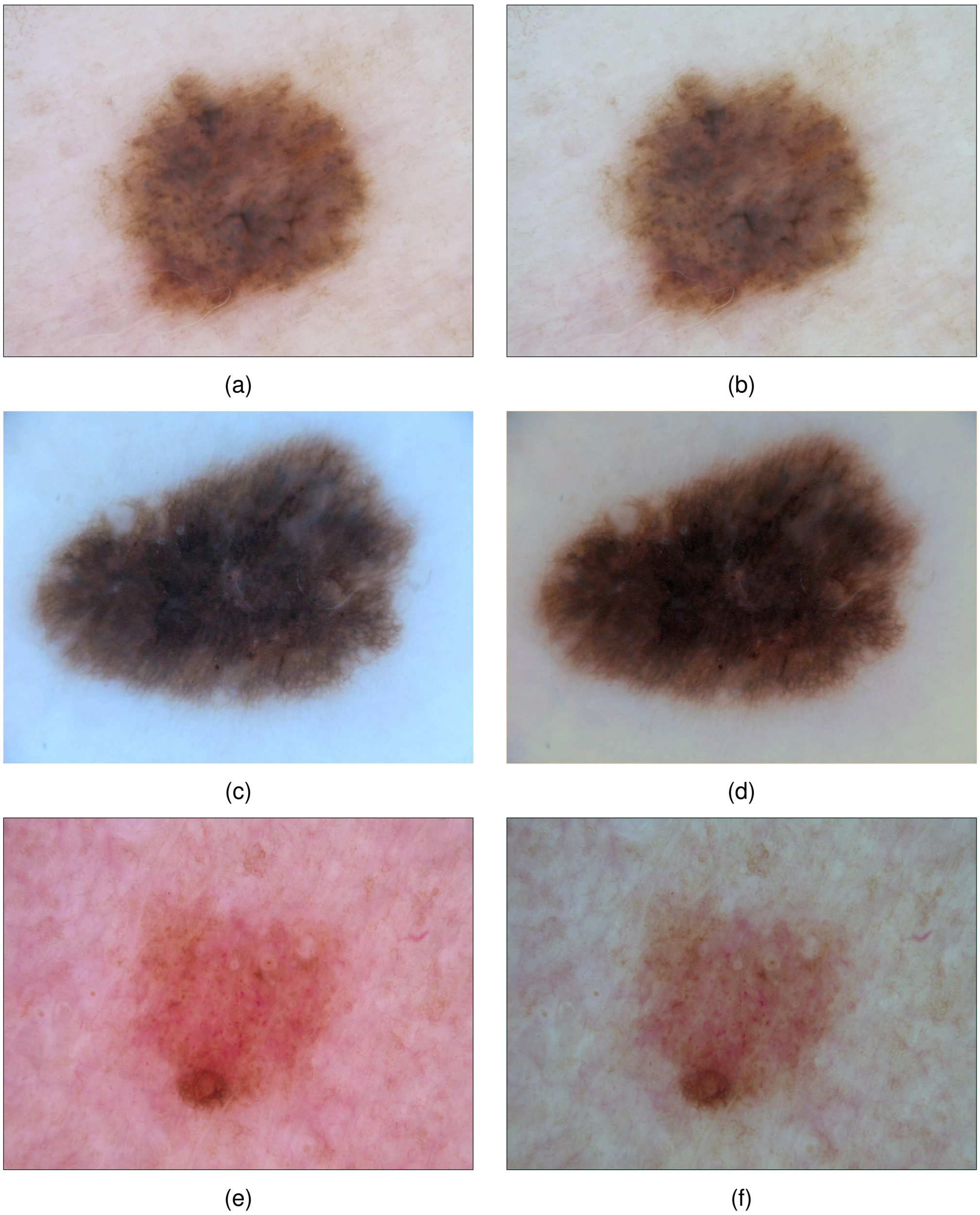
(c)

(d)

Figure 3.2: Examples of images with lesions of different size (left) and their cropping (right). Original images extracted from [7]

The segmentation step crops a bounding box around the lesion, in order to focus the feature extraction procedure on the lesion, discarding the healthy skin. This procedure discards information associated to healthy skin and emphasizes lesion features. This maximizes the differences in features between melanoma and non-melanoma skin lesions, since skin features, which are similar whatever the lesion is, are dismissed. It is also important, however, to avoid cropping immediately at the border of the lesion, since transition from healthy skin to lesion convey useful information about malignancy. Finally, it is also important to maintain the aspect ratio of the images, which varies from image to image. This means that all images can not be of the same size without being distorted, so a limit of 1000 pixels is imposed for each dimension of the image as standard. Therefore, there is no image greater than 1000 pixels high or wide.

Since the images are provided by different hospitals, they are captured by using different dermatoscopes and under different conditions, which results in images with different color spectra. This introduces significant color differences across images, which makes the classification task more difficult, since similarities between in-class images are harder to find. To overcome this difficulty, every image, after being loaded and cropped, goes through a *color normalization* function [43] that makes use of the *Shades of Gray* algorithm [44]. The results of the color normalization function are demonstrated in Figure 3.3.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3.3: Dermoscopic images taken with different dermatoscopes and lighting conditions, before (left) and after (right) color normalization. Original images extracted from [7]

3.3 Patch feature extraction

Once a given image is loaded and goes through pre-processing, its features are extracted. In order to do this, the image is broken into non-overlapping patches of size 16×16 pixels and local features are extracted from each of these. Two types of features were used: color and texture features, which are the two main sources of information for a dermoscopic image.

As further described in [42], there are several options for both color and texture features. Regarding color, the most used features within dermoscopy problems are the color mean and variance within the patch [45], [46]. In this baseline system, however, a slightly more complex type of features will be used: color histograms, which have been successfully adopted in other works [30].

Each pixel in the patch has three color components (RGB channels). The color content of each patch is characterized by three color histograms. The histograms hold the information for the distribution of the three color channels in that specific patch, for that color channel. So, for every patch P_c, $c \in \{1, 2, 3\}$, three histograms h_c with $B = 16$ bins each are computed as

$$h_c(i) = \frac{1}{L^2} \sum_{x=1}^{L} \sum_{y=1}^{L} b_i(P_c(x, y)), \quad i = 1, ..., B, \tag{3.1}$$

where $L \times L$ is the size of patch P_c, and $b_i(\cdot)$ is the characteristic function of the ith bin of histogram h_c

$$b_i(P_c(x, y)) = \begin{cases} 1, & \text{if pixel } P_c(x, y) \text{ belongs to the } i\text{th bin} \\ 0, & \text{otherwise.} \end{cases} \tag{3.2}$$

As for the color features, there are also several options for texture features. Examples include statistics of pairs of neighboring pixels, leading to the co-occurrence matrix [47], aplication of image transforms, such as Fourier transform [48], linear filters [49], Laplacian pyramids [50], or wavelets [51]. In the baseline system, gradient histograms were chosen.

The gradient of an image conveys information on the intensity changes near each pixel. In this case, the gradient of the gray-scale image is computed for every patch. At every pixel (x, y), the horizontal and vertical components of the gradient, $g_1(x, y)$ and $g_2(x, y)$, are computed using Sobel masks. These two components are then used to calculate the gradient magnitude $||g(x, y)||$ and orientation $\phi(x, y)$

$$||g(x, y)|| = \sqrt{g_1^2(x, y) + g_2^2(x, y)}, \qquad \phi(x, y) = \tan^{-1}\left(\frac{g_2(x, y)}{g_1(x, y)}\right). \tag{3.3}$$

The gradient magnitude and orientation information are then used to build two more texture histograms, for every patch, in a similar way to what is done with the color histograms:

$$h_m(i) = \frac{1}{N} \sum_{x=1}^{L} \sum_{y=1}^{L} b_i(||g(x, y)||), \quad i = 1, ..., B_m, \tag{3.4}$$

$$h_\phi(i) = \frac{1}{N} \sum_{x=1}^{L} \sum_{y=1}^{L} b_i(\phi(x, y)), \quad i = 1, ..., B_\phi. \tag{3.5}$$

The gradient orientation is an angle in to the interval $[-\pi, \pi[$, therefore these are used as minimum and maximum. For the magnitude histogram, first the maximum magnitude is found across all images in the training set, with the minimum being zero.

3.4 Dictionary learning and sparse coding

As discussed in chapter 2, sparse coding represents input vectors using an overcomplete dictionary. This idea was applied to feature vectors associated to each patch. In the system described thus far, we have three different sets of features (color histograms, gradient magnitude and gradient orientation histograms), which extract different types of information from the image. Due to this, three dictionaries were learned, one for each type of features. The structure of the system, taking in consideration the three types of features, is shown in Figure 3.4.

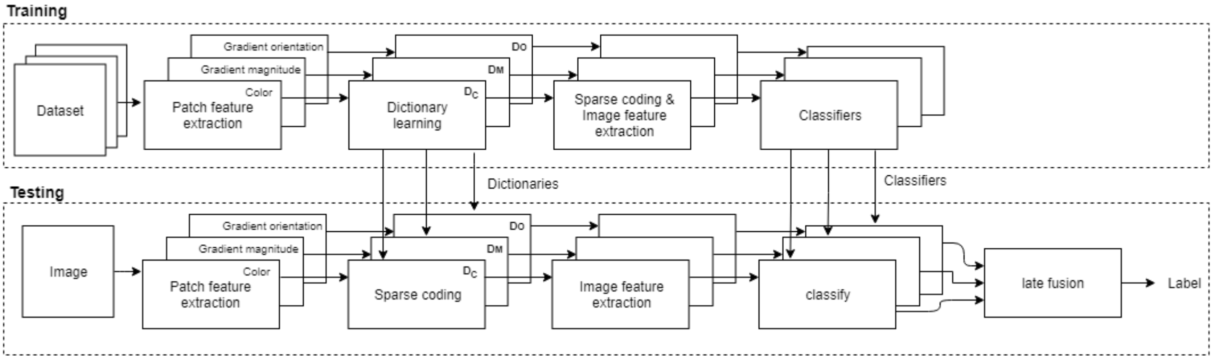


Figure 3.4: Block diagram of the baseline system taking in consideration the different types of features

The dictionaries were built using the *ODL*, described in Section 2.3. The input for dictionary learning are the local feature histograms obtained before, from all the training patches. The sparse representation of a given type of features, which is essentially the matrix notation for sparse coding introduced in section 2.1, can then be written as

$$\mathbf{X} \quad \approx \quad \mathbf{D} \quad \boldsymbol{A}, \tag{3.6}$$

where in the input **X** $\in \mathbb{R}^{n\times p}$, n is the size of the feature vector for one patch, p is the total number of patches in the dataset, **D** $\in \mathbb{R}^{n\times k}$ is the dictionary, with k atoms and $\boldsymbol{A} \in \mathbb{R}^{k\times p}$ are the sparse codes.

3.5 Image features extraction

In order to predict the image label, a set of image features must be extracted. Image features, as the name indicates, are global features that characterize the image as a whole. Therefore, the available sparse codes for a given image, one per patch, must be transformed into a single vector that describes the image.

A sparse code, by definition is a sparse vector of weights that multiplies the atoms in the dictionary to approximate the patch features. If looked at in a different perspective, a sparse code gives information on which atoms are important to characterize its respective patch features. In the context of this problem, the aggregation of sparse codes for a given image gives the importance of each atom in representing the image. These weights are the features chosen to represent the image. However, since the images have different sizes, the number of sparse codes also varies from image to image, which means that they cannot be directly used as features. To circumvent this, a histogram is made averaging the sparse codes of a given image

$$h_s = \frac{1}{p}\sum_{i=1}^{p} \alpha_i, \tag{3.7}$$

which in turns gives the average use of every atom in representing the patches of that image. In (3.7), for a given image, its histogram h_s is computed by averaging the p sparse codes, corresponding to the p patches that form the image.

3.6 Classifier

The final step of this initial CAD system is the classifier, trained with the training data: the global image features and the corresponding labels, in a supervised way.

The chosen classifier for the baseline system is the *Support Vector Machine* (SVM) [52]. The SVM is a popular classifier that has been applied to a wide range of problems. In its simplest version, it tries to learn an hyperplane that separates the training samples from two classes. The hyperplane can be replaced by more complex surfaces by using kernel functions. A brief discussion of the SVM method is found in appendix A.

Separate classifiers were trained for each type of features: color, gradient magnitude, and gradient orientation. The final predicted label for a given image comes from averaging the class-specific probabilities of that image, given by all three classifiers.

3.7 Experimental setup

This section describes the experimental set up used to tune the baseline CAD system.

The system was implemented in python and the SVM classifier was imported from the package *Scikit-learn* [53]. Other packages, like *Numpy* and *Scipy*, amongst others, were used to build several additional functionalities necessary, as well as the *Spams* package for dictionary learning and sparse code computation [54].

With the standard parameters, the SVM classifier could not distinguish between melanoma and non-melanoma skin lesions as it would assign every sample to one of the classes. Further tuning was thus required.

The simpler way to tune a classifier is to train it using the training samples and use the validation samples to tune it. As the ISIC dataset supplies both training and validation data, this would be possible, but a different method was chosen instead. To tune the classifier, k-folds cross-validation was used instead. This consists in splitting the training set into k parts (folds), one of which is used for validation and the others are used to train the classifier. The validation fold will rotate at each iteration such that all folds serve as validation fold. Each of these folds has approximately the same number of samples and is well balanced, that is, has roughly the same amount of samples from both classes. The training set to train and tune the classifier was constituted by the original training and validation data. The test data was never used.

A different procedure was adopted with the EDRA dataset, as it does not have separate sets for training and testing, nested cross-validation was used. Nested cross-validation consists of two cycles, an inner cycle that performs normal cross-validation for model tuning and an outer cycle that evaluates the chosen model on the portion of the dataset that was not used in inner cycle of cross-validation. This allows for a computation of an average test score for the dataset.

The parameters that were tuned during the process for the SVM classifier were: *i)* the penalty parameter C for an error *ii)* the rbf (see appendix A) kernel coefficient γ *iii)* the number of atoms in the dictionaries.

3.8 Results

In order to evaluate the performance of the CAD system, three evaluation metrics were selected: sensitivity (SE), specificity (SP) and balanced accuracy (BACC). These evaluation metrics are further described in appendix B.

The baseline system described in this chapter was trained and evaluated on the aforementioned datasets. In Table 3.1 the average scores for the color, gradient magnitude, and gradient orientation classifiers are presented as well as the global scores for the test set of the ISIC dataset.

In Table 3.2, similar average scores for the individual classifiers are presented, but for the EDRA dataset as well as the average test score obtained in the nested cross-validation.

In both Tables 3.1 and 3.2 two test scores are presented, one that uses all three previously mentioned classifiers to reach the decision and another that uses only the classifiers that are built with the color and gradient magnitude features. This was done because during the cross-validation process, the classifier trained with the gradient orientation features presented quite lower results comparing to the other two. A comparison between these two will then be made to ascertain the viability of using the gradient orientation features and classifier in the baseline system.

Table 3.1: classifier performances on the test set for the ISIC dataset.

Classifier	SE (%)	SP (%)	*BACC* (%)
Color	$43,59$	$75,16$	$59,37$
Magnitude	$56,41$	$63,77$	$60,09$
Orientation	$57,27$	$54,04$	$55,65$
Fusion with 3 classifiers	$47,01$	$73,71$	$60,36$
Fusion with 2 classifiers	$47,01$	$75,36$	$\mathbf{61,19}$

Table 3.2: classifier performances by nested cross-validation for the EDRA dataset.

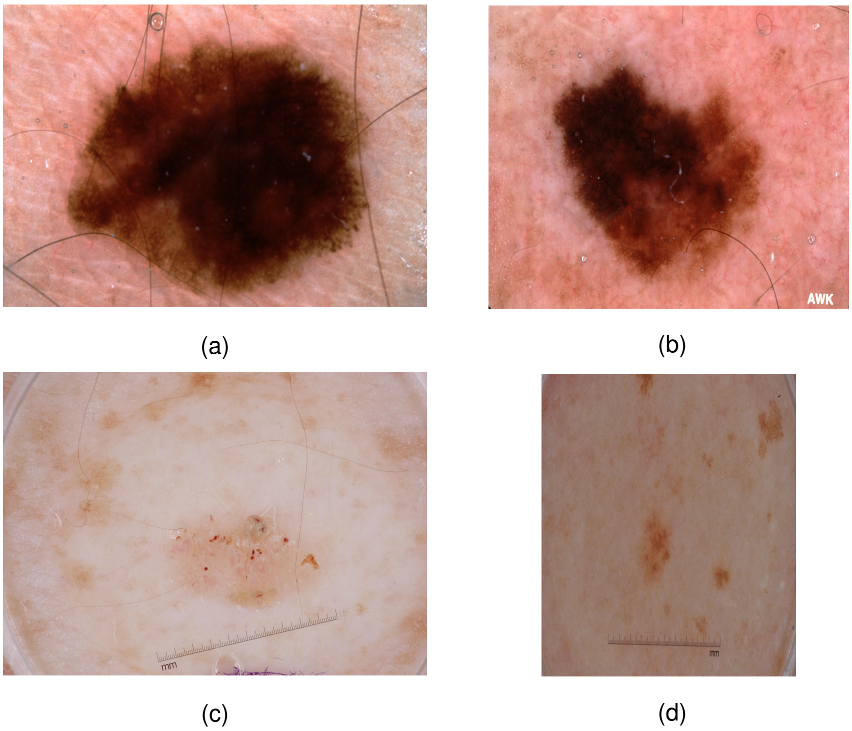
Classifier	SE (%)	SP (%)	*BACC* (%)
Color	$64,67$	$82,18$	$73,42$
Magnitude	$55,50$	$77,97$	$66,73$
Orientation	$55,53$	$71,28$	$63,41$
Fusion with 3 classifiers	$60,11$	$87,63$	$73,87$
Fusion with 2 classifiers	$61,40$	$86,51$	$\mathbf{73,96}$

As can be observed, there is a big difference in the results obtained in both datasets. This difference is mainly due to the variety and difficulty of the images in these datasets. Both of them can be considered as difficult datasets, compared with others which have been used in dermoscopy analysis. However, the ISIC dataset is significantly harder.

It should also be noted that the proposed CAD system achieves better results for both datasets if the classifier trained on the orientation features is excluded. Therefore, **orientation features will be excluded from now on, in this thesis**.

Figures 3.5 and 3.6 show examples of images correctly and incorrectly classified, for the ISIC and EDRA datasets, respectively. Just by observing the many similarities between correctly and incorrectly classified images, one can get an idea of just how difficult this classification problem is.

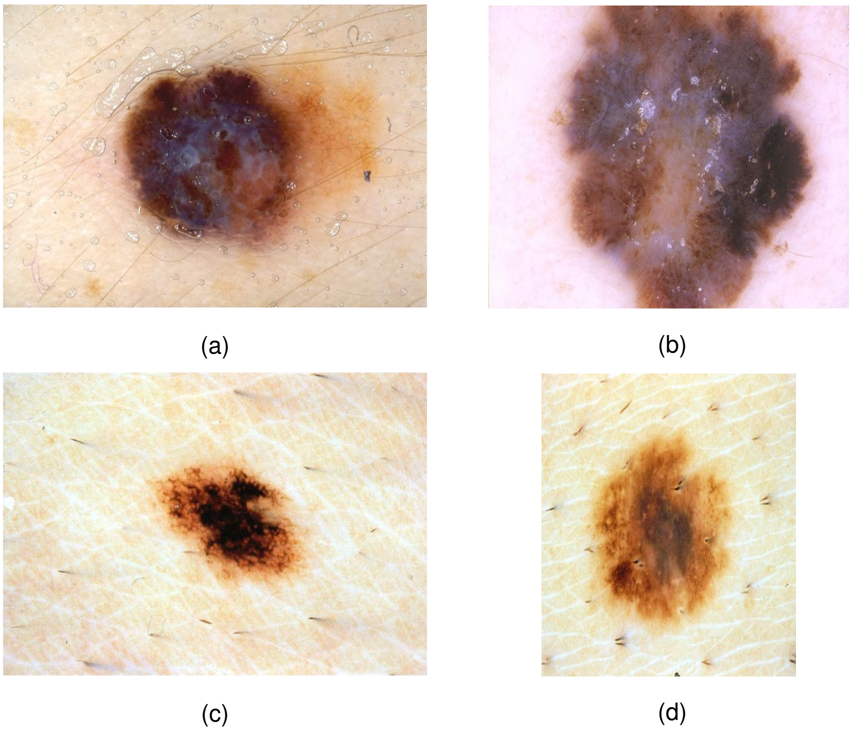
The **baseline system** to be considered is then the one presented in this chapter but with the exclusion of the gradient orientation features and their classifier. Therefore, there are two types of features, which are used to train to dictionaries and consequently, two classifiers whose decision scores are averaged to reach a final decision on whether a skin lesion is a melanoma or not.



(a) (b)

(c) (d)

Figure 3.5: Image classification in the ISIC dataset: (a) True positive (b) False positive (c) True negative (d) False negative.



(a) (b)

(c) (d)

Figure 3.6: Image classification in the EDRA dataset: (a) True positive (b) False positive (c) True negative (d) False negative.

Chapter 4

Discriminative Dictionariy Learning

When sparse coding was first introduced in image representation tasks, it was used to reduce the dimensionality of the data [28] [55] and improve the performance of different tasks, such as image denoising. With the adoption of sparse representations in image classification tasks, it was not only necessary for the learned dictionaries to accurately represent the image, but also to provide discriminative information regarding the different classes. Therefore, different works started to propose strategies to enhance the discriminative properties of the dictionaries [56], [57], [58].

A discriminative dictionary contains specific subsets of atoms that are more specialized in representing a given class. For example, given two classes A and B, a set of atoms in the dictionary is more consistently used by the inputs of class A, while another set of atoms is more consistently used for inputs of class B. This means that if the inputs show a high inter-class variability and low intra-class differences, they will tend to select the same atoms within the dictionary, as other inputs of the same class [56]. This enhances the ability to distinguish between inputs of different classes since their sparse codes would be very different, which is the goal of classification.

4.1 Concatenation of class-specific dictionaries

In the current problem (melanoma detection), there are two classes, melanoma skin lesions and non-melanoma skin lesions. As mentioned above, a discriminative dictionary is one that contains separate atoms to represent each of the classes. The simplest strategy to obtain a discriminative dictionary is to simply learn one dictionary using melanoma training images and learn another dictionary using non-melanoma images, and then concatenate them.

The block diagram for this new classification system is shown in figure 4.1. It is identical to the baseline system apart from the dictionary learning step, marked in green, which was modified.

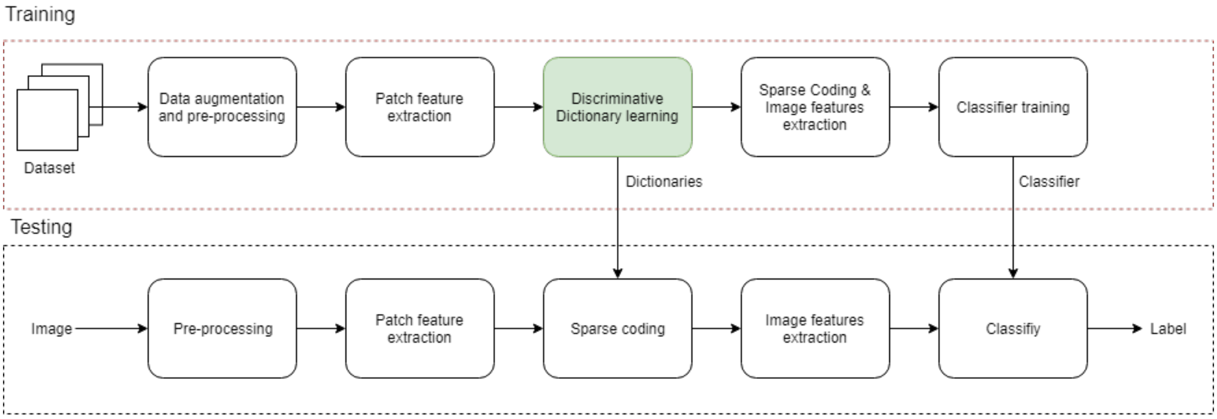


Figure 4.1: Block diagram of classification system with discriminative dictionaries obtained through concatenation of class-specific dictionaries. The highlighted block in green is the one modified with respect to the baseline system

In order to assess the impact of using this type of discriminative dictionaries, the baseline was slightly modified to include these dictionaries and the results are shown in Table 4.1 for the ISIC dataset and Table 4.2 for the EDRA dataset. As mentioned in section 3.8, the orientation features and its classifier were dropped due to poor results, thus only the color and magnitude of gradient will be considered.

Table 4.1: Individual classifier performances and performance on the test set for ISIC dataset applied to baseline with concatenation of class-specific dictionaries.

Classifier	SE (%)	SP (%)	*BACC* (%)
Color	$47,09$	$61,49$	$54,25$
Magnitude	$45,3$	$72,26$	$58,78$
Fusion	$44,44$	$72,88$	$58,66$

Table 4.2: Individual classifier performance and nested cross-validation performance for the EDRA dataset applied to baseline with concatenation of class-specific dictionaries.

Classifier	SE (%)	SP (%)	*BACC* (%)
Color	$65,65$	$84,23$	$74,93$
Magnitude	$57,46$	$76,64$	$67,05$
Fusion	$64,84$	$88,82$	$76,83$

The results evolved in opposite directions for both datasets. Applying discriminative dictionaries to the ISIC dataset led to worse performances for both the individual and combined classifiers. However, the application of discriminative dictionaries to the EDRA dataset improved the results. The performances of all the classifiers slightly improve, as well as the overall balanced accuracy. The sensibility improved $3,44\%$, the specificity improved $2,11\%$, while the balanced accuracy improved $2,87\%$, which shows promise.

The main limitation of the method used in this section is that there is no guarantee that it learns a dictionary that has completely separated atoms to represent the two classes, since melanoma and non-melanoma skin lesions share several properties. Thus, there will be some atoms that can be easily used by both classes. Making sure the dictionaries represent only their specific class might improve the result, since the resulting histograms that serve as image features will have a bigger incidence in different sets of bins for different classes.

4.2 Concatenation of class-specific Sparse Codes

Instead of concatenating the class-specific dictionaries and computing the sparse codes for the resulting dictionary, we will instead estimate two sparse codes for each image patch: one for the melanoma dictionary and another one for the non-melanoma. Then, we will concatenate them into one single sparse code. This is done for both types of features and two classifiers are trained, identically to the baseline.

The block diagram for this new classification system is shown in Figure 4.2. The modified sparse coding block is highlighted in green.

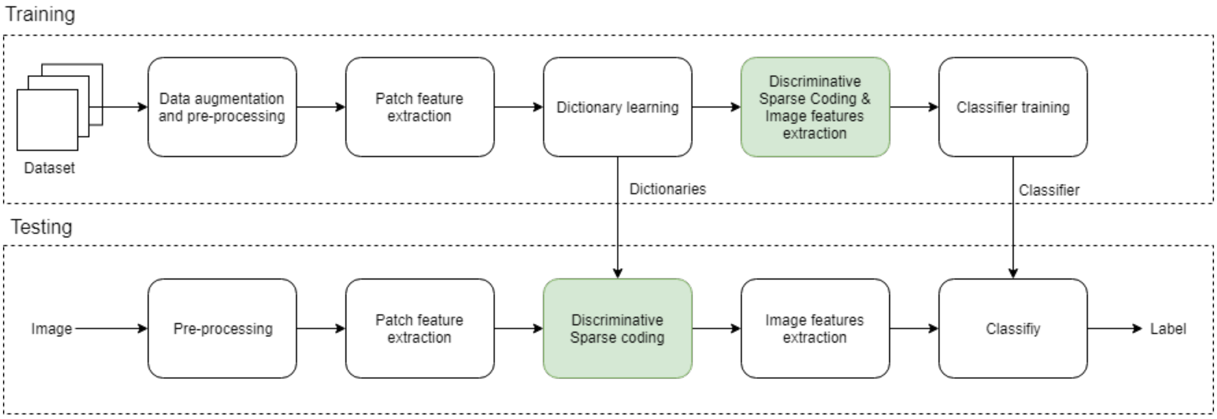


Figure 4.2: Block diagram of classification system with concatenation of class-specific sparse codes. The highlighted blocks in green are the ones modified with respect to the baseline system

The results for the ISIC and EDRA datasets are presented, respectively, in Tables 4.3 and 4.4.

Table 4.3: Individual classifier performances and overall test performance for ISIC dataset applied to baseline with concatenation of class-specific sparse codes.

Classifier	SE (%)	SP (%)	*BACC* (%)
Color	$39,32$	$76,81$	$58,06$
Magnitude	$50,43$	$64,39$	$57,41$
Fusion	$41,03$	$75,57$	$58,3$

Table 4.4: Individual classifier performances and overall test performance for EDRA dataset applied to baseline with concatenation of class-specific sparse codes.

Classifier	SE (%)	SP (%)	*BACC* (%)
Color	$63,27$	$84,77$	$74,02$
Magnitude	$60,00$	$79,35$	$67,67$
Fusion	$61,24$	$87,59$	$74,41$

This method achieved worse results than the methodology described in Section 4.1. Once again, the results for the ISIC dataset were overall worse than the ones achieved by the baseline. For the EDRA dataset, this method still managed to obtaine a marginal improvement of $0,45\%$ with respect to the baseline.

4.3 Clustering of dictionary atoms

As mentioned in the previous sections, one of the possible reasons for discriminative dictionaries not working as well as expected is the similarity between inter-class images, which results in the existence of atoms that are common to both classes in the class-specific dictionaries. A logical step to address this issue would be to remove the common atoms, in order to improve the discriminative properties of the dictionaries. This may constrain the images to use more class-specific atoms, improving the classification.

Ensuring that the class-specific dictionaries do not share common atoms has already been adopted in other works. In [56], the optimization problem for the dictionary learning is changed such that there are class-specific dictionaries that contain the most distinctive atoms which are used for classification, as well as a common dictionary, only used for representation. A different strategy is adopted in this work. First, a separate dictionary is learned for each of the classes. Then, the two dictionaries are concatened. Finally, similar atoms are removed using a hiearchical clustering algorithm. In the following subsection we detail the adopted clustering approach.

4.3.1 Hierarchical clustering

Given a collection of vectors, the goal of agglomerative hiearchical clustering is to iteratively group them until there is only a single cluster, such that at the beginning each vector represents a cluster and by the end only one cluster will remain. At each step of the algorithm, two clusters are grouped together if they are the most ”similar” ones. The ”similarity” between clusters can be measured using a single or complete-linkage strategy. Different metrics can be used to compute the distances, such as Euclidean distance, correlation, cosine distance, among others [59] [60].

In context of this thesis, in the first step of the algorithm, each cluster corresponds to an atom in the dictionary. It is not desirable that only one cluster remains, that is, only one atom. Hierarchical clustering must be performed until a stopping criterion is met. The criterion used in this thesis is that the maximal distance between clusters must be below a given threshold. This distance, defined as clustering threshold, is considered to be a hyperparameter of the model, which is tuned using cross-validation, similarly to the other hyperparameters. The atoms identified as common to both classes are placed in a separate dictionary and discarded. The remaining atoms from both class-specific dictionaries will be concatenated, as in section 4.1.

The block diagram for this new classification system is shown in figure 4.3. It differs from the baseline system by an additional block, marked in red, which applies single-linkage clustering to the learned discriminative dictionaries.

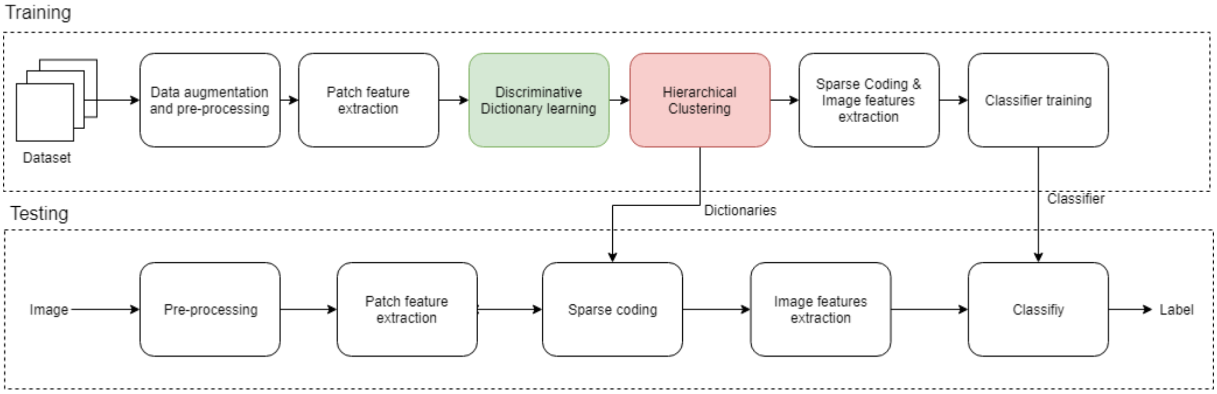


Figure 4.3: Block diagram of classification system with clustering of dictionary atoms. The highlighted blocks in green are the ones modified with respect to the baseline system

The results for the ISIC and EDRA datasets are presented, respectively, in Tables 4.5 and 4.6.

Table 4.5: Individual classifier performances and test performance for ISIC dataset applied to the baseline system with atom clustering applied to a concatenation of class-specific dictionaries.

Classifier	SE (%)	SP (%)	*BACC* (%)
Color	$45,3$	$68,32$	$56,81$
Magnitude	$55,56$	$57,143$	$56,35$
Fusion	$51,28$	$68,74$	$60,01$

Table 4.6: Individual classifier performances and test performance for EDRA dataset applied to the baseline system with atom clustering applied to a concatenation of class-specific dictionaries.

Classifier	SE (%)	SP (%)	*BACC* (%)
Color	$60,7$	$82,04$	$71,52$
Magnitude	$55,95$	$73,91$	$64,93$
Fusion	$61,52$	$83,79$	$72,66$

With respect to the ISIC dataset, performing hierarchical clustering of dictionary atoms improved the results comparing to the discriminative approach presented in Section 4.1. Despite this, it still fell slightly short of the performance of the baseline system. For the EDRA dataset, with a balanced accuracy of $72,66\%$, this method proved to be worse than all methods tested in this chapter and the baseline system.

Chapter 5

Deep Features

Thus far, the focus on the improvement on the baseline system has been in the dictionary learning and sparse coding section, using the hand-crafted features detailed in Section 3.3. This chapter aims at investigating the use of a different kind of features, extracted with a convolutional neural network (CNN).

5.1 Deep learning

With the exponential improvement of computing capacity in recent years, paired with an increase in the number of images available for model training, deep learning methods and convolutional neural networks (CNNs) in particular, became the state of the art in a wide range of image classification tasks [61] [62] [63], including skin cancer detection [64] [26] [27] [25] [65].

Unlike other pattern recognition methods, such as the one presented in Chapter 3, which require the design of hand-crafted features, neural networks are able to learn the features themselves, directly from the images. This makes them easy to use, which also fuels their popularity.

An example of a CNN is shown in Figure 5.1. The basic architecture of a convolutional neural network typically comprises three different types of layers [66]. The convolution layers consist of a number of small filters, which are applied to the input image, or to the outputs of previous layers. This results in a bank of feature maps. These maps then go through an activation function such as the rectified linear unit (ReLu) [67] [68]. This convolutional process may be followed by a pooling layer, which reduces the spatial size of of its input. A commonly used pooling function is max pooling [69]. Max pooling passes a filter through the feature maps, keeping only the highest value caught by the filter. At the end of the CNN, there are fully connected layers, responsible for performing classification. The final layer normally has the same number of units as the number of classes in the classification problem. In this case, the activation function is a softmax function that transforms the output into a number in the range $[0, 1]$, which gives the probability of a given image belonging to a given class.

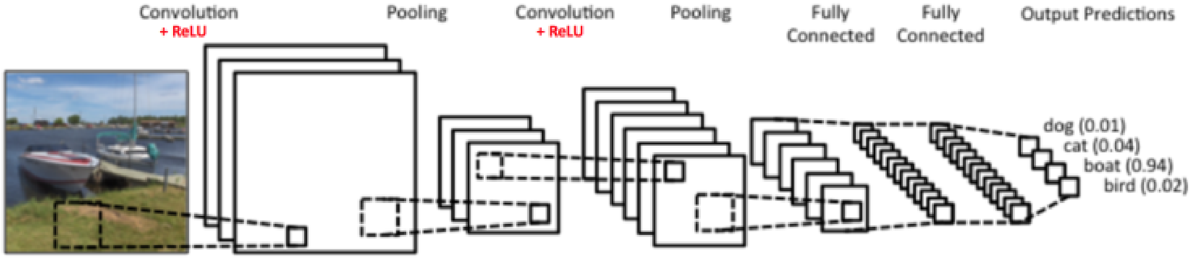


Figure 5.1: Example of a convolutional neural network. Source [70]

5.2 Deep features

This section describes the experiments using features extracted with a CNN, namely, the VGG19 [1], a convolutional neural network with 19 layers, trained on the ImageNet database [71]. A brief explanation of the VGG19, as well as the extracted features is found in appendix C.

5.2.1 Deep features applied in baseline

First of all, the deep features were used in the baseline system, replacing the hand-crafted ones. This means that instead of two types of features (color and magnitude of gradient) that are used to obtain two dictionaries and classifiers, we will estimate only one dictionary and classifier.

The results for the baseline using exclusively the deep features are presented in Tables 5.1 and 5.2 for the ISIC and EDRA datasets, respectively, as well as the previous results for the baseline system and the fusion of the 3 classifiers.

Table 5.1: Individual classifier performances and test performance for ISIC dataset with the inclusion of deep features in the baseline system.

Classifier	SE (%)	SP (%)	*BACC* (%)
Color	$43,59$	$75,16$	$59,37$
Magnitude	$56,41$	$63,77$	$60,09$
Deep	$51,28$	$62,11$	$56,7$
Fusion with 2 classifiers	$47,01$	$75,36$	$\mathbf{61,19}$
Fusion with 3 classifiers	$47,01$	$75,16$	$61,08$

Table 5.2: Individual classifier performances and test performance for EDRA dataset with the inclusion of deep features in the baseline system.

Classifier	SE (%)	SP (%)	*BACC* (%)
Color	$64,67$	$82,18$	$73,42$
Magnitude	$55,50$	$77,97$	$66,73$
Deep	$52,19$	$78,86$	$65,53$
Fusion with 2 classifiers	$61,40$	$86,51$	$73,96$
Fusion with 3 classifiers	$63,37$	$86,95$	$\mathbf{75,16}$

The system based on deep features seems to achieve lower performances than both of the hand-crafted ones. Similarly to the experiments with the baseline model, the late fusion between the hand-crafted and deep classifiers was performed (see the bottom rows of the two tables). This late fusion produced different results for the datasets. While for the ISIC dataset, the balanced accuracy remained somewhat the same, the results for the EDRA dataset benefited from the inclusion of deep features with an overall increase in sensitivity, specificity and balanced accuracy.

5.2.2 Deep features applied in discriminative dictionaries

The deep features are now incorporated in the system that uses discriminative dictionaries, described in Section 4.1. Using this system, the deep features once again replaced the hand-crafted ones and a single classifier was used to classify the test images whose results are presented in Tables 5.3 and 5.4 for the ISIC and EDRA datasets, respectively. Adopting the same strategy to report the results as in the previous section, we also show the performances for the fusion of the three classifiers.

Table 5.3: Individual classifier performances and test performance for ISIC dataset with the inclusion of deep features in the system with class-specific dictionaries.

Classifier	SE (%)	SP (%)	*BACC* (%)
Color	$47,09$	$61,49$	$54,25$
Magnitude	$45,30$	$72,26$	$58,78$
Deep	$56,41$	$54,24$	$55,33$
Fusion with 2 classifiers	$44,44$	$72,88$	$58,66$
Fusion with 3 classifiers	$47,01$	$73,09$	$\mathbf{60,05}$

Table 5.4: Individual classifier performances and test performance for EDRA dataset with the inclusion of deep features in the system with class-specific dictionaries.

Classifier	SE (%)	SP (%)	*BACC* (%)
Color	$65,65$	$84,23$	$74,93$
Magnitude	$57,46$	$76,64$	$67,05$
Deep	$54,22$	$79,02$	$66,62$
Fusion with 2 classifiers	$64,84$	$88,82$	$\mathbf{76,83}$
Fusion with 3 classifiers	$62,21$	$88,54$	$75,38$

The combination of deep features with discriminative dictionaries showed different behaviours for the two datasets. The inclusion of deep features and a third classifier in the late fusion improved the results comparing to using only two classifiers, with the ISIC dataset.

5.2.3 Deep features applied in discriminative sparse codes

The deep features are now incorporated in the system presented in Section 4.2, *i.e.*, the one that computes separate sparse codes for each class. The performances of the individual classifiers and the late fusion with and without the classifier trained with deep features are presented in tables 5.5 and 5.6 for the ISIC and EDRA datasets, respectively.

Table 5.5: Individual classifier performances and test performance for ISIC dataset with the inclusion of deep features in the system with class-specific sparse codes.

Classifier	SE (%)	SP (%)	*BACC* (%)
Color	$39,32$	$76,81$	$58,06$
Magnitude	$50,43$	$64,39$	$57,41$
Deep	$51,28$	$59,63$	$55,46$
Fusion with 2 classifiers	$41,03$	$75,57$	$58,30$
Fusion with 3 classifiers	$47,01$	$76,19$	$\mathbf{61,6}$

Table 5.6: Individual classifier performances and test performance for EDRA dataset with the inclusion of deep features in the system with class-specific sparse codes.

Classifier	SE (%)	SP (%)	*BACC* (%)
Color	$63,27$	$84,77$	$74,02$
Magnitude	$60,00$	$79,35$	$67,67$
Deep	$55,42$	$79,78$	$67,60$
Fusion with 2 classifiers	$61,24$	$87,59$	$74,41$
Fusion with 3 classifiers	$61,29$	$88,82$	$\mathbf{75,06}$

The inclusion of deep features in this system has improved the performance in both datasets. Even though the system presented in Section 4.2 achieved lower results than the baseline system for the ISIC dataset, the inclusion of a dictionary and a classifier based on deep features, in the decision process increased the results on the test set, both with respect to the results presented in 4.2 and the previous approaches presented in this chapter.

The performance of this system with the inclusion of deep features in the EDRA dataset also improved the results comparing to the one presented in Section 4.2.

5.2.4 Deep features in system with clustering of dictionary atoms

The deep features are now included in the system presented in Section 4.3. The performances of the individual classifiers and the late fusion with and without the classifier trained with deep features are presented in Tables 5.5 and 5.6 for the ISIC and EDRA datasets, respectively. Once again, the inclusion of deep features and a third classifier in late fusion achieved better results for both datasets. Even though other configurations achieved better performances for the EDRA dataset, this is the best performing model for ISIC.

Table 5.7: Individual classifier performances and test performance for ISIC dataset with the inclusion of deep features in the system with atom clustering applied to a concatenation of class-specific dictionaries.

Classifier	SE (%)	SP (%)	*BACC* (%)
Color	45,30	68,32	56,81
Magnitude	55,56	57,14	56,35
Deep	54,7	62,94	58,82
Fusion with 2 classifiers	51,28	68,74	60,01
Fusion with 3 classifiers	56,41	71,43	**63,50**

Table 5.8: Individual classifier performances and test performance for EDRA dataset with the inclusion of deep features in the system with atom clustering applied to a concatenation of class-specific dictionaries.

Classifier	SE (%)	SP (%)	*BACC* (%)
Color	60,70	82,04	71,52
Magnitude	55,95	73,91	64,93
Deep	55,74	79,45	67,60
Fusion with 2 classifiers	61,52	83,79	72,66
Fusion with 3 classifiers	61,37	86,29	**73,83**

Chapter 6

Comparison and Assessment of the Proposed System

In this section, we compare our best performing system, presented in Section 5.2.4, with the participants of the 2017 ISIC challenge [72]. Some experiments to assess the the relevance of initialization in the dictionary estimation and the amount of data are also performed.

6.1 In the context of the ISIC 2017 challenge

As mentioned before, the ISIC dataset is provided by the International Skin Imaging Collaboration, which holds a challenge every year on skin cancer detection. This particular dataset is from the 2017 edition [72]. Thus it is possible to compare the proposed model with the contestants of that edition.

There are three sub-challenges each year: lesion segmention, lesion dermoscopic feature extraction, and lesion classification. The lesion classification challenge also ranks and awards the contestants in three categories:

- ROC AUC for melanoma classification;
- ROC AUC for seborrheic keratosis classification;
- ROC AUC for melanoma and seborrheic keratosis classifications combined (mean value).

Since this thesis is solely focused on melanoma classification, only that segment of the challenge will be considered. The receiving operator characteristics curve (ROC curve) is a graph that compares sensitivity and specificity at all classification thresholds [73]. The SVM classifier assigns all images a score in the range $[0, 1]$ and the classification threshold is the value at which the decision is made. This probabilities are calibrated using *Platt scaling* [74]. For example, if the classification threshold is 0.5, all scores below 0.5 are assigned to one class and all values above it are assigned to the other. The area under the curve (AUC) of the ROC curve gives a measurement of performance across all possible classification thresholds [73] [75].

The ROC AUC score for our system, as well as the scores of a small selection of the contestants is presented in Table 6.1. The table not only presents the ROC AUC scores of the contestants, but also their sensitivity, specificity and BACC as well as their final position on the category for melanoma classification.

Paper	Position	ROC AUC	SE (%)	SP (%)	*BACC* (%)
[76]	1^{o}	$87,40$	$54,70$	95	$74,85$
[77]	5^{o}	$83,60$	$35,00$	$96,5$	$65,75$
[78]	9^{o}	$79,10$	$17,10$	99	$58,05$
[79]	11^{o}	$78,30$	$47,00$	$91,5$	$69,25$
[80]	15^{o}	$75,90$	$30,80$	$95,9$	$63,35$
[81]	17^{o}	$71,50$	$40,20$	$81,2$	$60,70$
Proposed	$--$	$64,48$	$56,41$	$71,43$	$63,50$
[82]	19^{o}	$63,60$	$10,30$	$93,2$	$51,75$
[83]	20^{o}	$62,30$	$41,90$	$82,8$	$62,35$
[84]	22^{o}	$49,50$	$47,00$	$51,1$	$49,05$

Table 6.1: Results for some contestants of the 2017 ISIC challenge.

In the context of the ISIC 2017 competition leader board for the melanoma classification category, our system is on the $78,26\%$ percentile, which means that $78,26\%$ of the contestants achieved a higher ROC AUC than us, which is not very good. However, if the leader board was arranged according to the BACC, the proposed system would rank in the 39.13% percentile, which is significantly higher.

It should be noted that the ISIC challenge allows for the use of external data, not provided by them. A fraction of contestants use external data and it may be unfair to compare their systems with those that are trained without extra data, given the relevant role training data plays in classifier performance. If the systems trained with extra data are then excluded, our system would be in the $71,43\%$ percentile for the ROC AUC and the $35,71\%$ for the balanced accuracy.

It should be taken into consideration that in this thesis we selected the best model based on the metrics described in appendix B, not on ROC AUC which was used by the other methods. Similarly, the same reasoning applies when comparing the sensitivity, specificity and balanced accuracy these systems achieved with ours, since these systems did not aim to get the best possible results on these evaluation metrics.

6.2 Relevance of Dictionary Initialization and Datasets

This section discusses the influence of dictionary initialization and of the training set in the final results.

6.2.1 Dictionary initialization

By inspection of the experimental results we realized that, not only the parameters (number of atoms, clustering threshold, C and γ for the SVM) tuned in the cross-validation influenced the performance on the test set, but the dictionary initialization as well. The dictionary initialization is handled by the dictionary learning function from the spams package [54]. This toolbox randomly takes k feature vectors from the training set to serve as the initial atoms. This introduces a variability in the estimated dictionaries and influences the classification performance of the model.

We wanted to determine the degree of influence of the initialization process. Thus, we carried out a simple experiment that consisted of selecting the best configurarion of parameters for the ISIC 2017 obtained through cross-validation and the model described in Section 5.2.4, learn a set of 100 dictionaries, and use them to train 100 classification systems. Then, the sensitivity, specificity, and balanced accuracy were computed for each of them, on the test set. The results are presented in Figures 6.1, 6.2 and 6.3 respectively.

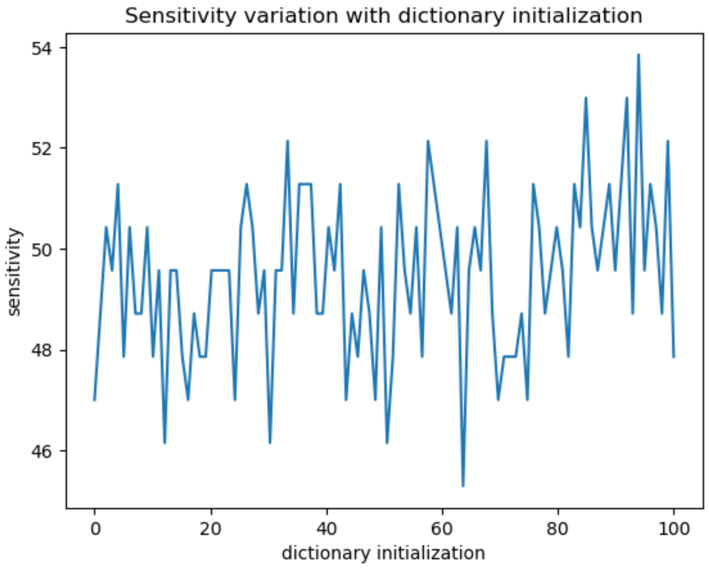


Figure 6.1: Variation of sensitivity, specificity and balanced accuracy with dictionary initialization for the ISIC dataset.

Through inspection of the figures it is clear that the dictionary initialization does introduce some significant variability in the final results. This variability is more apparent in the sensitivity with values in the range $[45.29, 53.85]$, while the specificity is in the range $[71.84, , 75.16]$, which translates into the balanced accuracy being in the range $[59.41, 63.57]$.

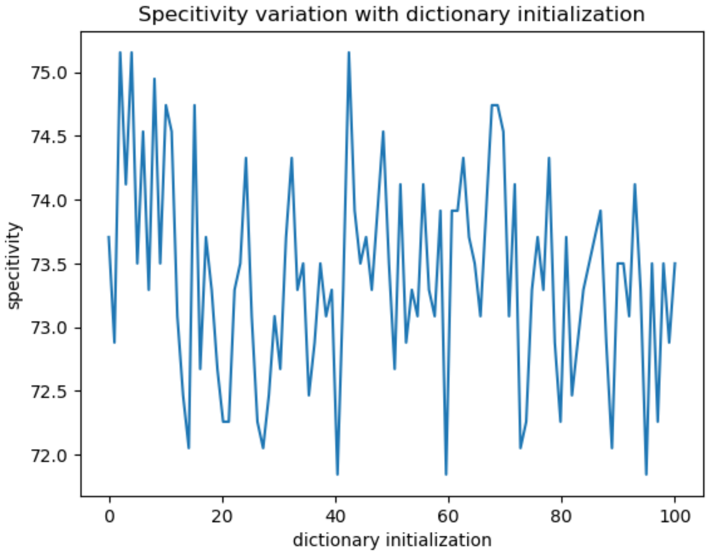


Figure 6.2: Variation of sensitivity, specificity and balanced accuracy with dictionary initialization for the ISIC dataset.

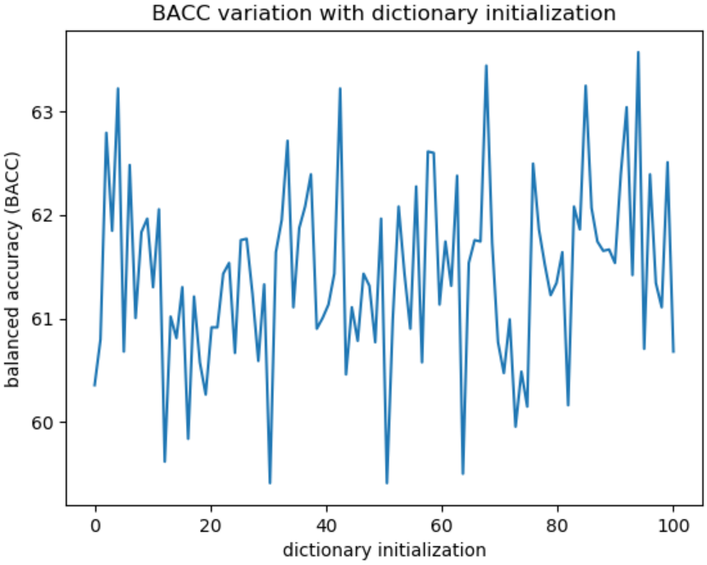


Figure 6.3: Variation of sensitivity, specificity and balanced accuracy with dictionary initialization for the ISIC dataset.

6.2.2 ISIC and EDRA

We have treated the ISIC and EDRA datasets as two separate datasets and reported results for all of the methods in both datasets. Here, and to ascertain the influence of the training data in the final result, the EDRA dataset is merged with the training and validation images of the ISIC dataset. This augmented set is then used to train the model described in section 5.2.4 and evaluated on the test set of the ISIC dataset. The results are presented in table 6.2.

Table 6.2: Proposed system performance on ISIC test set when trained with a mergure of the EDRA dataset and the ISIC training and validation sets.

Training Set	SE (%)	SP (%)	*BACC* (%)	ROC AUC
ISIC	$56,41$	$71,43$	$63,50$	$64,48$
Augmented set	$58,12$	$71,22$	$64,67$	$67,41$

The addition of the EDRA images to the training set resulted in an improvement of $1,17\%$ with respect to the same system trained only with the ISIC training and validation sets. This improvement also extended to the ROC AUC of the system, where it was more pronounced. Even though there was an improvement in the BACC, this value is still low, probably due to the EDRA images not being representative of the ISIC test set. This just goes to show the difficulty of the ISIC test set.

Chapter 7

Conclusion

This chapter discusses the achievements obtained in this thesis and proposes some future work that build upon what was done so far.

7.1 Achievements

This thesis focused on the analysis of several methods and algorithms based on sparse representations.

In Chapter 3, an initial baseline system was proposed to tackle the problem of melanoma classification. This system made use of handcrafted features such as color and gradient histograms represented by sparse coding using over-complete dictionaries. The baseline system is topped by a late fusion of support vector machines that performs the final classification.

In Chapter 4, the notion of discriminative dictionaries is introduced, as well as a couple of methods that make use of them. Their integration is the baseline system is also discussed. The chapter ends by introducing hierarchical clustering applied to the dictionary atoms with the objective of removing inter-class common atoms.

Chapter 5 explores the use of deep learning for the problem of melanoma detection. Transfer learning is made using a convolutional neural network pre-trained on the ImageNet dataset, namely the VGG19. Features are extracted from this network and are applied in the baseline system as yet another source of information for melanoma classification.

Two datasets were used, the ISIC and EDRA datasets. Promising results were achieved for both datasets, with different systems. The System presented in section 5.2.4 achieved a BACC of $63,50\%$ on the ISIC dataset and the system presented in Section 4.1 achieved a BACC of $76,83\%$ for the EDRA dataset.

7.2 Future work

The final system obtained, even though it achieved promising results, is quite simple compared to some state-of-the-art methods for image classification which mainly make use of convolutional neural networks. It would be interesting to test this system on other public image datasets to see how it performs in an area other than skin cancer detection. The application of hierarchical clustering to any dictionary could also be further researched, since it quite increased the capability of the system in such a difficult dataset as is the ISIC dataset. Its inclusion in more modern deep learning systems for feature pruning could also be considered and studied.

With the increasing complexity of deep learning models, with neural networks that need to learn millions of parameters, making its use not only of memory, but also time consuming, methods like sparse representations and clustering, which cuts a lot of redundancies and therefore reduces memory and boosts efficiency, seem to be a promising direction in the near future.

Bibliography

[1] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1409.1556`.

[2] American Cancer Institute. `https://www.cancer.org/cancer/skin-cancer.html`. [Online; accessed 01-April-2019].

[3] R. Conic, C. I. Cabrera, A. A. Khorana, and B. R. Gastman. Determination of the impact of melanoma surgical timing on survival using the national cancer database. 78, 10 2017.

[4] K. Korotkov and R. Garcia. Computerized analysis of pigmented skin lesions: A review. 56, 10 2012.

[5] S. Pathan, K. G. Prabhu, and P. C. Siddalingaswamy. Techniques and algorithms for computer aided diagnosis of pigmented skin lesions - a review. *Biomed. Signal Proc. and Control*, 39:237–262, 2018.

[6] A. Shain and B. C Bastian. From melanocytes to melanomas. *Nature reviews. Cancer*, 16, 04 2016. doi: 10.1038/nrc.2016.37.

[7] ISIC home page. `https://www.isic-archive.com/#!/topWithHeader/wideContentTop/main`, . [Online; accessed 06-April-2019].

[8] A. Steiner, H. Pehamberger, and K. Wolff. In vivo epiluminescence microscopy of pigmented skin lesions. ii. diagnosis of small pigmented skin lesions and early detection of malignant melanoma. *Journal of the American Academy of Dermatology*, 17(4):584 – 591, 1987. ISSN 0190-9622. doi: https://doi.org/10.1016/S0190-9622(87)70240-0. URL `http://www.sciencedirect.com/science/article/pii/S0190962287702400`.

[9] W. Stolz, A. Riemann, A. B. Cognetta, L. Pillet, W. Abmayer, D. Holzel, P. Bilek, F. Nachbar, M. Landthaler, and O. Braun-Falco. Abcd rule of dermatoscopy: A new practical method for early recognition of malignant melanoma. *European Journal of Dermatology*, 4:521–527, 01 1994.

[10] G. Argenziano, G. Fabbrocini, P. Carli, V. De Giorgi, E. Sammarco, and M. Delfino. Epiluminescence Microscopy for the Diagnosis of Doubtful Melanocytic Skin Lesions: Comparison of the ABCD Rule

of Dermatoscopy and a New 7-Point Checklist Based on Pattern Analysis. *Archives of Dermatology*, 134(12):1563–1570, 12 1998. ISSN 0003-987X. doi: 10.1001/archderm.134.12.1563. URL `https://doi.org/10.1001/archderm.134.12.1563`.

[11] M. E. Celebi and A. Zornberg. Automated quantification of clinically significant colors in dermoscopy images and its application to skin lesion classification. *IEEE Systems Journal*, 8(3):980–984, 2014. doi: 10.1109/JSYST.2014.2313671. URL `https://doi.org/10.1109/JSYST.2014.2313671`.

[12] R. Stanley, W. Stoecker, and R. Moss. A relative color approach to color discrimination for malignant melanoma detection in dermoscopy images. *Skin research and technology : official journal of International Society for Bioengineering and the Skin (ISBS) [and] International Society for Digital Imaging of Skin (ISDIS) [and] International Society for Skin Imaging (ISSI)*, 13:62–72, 03 2007. doi: 10.1111/j.1600-0846.2007.00192.x.

[13] M. E. Celebi, H. A. Kingravi, B. Uddin, H. Iyatomi, Y. A. Aslandogan, W. V. Stoecker, and R. H. Moss. A methodological approach to the classification of dermoscopy images. *Comp. Med. Imag. and Graph.*, 31(6):362–373, 2007. doi: 10.1016/j.compmedimag.2007.01.003. URL `https://doi.org/10.1016/j.compmedimag.2007.01.003`.

[14] H. Iyatomi, H. Oka, M. E. Celebi, M. Hashimoto, M. Hagiwara, M. Tanaka, and K. Ogawa. An improved internet-based melanoma screening system with dermatologist-like tumor area extraction algorithm. *Comp. Med. Imag. and Graph.*, 32(7):566–579, 2008. doi: 10.1016/j.compmedimag.2008.06.005. URL `https://doi.org/10.1016/j.compmedimag.2008.06.005`.

[15] M. Rastgoo, R. García, O. Morel, and F. Marzani. Automatic differentiation of melanoma from dysplastic nevi. *Comp. Med. Imag. and Graph.*, 43:44–52, 2015. doi: 10.1016/j.compmedimag.2015.02.011. URL `https://doi.org/10.1016/j.compmedimag.2015.02.011`.

[16] R. Erol, M. Bayraktar, S. Kockara, S. Kaya, and T. Halic. Texture based skin lesion abruptness quantification to detect malignancy. *BMC Bioinformatics*, 18(14):51–60, 2017. doi: 10.1186/s12859-017-1892-5. URL `https://doi.org/10.1186/s12859-017-1892-5`.

[17] H. Iyatomi, H. Oka, M. E. Celebi, M. Tanaka, and K. Ogawa. Parameterization of dermoscopic findings for the internet-based melanoma screening system. pages 189 – 193, 05 2007. ISBN 1-4244-0707-9. doi: 10.1109/CIISP.2007.369315.

[18] Q. Abbas, M. E. Celebi, and I. Fondón. Computer-aided pattern classification system for dermoscopy images. *Skin research and technology : official journal of International Society for Bioengineering and the Skin (ISBS) [and] International Society for Digital Imaging of Skin (ISDIS) [and] International Society for Skin Imaging (ISSI)*, 18:278–89, 08 2011. doi: 10.1111/j.1600-0846.2011.00562.x.

[19] K. Møllersen, M. Zortea, K. Hindberg, T. Schopf, S. Skrøvseth, and F. Godtliebsen. *Improved Skin Lesion Diagnostics for General Practice by Computer-Aided Diagnostics*, pages 247–292. 09 2015. doi: 10.1201/b19107-10.

[20] W. Stoecker, W. Weiling Li, and R. Moss. Automatic detection of asymmetry in skin tumors. *Computerized medical imaging and graphics : the official journal of the Computerized Medical Imaging Society*, 16:191–7, 05 1992. doi: 10.1016/0895-6111(92)90073-I.

[21] S. Seidenari, G. Pellacani, and C. Grana. Colors in atypical nevi: A computer description reproducing clinical assessment. *Skin research and technology : official journal of International Society for Bioengineering and the Skin (ISBS) [and] International Society for Digital Imaging of Skin (ISDIS) [and] International Society for Skin Imaging (ISSI)*, 11:36–41, 03 2005. doi: 10.1111/j.1600-0846.2005.00097.x.

[22] M. Zortea, T. R. Schopf, K. Thon, M. Geilhufe, K. Hindberg, H. M. Kirchesch, K. Møllersen, J. Schulz, S. O. Skrøvseth, and F. Godtliebsen. Performance of a dermoscopy-based computer vision system for the diagnosis of pigmented skin lesions compared with visual evaluation by experienced dermatologists. *Artificial Intelligence in Medicine*, 60(1):13–26, 2014. doi: 10.1016/j.artmed.2013.11.006. URL https://doi.org/10.1016/j.artmed.2013.11.006.

[23] C. Barata, M. Figueiredo, M. E. Celebi, and J. Marques. Local features applied to dermoscopy images: Bag-of-features versus sparse coding. pages 528–536, 05 2017.

[24] M. Rastgoo, G. Lemaitre, O. Morel, J. Massich, R. García, F. Mériaudeau, F. Marzani, and D. Sidibé. Classification of melanoma lesions using sparse coded features and random forests. In *Medical Imaging 2016: Computer-Aided Diagnosis, San Diego, California, United States, 27 February - 3 March 2016*, page 97850C, 2016. doi: 10.1117/12.2216973. URL https://doi.org/10.1117/12.2216973.

[25] K. Matsunaga, A. Hamada, A. Minagawa, and H. Koga. Image classification of melanoma, nevus and seborrheic keratosis by deep neural network ensemble. 03 2017.

[26] A. Menegola, M. Fornaciali, R. Pires, S. E. F. de Avila, and E. Valle. Towards automated melanoma screening: Exploring transfer learning schemes. *CoRR*, abs/1609.01228, 2016. URL http://arxiv.org/abs/1609.01228.

[27] V. Pomponiu, H. Nejati, and N. Cheung. Deepmole: Deep neural networks for skin mole lesion classification. In *2016 IEEE International Conference on Image Processing, ICIP 2016, Phoenix, AZ, USA, September 25-28, 2016*, pages 2623–2627, 2016. doi: 10.1109/ICIP.2016.7532834. URL https://doi.org/10.1109/ICIP.2016.7532834.

[28] M. Elad, M. A. T. Figueiredo, and Y. Ma. On the role of sparse and redundant representations in image processing. *Proceedings of the IEEE*, 98(6):972–982, 2010. doi: 10.1109/JPROC.2009.2037655. URL https://doi.org/10.1109/JPROC.2009.2037655.

[29] N. C. F. Codella, J. Cai, M. Abedini, R. Garnavi, A. Halpern, and J. R. Smith. Deep learning, sparse coding, and svm for melanoma recognition in dermoscopy images. In *MLMI*, 2015.

[30] C. Barata, M. E. Celebi, and J. Marques. A survey of feature extraction in dermoscopy image analysis of skin cancer. *IEEE Journal of Biomedical and Health Informatics*, PP:1–1, 06 2018. doi: 10.1109/JBHI.2018.2845939.

[31] Interactive atlas of dermoscopy. `http://dermoscopy.org/`. [Online; accessed 09-October-2018].

[32] H. Li and F. Liu. Image denoising via sparse and redundant representations over learned dictionaries in wavelet domain. In *Proceedings of the Fifth International Conference on Image and Graphics, ICIG 2009, Xi'an, Shanxi, China, 20-23 September 2009*, pages 754–758, 2009. doi: 10.1109/ICIG.2009.101. URL `https://doi.org/10.1109/ICIG.2009.101`.

[33] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Trans. Image Processing*, 17(1):53–69, 2008. doi: 10.1109/TIP.2007.911828. URL `https://doi.org/10.1109/TIP.2007.911828`.

[34] M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies. Sparse representations in audio and music: From coding to source separation. *Proceedings of the IEEE*, 98(6):995–1005, 2010. doi: 10.1109/JPROC.2009.2030345. URL `https://doi.org/10.1109/JPROC.2009.2030345`.

[35] R. Grosse, R. Raina, H. Kwong, and A. Y. Ng. Shift-invariant sparse coding for audio classification. 9, 06 2012.

[36] M. Wang, W. Xu, and A. Tang. On the performance of sparse recovery via l_p-minimization (0 <= p <= 1). *IEEE Trans. Information Theory*, 57(11):7255–7278, 2011. doi: 10.1109/TIT.2011.2159959. URL `https://doi.org/10.1109/TIT.2011.2159959`.

[37] D. Donoho and I. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81, 11 1998. doi: 10.1093/biomet/81.3.425.

[38] D. Donoho and I. M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90, 12 1999. doi: 10.1080/01621459.1995.10476626.

[39] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 689–696, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553463. URL `http://doi.acm.org/10.1145/1553374.1553463`.

[40] J. A. Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Trans. Information Theory*, 50(10):2231–2242, 2004. doi: 10.1109/TIT.2004.834793. URL `https://doi.org/10.1109/TIT.2004.834793`.

[41] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

[42] C. Barata, M. Ruela, M. Francisco, T. Mendonça, and J. S. Marques. Two systems for the detection of melanomas in dermoscopy images using texture and color features. *IEEE Systems Journal*, 8 (3):965–979, 2014. doi: 10.1109/JSYST.2013.2271540. URL `https://doi.org/10.1109/JSYST.2013.2271540`.

[43] C. Barata, M. E. Celebi, and J. S. Marques. Improving dermoscopy image classification using color constancy. *IEEE J. Biomedical and Health Informatics*, 19(3):1146–1152, 2015. doi: 10.1109/JBHI.2014.2336473. URL `https://doi.org/10.1109/JBHI.2014.2336473`.

[44] G. D. Finlayson and E. Trezzi. Shades of gray and colour constancy. In *The Twelfth Color Imaging Conference: Color Science and Engineering Systems, Technologies, Applications, CIC 2004, Scottsdale, Arizona, USA, November 9-12, 2004*, pages 37–41, 2004. URL `http://www.ingentaconnect.com/content/ist/cic/2004/00002004/00000001/art00008`.

[45] H. Ganster, A. Pinz, R. Röhrer, E. Wildling, M. Binder, and H. Kittler. Automated melanoma recognition. *IEEE Trans. Med. Imaging*, 20(3):233–239, 2001. doi: 10.1109/42.918473. URL `https://doi.org/10.1109/42.918473`.

[46] H. Iyatomi, H. Oka, M. E. Celebi, M. Hashimoto, M. Hagiwara, M. Tanaka, and K. Ogawa. An improved internet-based melanoma screening system with dermatologist-like tumor area extraction algorithm. *Computerized medical imaging and graphics : the official journal of the Computerized Medical Imaging Society*, 32:566–79, 11 2008. doi: 10.1016/j.compmedimag.2008.06.005.

[47] R. M. Haralick, K. S. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Trans. Systems, Man, and Cybernetics*, 3(6):610–621, 1973. doi: 10.1109/TSMC.1973.4309314. URL `https://doi.org/10.1109/TSMC.1973.4309314`.

[48] R. Azencott, J. Wang, and L. Younes. Texture classification using windowed fourier filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(2):148–153, 1997. doi: 10.1109/34.574796. URL `https://doi.org/10.1109/34.574796`.

[49] T. Randen and J. H. Husøy. Filtering for texture classification: A comparative study. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(4):291–310, 1999. doi: 10.1109/34.761261. URL `https://doi.org/10.1109/34.761261`.

[50] *Proceedings 1995 International Conference on Image Processing, Washington, DC, USA, October 23-26, 1995*, 1995. IEEE Computer Society. ISBN 0-8186-7310-9. URL `http://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=11606`.

[51] S. Arivazhagan, L. Ganesan, and S. P. Priyal. Texture classification using gabor wavelets based rotation invariant features. *Pattern Recognition Letters*, 27(16):1976–1982, 2006. doi: 10.1016/j.patrec.2006.05.008. URL `https://doi.org/10.1016/j.patrec.2006.05.008`.

[52] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 01 1995. doi: 10.1023/A:1022627411411.

[53] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[54] J. Mairal. SPAMS library. `http://spams-devel.gforge.inria.fr/`. [Online; accessed 05-March-2018].

[55] M. W. Marcellin, M. Gormish, A. Bilgin, and M. Boliek. Overview of jpeg-2000. *Data Compression Conference Proceedings*, 05 2000.

[56] S. Kong and D. Wang. A dictionary learning approach for classification: Separating the particularity and the commonality. In *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part I*, pages 186–199, 2012. doi: 10.1007/978-3-642-33718-5_14. URL `https://doi.org/10.1007/978-3-642-33718-5_14`.

[57] J. Mairal, F. R. Bach, and J. Ponce. Task-driven dictionary learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(4):791–804, 2012. doi: 10.1109/TPAMI.2011.156. URL `https://doi.org/10.1109/TPAMI.2011.156`.

[58] Q. Zhang and B. Li. Discriminative K-SVD for dictionary learning in face recognition. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 2691–2698, 2010. doi: 10.1109/CVPR.2010.5539989. URL `https://doi.org/10.1109/CVPR.2010.5539989`.

[59] O. Yim and K. Ack Baraly. Hierarchical cluster analysis: Comparison of three linkage measures and application to psychological data. *The Quantitative Methods for Psychology*, 11:8–21, 02 2015. doi: 10.20982/tqmp.11.1.p008.

[60] H. Finch. Comparison of distance measures in cluster analysis with dichotomous data. *Journal of Data Science*, 3:85–100, 01 2005.

[61] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: An astounding baseline for recognition. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 512–519, June 2014. doi: 10.1109/CVPRW.2014.131.

[62] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732, June 2014. doi: 10.1109/CVPR.2014.223.

[63] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu. Cnn-rnn: A unified framework for multi-label image classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[64] N. Codella, J. Cai, M. Abedini, R. Garnavi, A. Halpern, and J. R. Smith. Deep learning, sparse coding, and SVM for melanoma recognition in dermoscopy images. In *Machine Learning in Medical*

Imaging - 6th International Workshop, MLMI 2015, Held in Conjunction with MICCAI 2015, Munich, Germany, October 5, 2015, Proceedings, pages 118–126, 2015. doi: 10.1007/978-3-319-24888-2_15. URL https://doi.org/10.1007/978-3-319-24888-2_15.

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

[66] CS231n: Convolutional Neural Networks for Visual Recognition, class notes. http://cs231n.github.io/convolutional-networks/. [Online; accessed 06-April-2019].

[67] M. D. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, and G. E. Hinton. On rectified linear units for speech processing. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3517–3521, May 2013. doi: 10.1109/ICASSP.2013.6638312.

[68] G. E. Dahl, T. N. Sainath, and G. E. Hinton. Improving deep neural networks for lvcsr using rectified linear units and dropout. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8609–8613, May 2013. doi: 10.1109/ICASSP.2013.6639346.

[69] J. Nagi, F. Ducatelle, G. Di Caro, D. Ciresan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, and L. Maria Gambardella. Max-pooling convolutional neural networks for vision-based hand gesture recognition. pages 342–347, 11 2011. doi: 10.1109/ICSIPA.2011.6144164.

[70] Clarifai. https://www.clarifai.com/technology. [Online; accessed 09-April-2019].

[71] ImageNet website. http://www.image-net.org/, . [Online; accessed 03-April-2019].

[72] ISIC 2017 challenge homepage. https://challenge.kitware.com/#challenge/n/ISIC_2017%3A_Skin_Lesion_Analysis_Towards_Melanoma_Detection, . [Online; accessed 06-April-2019].

[73] How and When to Use ROC Curves and Precision-Recall Curves for Classification in Python. https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/, . [Online; accessed 06-April-2019].

[74] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Adv. Large Margin Classif.*, 10, 06 2000.

[75] Understanding AUC-ROC Curve. https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5, . [Online; accessed 16-April-2019].

[76] A. Menegola, J. Tavares, M. Fornaciali, L. T. Li, S. E. F. de Avila, and E. Valle. RECOD titans at ISIC challenge 2017. *CoRR*, abs/1703.04819, 2017. URL http://arxiv.org/abs/1703.04819.

[77] T. Devries and D. Ramachandram. Skin lesion classification using deep multi-scale convolutional neural networks. *CoRR*, abs/1703.01402, 2017. URL http://arxiv.org/abs/1703.01402.

[78] C. N. Vasconcelos and B. N. Vasconcelos. Increasing deep learning melanoma classification by classical and expert knowledge based image transforms. *CoRR*, abs/1702.07025, 2017. URL `http://arxiv.org/abs/1702.07025`.

[79] A. Galdran, A. Alvarez-Gila, M. I. Meyer, C. L. Saratxaga, T. Araujo, E. Garrote, G. Aresta, P. Costa, A. M. Mendonça, and A. J. C. Campilho. Data-driven color augmentation techniques for deep skin image analysis. *CoRR*, abs/1703.03702, 2017. URL `http://arxiv.org/abs/1703.03702`.

[80] Y. Li and L. Shen. Skin lesion analysis towards melanoma detection using deep learning network. *CoRR*, abs/1703.00577, 2017. URL `http://arxiv.org/abs/1703.00577`.

[81] M. Berseth. ISIC 2017 - skin lesion analysis towards melanoma detection. *CoRR*, abs/1703.00523, 2017. URL `http://arxiv.org/abs/1703.00523`.

[82] W. Zhang, L. Gao, and R. Liu. Using deep learning method for classification: A proposed algorithm for the ISIC 2017 skin lesion classification challenge. *CoRR*, abs/1703.02182, 2017. URL `http://arxiv.org/abs/1703.02182`.

[83] G. W. Jiji and P. J. D. Raj. An extensive technique to detect and analyze melanoma: A challenge at the international symposium on biomedical imaging (ISBI) 2017. *CoRR*, abs/1702.08717, 2017. URL `http://arxiv.org/abs/1702.08717`.

[84] S. Guo, Y. Luo, and Y. Song. Random forests and VGG-NET: an algorithm for the ISIC 2017 skin lesion classification challenge. *CoRR*, abs/1703.05148, 2017. URL `http://arxiv.org/abs/1703.05148`.

[85] T. Fletcher. Support vector machines explained. 01 2009.

[86] A. Ng. Cs229 lecture notes. 2018.

[87] ImageNet 2014 challenge homepage. `http://image-net.org/challenges/LSVRC/2014/`, . [Online; accessed 03-April-2019].

[88] Y. Zheng, C. Yang, and A. Merkulov. Breast cancer screening using convolutional neural network and follow-up digital mammography. page 4, 05 2018. doi: 10.1117/12.2304564.

Appendix A

Support Vector Machine

The *Support Vector Machine* (SVM) is a popular binary classifier and has been used in a wide range of problems including the detection of skin cancer. In its simplest version, it separates the data using a hyperplane decision boundary.

This appendix provides a brief explanation of SVMs and is based on [85],[86].

A.1 Linearly separable data

Let $(\mathbf{x_i}, y_i), i = 1, ..., N$ be a dataset where $\mathbf{x_i}$ is an input vector of dimension d, *i.e.*, $\mathbf{x_i} \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$ is the binary label associated to data point $\mathbf{x_i}$.

The simplest case for the application of the SVM algorithm is a two-class problem, where the data from two classes is linearly separable. This means that there is a hyperplane in input space $\mathbb{R}^d$ that separates the data from the two classes, defined by

$$\mathbf{w} \cdot \mathbf{x} + b = 0, \tag{A.1}$$

where **w** is the normal vector to the hyperplane and $\frac{b}{||\mathbf{w}||}$ is the perpendicular distance from the hyperplane to the origin. The linear separability property for a 2-dimensional problem can be viewed in figure A.1, where the red class and the blue class are linearly separable.

Another important concept in support vector machines is the concept of margin. Referring to figure A.1, d_1 and d_2 are the distances from H_1 and H_2 to the decision hyperplane, often referred to as margin hyperplanes. These margin hyperplanes, parallel to the hyperplane, contain the data points that are the closest to the decision hyperplane and these points are called the support vectors which must be found. If the hyperplane is placed such that it is equidistant from H_1 and H_2, then $d_1 + d_2$ is the margin of the classifier.

When the data points are linearly separable, as seen in figure A.1, there are an infinite number of hyperplanes that can separate the given data. it can be argued that the best hyperplane is the one with (*maximum margin*) [52]. The bigger the margin, the more reliable will be the classification, since there will be a bigger leeway in where new samples from each class can be while still being correctly

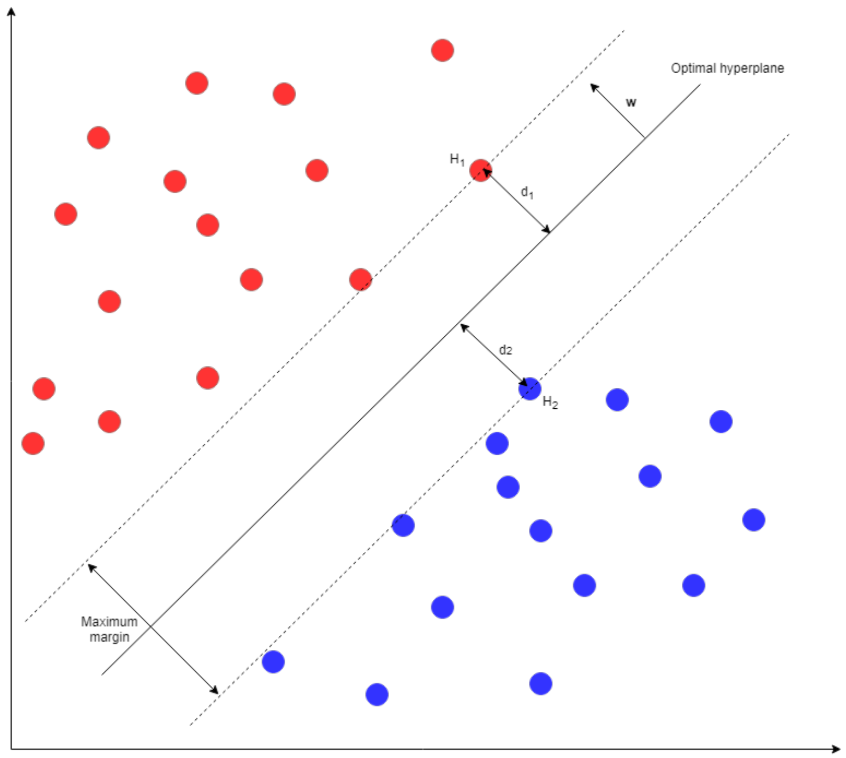


Figure A.1: Optimal hyperplane with linearly separable data in SVM algorithm

classified.

It can be proven that the margin m can be calculated by

$$m = \frac{2}{||\mathbf{w}||}, \tag{A.2}$$

where **w** is the vector orthogonal to the hyperplane [52]. Maximazing the margin then corresponds to minimizing the magnitude of vector **w**.

Ensuring that the data is linearly separated by a hyperplane defined by A.1 is the same has finding a **w** and b such that it is verified for each input $\mathbf{x_i}$

$$\begin{cases} \mathbf{w} \cdot \mathbf{x_i} + b \geq +1, & \text{for } y_i = +1 \\ \mathbf{w} \cdot \mathbf{x_i} + b \leq -1, & \text{for } y_i = -1 \end{cases}, \tag{A.3}$$

which can be combined into

$$y_i(\mathbf{w} \cdot \mathbf{x_i} + b) \geq 1. \tag{A.4}$$

Taking all this into account, the optimization problem to find the hyperplane (find **w** and b), such that it separates the data points and does so with maximum margin boils down to

$$\begin{aligned} \underset{\mathbf{w},b}{\text{minimize}} \quad & \tfrac{1}{2}||\mathbf{w}||^2 \qquad . \\ \text{subject to} \quad & y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \\ & \text{for } i = 1, ..., N, \end{aligned} \tag{A.5}$$

A.2 Non linearly separable data

Most classification problems are not linearly separated and cannot be solved with the technique described so far.

To classify non-linearly separable data using an hyperplane some data points should be allowed to be on the wrong side of the margin hyperplane. These points should be penalized in such a way that the classifier would make as few errors as possible while separating the majority of the data as intended. Since there are exceptions to A.4, this method is called soft margin. Slack variables $\zeta_i \geq 0$ are then introduced and a non-zero value for ζ_i assigns a cost penalization to sample x_i if the sample does not comply by the margin constraint.

The new optimization problem is the formulated as

$$\begin{aligned} \underset{\mathbf{w},b,\zeta\geq 0}{\text{minimize}} \quad & \tfrac{1}{2}||\mathbf{w}||^2 + C\textstyle\sum_i \zeta_i \\ \text{subject to} \quad & y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \zeta_i, \\ & \zeta_i \geq 0 \\ & \text{for } i = 1, ..., N, \end{aligned} \tag{A.6}$$

where C controls the penalties associated to data points on the wrong side of the margin hyperplanes.

A.3 Higher dimentions

Another way to deal with non linearly separable data is by mapping the input vectors $\mathbf{x}_i$ to a higher dimensional feature space *S* through a given transformation $\Phi(\mathbf{x})$, where $\Phi(\mathbf{x})$ is the feature mapping that transforms the input from $\mathbb{R}^d$ to $\mathbb{R}^{d'}$, where $d' > d$. A toy example is $\Phi(\mathbf{x}) = (x_1, x_2, x_1^2 + x_2^2)$, this transformation mapped the input $\mathbf{x}_i$ from $\mathbb{R}^2$ to $\mathbb{R}^3$.

The classification is then made by finding an hyperplane in the higher dimensional space *S* that corresponds to a flexible surface in the original dimensional space. This corresponds to using the same optimization problem for separable data but replacing its inner products between inputs $< x, x' >$ by the Kernel function $K(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}')$. A popular choice for the kernel is the radial basis function kernel (rbf)

$$K(\mathbf{x}, \mathbf{x}') = exp(-\gamma||\mathbf{x} - \mathbf{x}'||^2). \tag{A.7}$$

As stated, the radial basis function $K(\mathbf{x}, \mathbf{x}')$ defines a ”similarity” between the two training samples $\mathbf{x}$ and $\mathbf{x}'$ and is parametrized by the parameter γ that controls the width of the gaussian.

Appendix B

Evaluation Metrics

The problem at hand is the classification of dermoscopic images, and even though there are several different skin lesions, in this problem the focus is distinguishing between melanoma and non-melanoma skin lesions. It is therefore a binary classification problem and in such cases, and in some medical tests, two notions can be introduced: sensitivity and specificity.

Sensitivity (SE) measures true positives. Within the context of this problem, it shows the ratio between correctly classified melanoma skin lesions and the total number of melanoma skin lesions in the dataset. In a similar way, specificity (SP) measures the true negatives. That is, the ration between correctly classified non-melanoma skin lesions and the total number of non-melanoma skin lesion samples in the dataset that is being tested.

These two metrics describe how the system performs for each class, but it is also convenient to have a single value that measures the quality of the system. One way to obtain such a score would be to calculate the ratio between the number of samples correctly classified and the total number of samples. This however does not work in the presence of unbalanced data or when different types of error have different costs. For example, only 20% of the test dataset images correspond to melanoma skin lesions. Therefore if the classifier always said that the image is a non-melanoma skin lesion, it would still classify 80% of the samples correctly and one would get the impression that the system is good when it is definitely not the case.

The adopted measure is the average between sensitivity and specificity, which gives the average performance of the system on both classes. This metric is called balanced accuracy (BACC).

Appendix C

VGG 19

The convolutional neural network VGG19 was developed by Karen Simonyan and Andrew ZisserMAN [1], and was part of the model that came first and second in the localization and classification task, respectively, of the ImageNet 2104 challenge [87].

The VGG19 network is comprised of sixteen convolutional layers, arranged in 5 blocks and three fully connected layers in the end (total of 19 layers, hence the name), as can be seen in figure C.1.

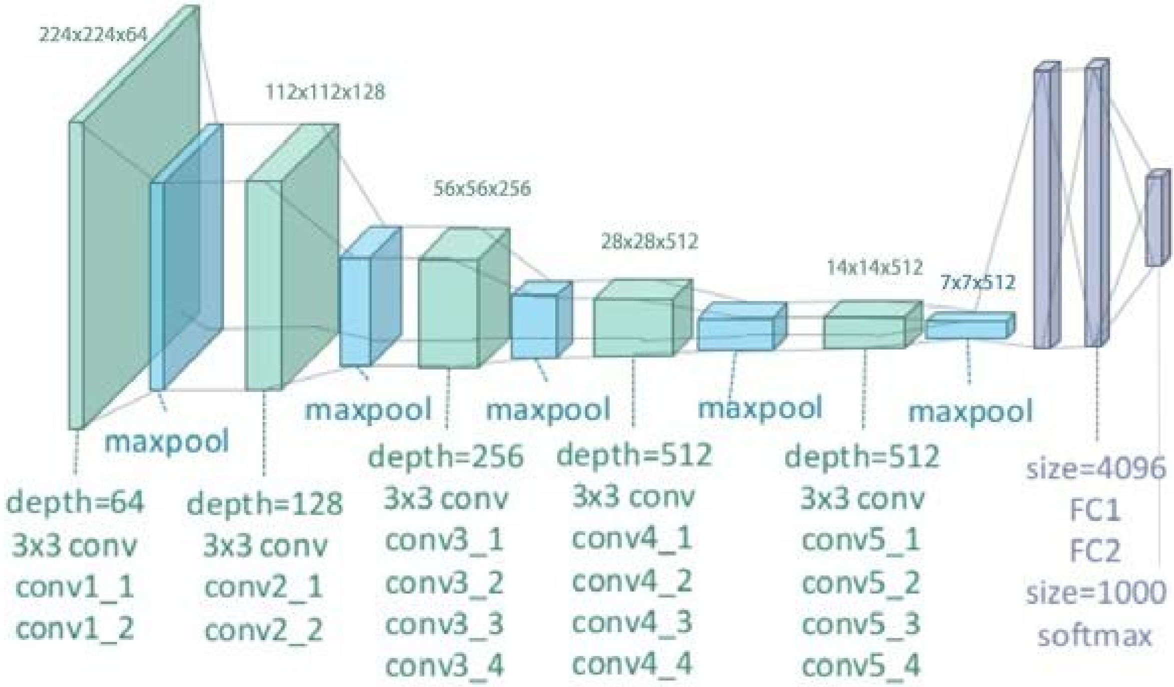


Figure C.1: VGG 19 architecture. Source [88]

C.1 Feature extraction

As explained, the VGG19 net is comprised of several blocks of convolutional layers, which will gradually reduce the size of the feature maps extracted by the convolution process. These feature maps are the deep features to be extracted, so what is left is to choose the layer to extract these features from and, depending on the layer, the features will have very different dimensions and levels of abstraction. The first few layers will perform convolution on inputs closely related to the original image, therefore the features will be very low-level, as they will be closely related to the structures in the image. However, this also means that the feature maps are very large, a feature vector from the second layer for example would have a size of $224 \times 224 = 50176$, which is very deterrent, given the memory limitations. The further one goes into the network, more high-level the features become, shortening in size in the process, allowing their use.

The features are then to be extracted from the fourth convolutional layer of the fifth block of layers. At the output of this layer are 512 activation maps of size 14×14, where each activation map will be a feature vector. This means that for every image, there are 512 feature vectors of size $14 \times 14 = 196$. These features will replace the hand-crafted and the systems previously described will be trained and tested with these features.

One final consideration that must be made is that the VGG19 net only accepts square images of size 224×224, therefore all dermoscopic images had to be resized to fit this. Since the great majority of the images are not square images, this means that their aspect ratio will be distorted, which may potentially hinder the results.