A Prototype System to Archive and Retrieve Histopathology Images by Content

Juan Carlos Caicedo Rueda

National University of Colombia
Engineering School
Master’s Thesis

A Prototype System to Archive and Retrieve Histopathology Images by Content

by

Juan Carlos Caicedo Rueda

Submitted to the Engineering School of the National University of Colombia, in partial fulfillment of the requirements for the degree of

Master of Science in Systems and Computer Engineering

Under the guidance of

Fabio A. Gonzalez and Eduardo Romero

Bogotá D.C., May 19, 2008
Dedication

To my parents Germán and Beatriz.
To my brothers Andres, Diana, Gabriel, Paula and Natalia.
For their endless love and support.
Contents

I Preliminaries 1

1 Introduction 2
1.1 Goals ................................................. 3
1.2 Main Contributions .................................. 4
1.3 Outline .............................................. 5

2 Problem Statement 7
2.1 Scope ............................................... 7
2.2 Previous work ....................................... 8
2.3 Histopathology Images ............................. 10

3 System Development 12
3.1 Image Retrieval Model ............................. 12
3.2 System Requirements ............................... 16
  3.2.1 Use cases and Functional Requirements .......... 16
  3.2.2 Non-Functional Requirements .................... 17
3.3 System Implementation ............................ 17
  3.3.1 Content-based Retrieval Module ................. 19
3.4 User Interface ...................................... 19

II System Effectiveness 22

4 Low-level Similarities for Histopathology Image Retrieval 23
  4.1 Low-level Feature Extraction ....................... 24
  4.2 Low-level Metrics ................................ 25
    4.2.1 Metric combination .......................... 27

5 A Semantic Similarity for Histopathology Image Retrieval 29
  5.1 Semantic Mapper Structure ....................... 30
    5.1.1 Individual concept identification ............ 30
    5.1.2 Learning algorithms .......................... 32
    5.1.3 Semantic feature vector .................... 33
    5.1.4 Model generalization ....................... 34
List of Figures

3.1 Overview of the image database system architecture . . . . . . . 13
3.2 Image retrieval framework . . . . . . . . . . . . . . . . . . . . . 14
3.3 Global system architecture . . . . . . . . . . . . . . . . . . . . . 18
3.4 Main menu of the user interface . . . . . . . . . . . . . . . . . 20
3.5 Screen shoots of the main user interface . . . . . . . . . . . . . . 21

4.1 An example basal-cell carcinoma image. . . . . . . . . . . . . . 24
4.2 Feature extraction results on three example images. . . . . . . 26
4.3 Multidimensional Scaling with the low-level metric . . . . . . . 28

5.1 The semantic mapping process. . . . . . . . . . . . . . . . . . . . 31
5.2 An example SVM hyperplane and the margin. . . . . . . . . . . 33
5.3 An example error rate vs complexity graph . . . . . . . . . . . . 36
5.4 Error rates using different models . . . . . . . . . . . . . . . . . 37
5.5 Multidimensional scaling with the semantic metric . . . . . . . 40

6.1 Precision vs. Recall graph for individual feature-metric pairs . 46
6.2 Precision vs. Recall for Morpheaform Pattern . . . . . . . . . . 47
6.3 Precision vs Recall graph of the linear combination approach . 48
6.4 Precision vs Recall graph using two similarities . . . . . . . . . 50

7.1 The triangle inequality . . . . . . . . . . . . . . . . . . . . . . . . 59

8.1 Sequential scan response times . . . . . . . . . . . . . . . . . . . 64
8.2 Response times using the low-level metric . . . . . . . . . . . . 65
8.3 Response times using the semantic metric . . . . . . . . . . . . 67
8.4 Average error in the ranking list . . . . . . . . . . . . . . . . . . 69

9.1 Use case diagram for the prototype system . . . . . . . . . . . . 76
9.2 Entity-Relationship diagram for the system core platform . . . . 76
List of Tables

2.1 Histopathology image concepts ................................. 10
3.1 Main use cases for the system development .................. 16
5.1 Final configuration of the SVM classifiers .................... 38
6.1 Effectiveness rank for each metric ............................ 45
6.2 Average precision values for the best feature-metric pairs .... 45
6.3 Query topics with the percentage of relevant images ........... 46
6.4 Performance measures for image retrieval .................... 49
6.5 Performance measures for image retrieval grouped by search topic 50
6.6 Comparative results per topic ................................ 52
6.7 ImageCLEFmed results ....................................... 54
8.1 Average response times ...................................... 65
8.2 Featured response times of the LAESA algorithm ............ 66
8.3 Featured response times of the LAESA algorithm using the semantic metric .................................. 67
Abstract

Nowadays, content-based medical image retrieval is an important research topic because of the large amounts of diagnostic images are daily produced. Those images may be useful as reference material in research, academic tasks and clinical environments. However, automated methods to access this information in an effective and efficient way are not fully available. Current hospital information systems are mainly based on a keyword search strategy, when users need to look for something. Nevertheless, annotations are sometimes unavailable or simply not enough to deal with the complexity of visual image contents. The aim of this research was to design, develop and evaluate a content-based image search system in a histopathology domain. Two main aspects have been studied to achieve a useful system: (1) how to effectively search relevant images, and (2) how to efficiently retrieve the relevant-identified results. Effectiveness means that the system is able to identify the most relevant images according to user information needs. This is accomplished using a semantic mapper that converts low-level features extracted from the image into its associated histopathology concepts. The semantic mapper was designed following a machine-learning approach to generate a semantic-metric space in which conceptual distances between images can be calculated. Efficiency deals with strategies to avoid large computational loads when the system is queried, in order to produce interactive responses. This system take advantage of the triangular inequality in the semantic metric space, following a branch and bound strategy to save the similarity calculation on irrelevant images. The experimental evaluation shows that the proposed model improves the effectiveness of the system from a precision of 67% in the first image retrieved with low-level features, to 80% using semantic features. In addition, the used indexing scheme increases the response time of the system in a factor of 10, compared with the sequential scan approach. Those results indicate that this system to retrieve histopathology images by content is effective and efficient.
Part I

Preliminaries
Chapter 1

Introduction

Medical images have been widely used to support clinical decisions in health centers for many years. The medical practice requires to evaluate patient’s health with reliable evidence to recommend effective treatments and medical images provide enough information about diseases’ state. Physicians evaluate medical images looking for patterns that characterize diseases and then make the pertinent recommendations. This practice is applied in hospitals many times per day, producing large volumes of medical images to be stored and managed [38]. However, once images have been used to diagnose, they are archived and rarely used again.

A large number of available medical images in health centers, hospitals and universities constitutes a knowledge base about diseases and diagnosis, useful for academic activities [49], research evaluations, and the decision making processes [10]. In academic environments, a collection of medical images is a useful asset to teach the recognition of patterns and pathologies. For research purposes, medical images support physicians reasoning, by comparing different files, finding correlations between cases and evaluating results of the applied treatments. In the common medical practice, an accessible knowledge base of previously-diagnosed cases may be useful to support the decision making process of new similar cases.

Medical practice is based on personal experience and on learning of others’ experience. In general, the opinion of an expert is better considered, when it is based on a large experiential evidence, of its own or of other experts [53]. In that sense, a large volume of medical images constitutes a knowledge base that collects the experience of many physicians, which have attended a wide diversity of patients. Consequently, effective access to this knowledge base may be a valuable tool to support the decision making process.

The management of clinical registries and digital images has been delegated to medical information systems. These systems support management processes, historical storage and patients’ follow up. The aim of these systems is often oriented to deliver the required information at the right time, in the right place, to the right persons to support quality and efficiency in the health-care practice.
However, current hospital information systems only provide functionalities to query for data using simple, clinical attributes, such as patient’s name or some dates, which generally are forgotten and do not allow identify valuable information for the decision making process.

Medical image databases and Picture Archiving and Communication Systems (PACS) are traditionally based on a search-by-keyword approach. Although medical registries are often composed of alphanumeric data, keywords may not be enough to find relevant images in large databases, mainly because keywords are not always available and images cannot be fully described by textual annotations. Medical images have specific visual patterns which indicate the presence or absence of a disease, with different shapes, intensities, colors and distributions.

Medical image databases may provide capabilities to search by image contents, taking advantage of visual properties. Research in that direction has evolved to Content Based Image Retrieval (CBIR) in different scenarios such as arts, advertising and films industry, among others. Comprehensive surveys on general purpose CBIR systems can be found in [25, 46, 54]. In the medical field a lot of work has also been done, mainly to model the semantics of clinical image analysis [26].

In general, a CBIR system is composed of three main modules: feature extraction, similarity functions and a retrieval algorithm. The feature extraction module is able to quantify some properties of image contents, using image processing techniques. Based on the recognized features a similarity function is defined to evaluate how close two images are. Given an example pattern, a retrieval algorithm is used to identify the set of most similar images using the similarity function. The example pattern is often an image that the user has at hand to query the system, a frequent scenario in medicine, when a physician is attending a new patient. This approach to query the system is known as query by example.

The design and development of effective and efficient CBIR systems is still a research problem, because of the nature of digital images involve two well known problems: the semantic gap and the computational load to manage large file collections. The semantic gap, as was defined by Smeulders et al. [46], deals with the lack of coincidence between the information that one can extract form the visual data and the interpretation that the same data have for a user in a given situation. It has linguistic and contextual consequences, and mainly depends on the domain knowledge to represent images. On the other hand, the computation load, when large image collections are managed, may make impractical the use of CBIR systems. Also, a CBIR system requires to design strategies to provide users with a search engine with interactive response [40].

1.1 Goals

The main goal of this research was to design a computational strategy to find and retrieve, from a repository, histopathology images using an example query
image. The following is the description of the general purposes of this research:

- To propose a computational strategy to calculate the similarity between two histopathology images, being as faithful as possible to the pathologist’s notion of similarity.
- To efficiently retrieve the subset of most similar images in the database.
- To evaluate the effectiveness and efficiency of the proposed model for content-based histopathology image retrieval. The effectiveness measures how good the system is for searching relevant images for the user. Efficiency deals with interaction and time responses.
- To design and implement a prototype system that allows users to present a query image in order to get as result, the most similar images in the database.

1.2 Main Contributions

This research presents an effective and efficient strategy for histopathology image retrieval. The following is the outline of the main contributions of this work:

- A base line for histopathology image retrieval has been identified using of low-level features. Images can be represented by simple visual features and it is needed to evaluate how good those features are to search histopathology images. In general, low-level features have no meaning to physicians, although a combination of them turns out to grossly match medical concepts. Results of such evaluation were published in [4, 6].

- The effective retrieval of histopathology images is accomplished by a computational strategy that compares images, based on machine learning techniques. The proposed strategy uses a semantic mapper to identify which histopathology concepts are present in an image, producing a semantic feature vector. The semantic mapper is a model based on machine learning algorithms that links low-level features to high-level concepts. To calculate the similarity between images a metric based on the Jacquard coefficient is used to emphasize the concepts shared by both images. This model has been published in [8].

- The efficient search, on the other hand, is achieved by the use of a metric indexing method, allowing fast retrieval while the collection is growing. The retrieval algorithm in this system uses a metric indexing method, based on the semantic similarity, allowing a retrieval mechanism faster than the sequential scanning approach. This approach has been published in [41].
A system architecture implementing the proposed image retrieval model has been developed. This system is based on a multi-tier architecture according to the J2EE standard. The image retrieval model and the system architecture have been published in [5, 9].

1.3 Outline

This document is organized in four main parts. Part I: Preliminaries, presents the general background and previous work in the research field. It also introduces the main aspects of the software development process. Part II: System Effectiveness, introduce the proposed approach to search images by content and presents the evaluation based on information retrieval measures. Part III: System Efficiency, analyzes the problem of fast image search based on similarity measures, and presents and evaluates the implementation of a metric indexing algorithm. Part IV: Summary, discusses the main aspects of this research and presents the conclusions and future work. In particular, each document Part is comprised of some specific Chapters that are organized as follows:

Part I: Preliminaries:

- Chapter 1 presents an introduction to the research problem, its goals and the main contributions made through this work. This Chapter also presents the main motivation and pertinence of this research.

- Chapter 2: Problem Statement, presents the general background and definition of the research problem. First, the scope of this work is defined and then, the main approaches in content-based image retrieval are presented and discussed. Finally, the specific problem of histopathology image retrieval is considered, and some key aspects about histopathology image contents are presented.

- Chapter 3: System development, is focused on the software development problem. A general model for image retrieval is presented to be able to configure the system with different search strategies. Then, the system requirements are briefly discussed to point the main aspects that have to be considered in the development work. The system implementation is described as well as the final user interface.

Part II: System Effectiveness:

- Chapter 4: Low-level Similarities for Histopathology Image Retrieval, describes all concerns about image contents analysis using low-level features. This Chapter presents a strategy to calculate the likeness of two images, based on visual appearance.

- Chapter 5: A Semantic Similarity for Histopathology Image Retrieval, presents the proposed strategy to calculate the similarity between two histopathology images based on conceptual interpretations. The proposed
model, based on machine learning techniques, links simple visual features with histopathology concepts, and defines a metric between semantic features.

- Chapter 6: *Effectiveness Evaluation*, describes a framework to evaluate the proposed image retrieval system based on results quality. This Chapter details the experimental setup, based on information retrieval measures, and discusses the results obtained with both the low-level and the semantic similarity.

Part III: *System Efficiency*:

- Chapter 7: *Search Algorithms*, discusses the problem of image search using a similarity measure. The main approach in experimental systems is to use the sequential scan algorithm that is computationally expensive in large databases with a large number of users. The implementation of an indexing method is presented, to speed up the system response when users are searching for similar images.

- Chapter 8: *Efficiency Evaluation*, presents experiments to evaluate the system performance based on response times. This evaluation was followed using different similarity functions to evaluate the performance improvement when the system is using an index for search. The trade-off between system effectiveness and efficiency is evaluated in this Chapter.

Part IV: Summary:

- Chapter 9: *Conclusions and Future Work*, presents a general discussion about this research, followed by the main conclusions and the future work in this area.
Chapter 2

Problem Statement

The problem of accessing images by content has been studied in a wide range of applications. The Photobook and QBIC was the first system that used visual properties to search similar images. The main idea behind a CBIR system is to define a set of representative features describing image contents and then, to use those features in order to calculate similarity between images. This chapter describes some previous approaches to medical image retrieval and defines the main features of the histopathology image retrieval problem.

2.1 Scope

This research studied the problem of Content-Based Medical Image Retrieval, specifically the problem of accessing histopathology images by content. The main goal of this research is to effectively search relevant images in a large collection of histopathology images. In addition, the problems of software development and efficient access have been approached in this research.

Two main features have been investigated in the proposed system: effectiveness and efficiency. Effectiveness deals with the problem of retrieving useful results for the user, according to its information needs. Efficiency deals with interactive responses, defined as getting results with short delays, generally lesser than 1 second. Both system features depends on the information volume of the image collection, that may be very large in real scenarios. In this research, an experimental collection of 1,502 images has been used, which is a representative sample of histopathology image collections.

The proposed system is a prototype that includes a user interface. Several iterations of software engineering were made to achieve the final prototype, mainly to extend the content-based methods. Although the final system is a prototype, it provides some final user tools to manage image collections on different applications, not only medical tasks. In addition, some image processing algorithms and retrieval methods are enabled to be used in general purpose images and photos, so that, software libraries and source code are reusable for
future works in this research area.

The intended uses of this histopathology CBIR system are teaching, research and decision making support. In teaching, students can query for similar images to realize concepts. Researches can require images with some particular patterns to support their reasoning. In the clinical practice, histopathology images are directly evaluated under the microscopy and when there exists a biological structure difficult to recognize, the pathologist can take a digital image and send it to the retrieval system. Query results in this system are images previously evaluated by other experts and have some annotations and textual descriptions, supporting the user reasoning.

Although the system has been devised for real use in a pathology laboratory, this research does not evaluates the clinical or academic impact on those environments. The main contribution of this research is the computational strategy to calculate the conceptual similarity between histopathology images, including a study of metric indexing techniques. This work has been mainly focused on the evaluation and validation of the computational methods according to the histopathology information needs.

2.2 Previous work

Early works in medical image retrieval were carried out by Petrakis et. al. [42, 44] with the development of the IDB, an image database system specially designed to calculate image similarity by spatial relationships. Images in IDB are segmented by the use of conventional algorithms and then, segments are edited by users to match semantic regions. The ASSERT system [48] is one of the best-known medical CBIR systems. Its approach involves the physician to select pathology bearing regions and anatomical landmarks when HRCT\(^1\) images are entered into the database, allowing accurate retrieval. These systems require manual description of image contents with hand-drawn regions, an impractical task when dealing with large databases.

Modeling visual-discriminative features based on low-level descriptors have been proposed, when dealing with large collections of heterogeneous types of medical images. Heterogeneous image modalities are common in real environments, so that features like texture, locality and color may help to retrieve similar images [24]. The FIRE system [15] brings together many histogram features, such as color and texture, that can be linearly combined for a particular metric. For specific image collections feature combinations may be adjusted to obtain reasonable results, but in general it is not expected to get good results, because there is no direct link between high-level interpretations and low-level features [28].

Another approach is the automatic spatial analysis based on the assumption that similar images have objects in corresponding positions. For instance, CT\(^2\)

\(^1\)High Resolution Computer Tomography
\(^2\)Computer Tomography
and MRI\textsuperscript{3} image representation with attributed relational graphs \cite{43}, allows a visual similarity analysis that includes spatial information such as nested objects and relative positions. A semantic net is introduced in \cite{20} to analyze spatial relationships in CT, MRI and x-ray images, together with a wavelet-based matching technique, robust to motion, noise and brightness. Nevertheless, image semantic sometimes is not codified in its spatial distribution but in concepts associated to those spatially located regions.

Shape analysis is also useful for some kind of medical images as proposed in \cite{45}. This kind of analysis requires previous segmentation and identification of objects, since it is a domain dependant task. For example, the automatic object recognition made by Thies et al. \cite{52} in carpal bone images, applies a formulation of a-priory knowledge about fingers size and topology. This approach has demonstrated to be accurate and it also allows individual object labeling and retrieval, although for some medical tasks robust techniques to identify objects have not been developed yet.

Since medical images can be acquired using different techniques (x-rays, magnetic resonance, microscopy, etc) and with different goals, Lehmann et al. \cite{27} proposed a code for unique classification of medical images, the so called IRMA code. This code is a complete and detailed reference of image categories, organized in a hierarchical form in order to support semantic queries on a database. Researchers are free to design models to identify such categories as a previous step for image retrieval. The classification of images has been studied by Gild et al. \cite{21} using nearest-neighbor classifiers to evaluate global features for medical image classification in 50 to 80 predefined categories.

Semantic analysis has been introduced with the use of models to interpret image contents like medical experts do. For example, a CBIR system for a collection of x-ray cervical spine images, performs a nine-point morphometry on vertebrae \cite{29}. This model captures a semantic knowledge of cervical spine images since distance between points are meaningful. Fuzzy methods were used by Barb and Chi-Ren \cite{2} on CT images, in which feature measurements are mapped to semantic terms by the definition of linguistic variables and semantic profiles. Our approach is most similar to the I-Browse system for histological images \cite{51}, which uses a semantic analyzer to operate on image sub-blocks. However, this model is quite complex because of the number of steps to decide about image semantics.

Efficient indexing of medical images is also an important issue in which few systems actually work. The most common approach is multidimensional indexing based on feature vectors like the ASSERT system used on region attributes. In \cite{45} shapes are indexed in a KD-tree to facilitate the efficient retrieval of similarly shaped curves. Indexing is used to avoid the sequential scanning for identifying similar images, because in real large image collections it is computationally expensive and not really usable.

\textsuperscript{3}Magnetic Resonance Image
2.3 Histopathology Images

Images in this work have been used to diagnose a special skin cancer known as basal-cell carcinoma. Basal-cell carcinoma is the most common skin disease in white populations and its incidence is growing worldwide [56]. It has different risk factors and its development is mainly due to ultraviolet radiation exposure. Pathologists confirm whether or not the disease is present after a biopsied tissue is evaluated under microscope. In this evaluation, physicians aim to recognize some characteristic patterns in tissues to determine whether the carcinoma is present or absent. This process is called differential diagnosis and it is mainly achieved by a visual analysis to understand cell structures. In [18], the structural patterns that characterize the basal-cell carcinoma are described.

<table>
<thead>
<tr>
<th>Concept</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pilosebaceous annexa</td>
<td>145</td>
</tr>
<tr>
<td>Cystic change</td>
<td>67</td>
</tr>
<tr>
<td>Elastosis</td>
<td>125</td>
</tr>
<tr>
<td>Eccrine glands</td>
<td>148</td>
</tr>
<tr>
<td>Lymphocyte infiltrate</td>
<td>140</td>
</tr>
<tr>
<td>Perineural invasion</td>
<td>6</td>
</tr>
<tr>
<td>Lesion with fibrosis</td>
<td>90</td>
</tr>
<tr>
<td>Micronodules</td>
<td>41</td>
</tr>
<tr>
<td>Necrosis</td>
<td>32</td>
</tr>
<tr>
<td>N-P-C, elastosis</td>
<td>52</td>
</tr>
<tr>
<td>N-P-C, fibrosis</td>
<td>50</td>
</tr>
<tr>
<td>N-P-C, infiltrate</td>
<td>176</td>
</tr>
<tr>
<td>N-P-C, pilosebaceous annexa</td>
<td>38</td>
</tr>
<tr>
<td>N-P-C, trabeculae</td>
<td>14</td>
</tr>
<tr>
<td>Morpheaform pattern</td>
<td>37</td>
</tr>
<tr>
<td>Rod trabeculae</td>
<td>60</td>
</tr>
<tr>
<td>Ulceration</td>
<td>15</td>
</tr>
<tr>
<td>Sanguineous vessel</td>
<td>122</td>
</tr>
</tbody>
</table>

The whole histopathology collection is composed of 5,995 images at $1,280 \times 1,024$ pixels, acquired under a Nikon microscope at the Pathology Lab. In the acquiring process a set of clinical cases was selected, related to different patients. Each patient’s sample is put under the microscopy and after a visual inspection, some images are captured at different zoom levels. Images were stored in a common directory in JPG format for their later analysis.

A subset of 1,502 images was studied and annotated by a pathologist to
describe its contents. The annotation process included the following steps:

1. The individual analysis of each image in the subset, to determine what kind of content they have. This analysis was done by an expert pathologist.

2. The enumeration of the main visual patterns present in images. This step produced a list of 30 visual structures, associated to tissue and cell properties.

3. The revision of the pattern list was done to study the relationships between them. Many patterns do not appear alone, instead, they are present together with other patterns building understandable configurations for physicians.

4. The definition of visual configurations with the most semantic relevance in histopathology was achieved, based on the 30-pattern list. This led to a list of 18 main concepts associated to basal-cell carcinoma images.

As a result, the later process gives a dataset with images and descriptions of their related concepts. One image may contain several concepts, and the annotations are useful to automatically validate if search results are relevant to the user information needs. Through this work, all the experimentation is based on the concept list with 18 entries, which is detailed in Table 2.1 with the number of examples in the collection. The first list of 30 visual structures was used only as reference to understand the domain knowledge and to interpret images.

An example of a histopathology concept is N-P-C, reported in Table 2.1. It is a convention for Nodule, Palisading cells and Cleft (N-P-C), which is a typical sign of basal cell carcinoma, not by the presence of any of them individually but by the visualization of all three visual structures together. Those annotations associated to images are the pathologist viewpoint about image contents and hence the best way to assess the system effectiveness.
Chapter 3

System Development

This work proposes a prototype system to archive and retrieve medical images by content. Many technologies are currently available to store medical images, but in general, they are focused to the workflow management and business operation, allowing to get reports, orders, schedules and patients data.

The proposed system is an image search engine to allow users taking full advantage of the stored image collection. This system may be helpful in the decision making process in clinical, academic and research areas. Figure 3.1 illustrates a general overview of the required system which provides support to image storing after its capture. Once images are uploaded to the system, they are analyzed in a feature extraction process and archived with the corresponding annotations and features. This storage is later used to resolve similarity queries and to deliver relevant results to the user.

The general principles that guide the system development are presented in this Chapter. Starting with important concepts of image retrieval in order to design a system with similarity search capabilities; following with technical and architectural requirements; and ending with the implemented system and the user interface.

3.1 Image Retrieval Model

Since the general goal of the system is allowing similarity searches, the first goal here is to define how images have to be compared. Figure 3.2 shows the general model used in this system: image database $I$ contains all histopathology samples and those images are mapped to some feature spaces. Each feature space requires a metric to allow a similarity evaluation according to the meaning of such space. For example, if one space represents color features, a metric between color properties is required for this specific feature space, in such way that images with similar colors get the highest similarity scores when are evaluated with the metric.

Formally, the feature extraction process is defined as: Let $I$ be the image
Figure 3.1: Overview of the image database system architecture

Overview of the image database system architecture. This system interacts with a medical environment in two ways, one represented by blue arrows and the other represented by red arrows. Blue arrows show the data flow when images are acquired and stored, archiving textual data, digital images and extracted features. Red arrows describe a query by example process, in which image features are compared to identify the most similar results in order to support clinical decisions.
Elements in the original images space $I$ are mapped to a feature space $F_k$ through a feature extraction process $E_k$. In those spaces, images are represented by particular features that can be compared by the use of a distance function $D_k$. 

Figure 3.2: Image retrieval framework
collection in the database. Let $F_k$ be the space defined for a feature $k$. The extraction process for a feature $k$ is defined as a function between $I$ and $F_k$:

$$E_k : I \rightarrow F_k \quad (3.1)$$

There exists a feature space onto which images are mapped when a specific feature is extracted, so that all images are now represented by their corresponding features in that space. In addition, many feature spaces have to be supported by the image database system and different measurement strategies must be defined for each space. If we assume that those feature spaces are metric spaces, distance functions can be devised for determining the similarity degree of images in each of such metric spaces. A metric space is a tuple $(F_k, D_k)$, where $F_k$ is a set and $D_k$ a metric such that:

1. $\forall x, y \in F_k: \quad D_k(x, y) \geq 0$. Non-negativity
2. $D_k(x, y) = 0$, if and only if $x = y$. Identity
3. $\forall x, y \in F_k: \quad D_k(x, y) = D_k(y, x)$. Symmetry
4. $\forall x, y, z \in F_k: \quad D_k(x, z) \leq D_k(x, y) + D_k(y, z)$. Triangle inequality

A metric $D_k$\footnote{Through this work, the term similarity and metric will be used indistinctly to talk about a function that allows to compare two objects. However, it is important to point out that they are different, as is discussed later in Chapter 7.} permits to introduce an order relationship between images according to the image representations in the $k$-th feature space. For example, we can select any point in a color representation space, let’s say the point representing the image with most red regions. Using the metric to evaluate color distances we can sort by closeness all other points in the same space. In such way, we can say that the first point in the sorted list represents the most similar image to the selected one, that may be another image with similar red regions. The last point in the list represents the most different image, that may be the image with no red regions.

Previous definitions allow to perform image comparisons using one feature and one metric. However, better comparisons may be achieved using many features and a combination of different metrics, as follows: Let $x, y \in I$ be images. Let $E_k$ be the feature extraction function of a feature $k$ and $D_k$ be a metric in the feature space $F_k$. A similarity function for different features is defined as the linear combination of metrics $D_k$ with importance factors $w_k$:

$$s(x, y) = \sum_k w_k D_k(E_k(x), E_k(y)) \quad (3.2)$$

Importance factors $w_k$ in Equation 3.2, are determined for the specific application, in this case, the histopathology image similarity. An exhaustive search approach has been followed to determine the best values for each $w_k$ for this application.
Table 3.1: Main use cases for the system development

<table>
<thead>
<tr>
<th>Actors - Role</th>
<th>Use cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrator - Maintenance</td>
<td>1. Create, delete and update users</td>
</tr>
<tr>
<td></td>
<td>2. Configure image collections and permissions</td>
</tr>
<tr>
<td>User - Explore collections</td>
<td>1. Search metadata</td>
</tr>
<tr>
<td></td>
<td>2. Visualize images</td>
</tr>
<tr>
<td></td>
<td>3. Export images and metadata</td>
</tr>
<tr>
<td>User - Upload</td>
<td>1. Select image sources</td>
</tr>
<tr>
<td></td>
<td>2. Configure image collection</td>
</tr>
<tr>
<td></td>
<td>3. Upload images</td>
</tr>
<tr>
<td>User - Query by example</td>
<td>1. Pick or upload image example</td>
</tr>
<tr>
<td></td>
<td>2. Visualize similarity results</td>
</tr>
</tbody>
</table>

Equations 3.1 and 3.2 define an important and practical requirement for software development: the system must allow the implementation of many features and many metrics, including a mechanism to configure some feature-metric pairs and the combination of different features in a single metric.

3.2 System Requirements

Design of such an image database system must consider extensibility and scalability. Extensibility through a modular architecture that incorporates new functionalities and capabilities. Scalability allowing new resources to be naturally plugged to the system in order to extend the service, covering more users or more data [30]. Two kind of requirements have been defined: functional requirements and technical requirements as is described in this section.

3.2.1 Use cases and Functional Requirements

Use cases capture the intended behavior of the system. They are expressed as services, tasks or functions for some users that interact with the system. In order to implement a first system prototype, the requirements included two user types: administrator and user. Both, have different roles to access and manage images in the system.

Table 3.1 shows the most representative use cases for the prototype system. As it was mentioned before, two actors interact with the system: the administrator with a maintenance role and the final user who explores, uploads and requires similar images. A more detailed view of use cases is shown in the use case diagram in Figure 9.1 of the Appendix.
3.2.2 Non-Functional Requirements

The system must be capable of managing any type of digital image format, allowing physicians to flexibly acquire and share images. In pathology, a digital camera coupled to the microscope allows users to capture images in TIFF, JPEG or BMP format. However, some hospitals store digital images using the well known standard format for medicine: DICOM.

Importantly, the image database system should not be designed for personal use as a desktop software. Main core system functionalities must be executed in a high-performance computing platform, allowing concurrent users through network services. In addition, images and medical data cannot be exposed to unauthorized access so that the system must provide an authentication security module.

The system has to be devised for allowing access through the web [12], but keeping in mind that the main use of a medical image database is to support diagnostic, academic and research activities. In addition, search of relevant images should be an easy process, that is to say, the system must provide content-based retrieval mechanisms such as query by example and relevance feedback.

3.3 System Implementation

The proposed system architecture is based on the Java 2 Enterprise Edition (J2EE) [1], that provides a multitiered and a distributed application model. In this model, each tier (also called layer) has a particular function in order to guarantee easy maintenance, extensibility and low coupling. Each tier offers services to other tiers, e.g. the Persistent tier provides support to retrieve and store results of the Business tier as well as the Business tier processes information fed to the Web tier. J2EE also supports transactions, web-services and provides a well defined security scheme to control the access from local and remote clients through a role-based security model for Enterprise Java Beans (EJB) and web components.

System architecture is composed of five main layers, a strategy which allows dividing processing responsibilities, data management and verification. The global model of the architecture can be viewed in Figure 3.3. As it was said before, each tier has a particular function and the loose interaction between the tiers results in a system with the whole functionality. Each layer is hereafter described:

1. **Client Tier**: It contains the Graphical User Interface (GUI) which allows interaction with the system and visualization of images and data. This tier has two client types: the web client whom uses the Internet browser and applets to access the system through the Web tier; and the standalone client that can be a remote application using RMI\(^2\) through JNDI\(^3\) to access the J2EE server.

---

\(^2\)Remote Method Invocation  
\(^3\)Java Naming and Directory Interface
Figure 3.3: Global system architecture

Global system architecture based on 5 main tiers with specific responsibilities. The persistent tier provides support to physical storage for images and data. The business tier contains modules for the main system operation. The presentation tier uses JSP and Servlets to format data. Client tier is composed of user interfaces that interacts with the presentation tier or directly with the business tier. A tier common to other tiers is the responsible of security, based on application server extensions.

2. **Web Tier.** It has Java Server Pages (JSP) and servlets. This tier processes requests and builds responses based on the Business tier results. This layer uses a local interface to invoke logical methods.

3. **Business Tier.** This is the system core, actually composed by Enterprise Java Beans (EJB) and plain Java classes. There are two types of EJBs: the Session Beans and the Entity Beans. Session Beans are objects representing clients in the server side, which can access logic methods such as: image archiving, image group configuration and image search by attributes, among others. Entity Beans constitute a database abstraction to execute SQL queries and to access relational databases with Java classes. This tier also has the Metadata Manager module to record images and extract textual information such as name, size and DICOM attributes; and the CBIR module, responsible for the feature extraction and similarity operations.

4. **Persistent Tier.** It provides tools to access the file system where images and their thumbnails are stored as well as the database that contains the metadata information and image features. Figure 9.2, in the Appendix, shows the general database design used to support the system configuration.

5. **Security Tier:** It provides access control to the application based on the role provided by the JBossSX Security extension, configured in a standard
XML descriptor. The security scheme verifies the user role and allows or refuses the access to some methods or domains.

The business tier in this architecture contains the content-based image retrieval module written mainly in the Java programming language. This module has an extensible architecture that allows the development and implementation of new feature extraction methods and similarity functions. This module property has been achieved using design patterns such as factories and composites, among others, to easily extend the system functionality through the best software practices. This module is in fact an image retrieval library with reusable algorithms and procedures through this system and other research and developments.

3.3.1 Content-based Retrieval Module

One of the main concerns of the proposed architecture is the CBIR module, which is located in the Business tier. The design of this module is based on design patterns which support extension by adding new feature extraction methods and new similarity functions. The CBIR module uses a standardized representation of digital images to apply algorithms independent of the format details. Image features are coded and stored in the Persistent layer to build a content-based index.

The CBIR module has four main submodules: feature representation, feature extraction algorithms, similarity functions, and retrieval algorithms. The feature representation submodule provides a class hierarchy with a common interface to access features data, and some specializations to handle specific features such as histograms, vectors, and trees. The feature extraction submodule uses the Template pattern to codify each feature extraction algorithm in a class method, and associate it to a class in the feature hierarchy. The similarity function submodule uses a hierarchy of metrics with the Command pattern, allowing an abstract configuration of the parameters to be compared to return the similarity score. The retrieval algorithm submodule, provides a framework to configure different retrieval algorithms with multiple feature-metric pairs and their associated importance factor, with the design of the Observer and Iterator patterns. Every submodule also includes a Factory pattern to dynamically create objects of its hierarchy.

With this structure, it is easy to develop new algorithms for feature extraction and similarity evaluation, making the module reusable and extensible. Currently, implemented features include the histogram features described in Chapter 4 and 5, as well as the similarity functions.

3.4 User Interface

The implemented service is a web-oriented platform, with a user interface accessible through a standard Internet browser. This section shows the main features of the user interface, which was enabled with the query by example paradigm to perform similarity searches.
Figure 3.4 shows the main page of the system when an administrator user is logged in, that provides the following features:

- **Left menu**: allow access to some tools to manage the system, such as to add and to drop users, to view and to edit personal information and to change the access password. This menu also has a section to upload images in a single or multiple way.

- **Right menu**: provides links to the local network.

- **Main menu**: enables the user to start searching and exploring collections in the system. It has two types of search options: Basic search, allowing the user to write some keywords to find images; and Content-based search, allowing the user to upload an example image and find similar images or view 30 random images to start a similarity exploration.

- **Top menu**: it has two options: Home, to return to the main menu and Logout, to quit from the system.

When the user starts to upload images, the system has an Applet to pick a directory and recursively upload all images inside. This process may take several minutes depending on the number of images and the size of each file. If the user has only one image to upload, the system has an alternative interface to pick the image directly and upload it.

Once the user has performed a query using the basic search or content-based search, it gets as result a set of relevant images in a grid. The grid allows to view...
Figure 3.5: Screen shoots of the main user interface

Screen shoots of the main user interface and some similarity search results

thumbnails of image results and, if the user wants to view details about one of them, it provides a link to get this information. If the user clicks in a thumbnail, a content-based search is launched using as example query the clicked image. Again, the results will appear in the grid view as is shown in Figure 3.5.

The user interface allows to download images in formats that are different to the original one. For example if the user uploads an image in DICOM format, it provides an option to download this image in JPG, PNG or BMP format. In addition, if this image has some associated data, such as patient name and appointment date, among others, the system allows to download a report-style PDF file with the image and the associated data.
Part II

System Effectiveness
Chapter 4

Low-level Similarities for Histopathology Image Retrieval

System effectiveness depends on the designed strategy to evaluate image similarity. If the image similarity strategy is poor, the user will get many non-relevant images as result and will need to spend many time searching for useful images. On the other hand, if the similarity strategy is good, the system will offer good results to users, that is to say the system is effective searching for relevant images. Section 3.1 has been presented the main concepts to take into account to build up an image retrieval system. In this Part, the feature extraction and similarity functions are presented in detail and are also evaluated.

In particular, the first steps towards an effective image retrieval strategy is modeling and evaluating low-level features. This Chapter presents the methods for histopathology image retrieval based on low-level features that are mainly visual information.

The aim of low-level feature modeling is mainly represent images with the most relevant visual data. So that, a basic analysis on the kind of content in histopathology images is required. Histologically, basal-cell carcinoma has numerous growth patterns, highlighted with few colors by staining techniques. In terms of image processing, histological images are distinguished by having more or less homogeneous textures or repeated patterns, which may be used to characterize the image. Experts determined that the shape of patterns is not a relevant property, because patterns may appear in regions of any size and any position. Depending on the type of pattern that the user is searching for, criteria such as tissues’ coloration, cells’ disposition, and regions’ contrast are important. Main information in histological images stands on repeated patterns of textures, with particular edges and slight color differences. Figure 4.1 shows a typical basal-cell carcinoma image in which the mentioned patterns are observed, i.e. a repeated texture with particular edges and color.
4.1 Low-level Feature Extraction

A statistical frame is used to analyze basic image features. Images are modeled as random variables to estimate their probability distribution in different feature spaces: luminance, color, textures, edges and invariant features. The following low-level features have been selected to analyze image contents in a visual perceptual level:

- **Gray scale and color histogram**: Histograms are represented using partitions of the color space as described in [50]. For gray-scale images, a 256 bins histogram was used. For color images, the RGB space was partitioned in $8 \times 8 \times 8 = 512$ bins.

- **Local Binary Partition histogram**: this is a texture feature that has been used in some image-retrieval systems [3]. For each pixel $P$ in the image, its 8 neighbors are examined comparing their intensity values with the intensity of the pixel $P$. If the intensity of the neighbor pixel is greater, a 1 value is assigned to the corresponding neighbor position, in other case a 0 value is assigned. The calculated values are used to build an 8-positions binary string per pixel. The binary string can take 256 different values, a 256 bins histogram is calculated.

- **Tamura texture histogram**: There are 6 different Tamura features: coarseness, contrast, directionality, linelikeness, regularity, and roughness. In
In this work, the first three were used as they are strongly correlated with human perception. In [14], the original Tamura’s formulation was adapted to calculate each feature per pixel, this is the approach we used. Likewise the color histogram, the space generated by the three features is partitioned in $8 \times 8 \times 8 = 512$ bins to calculate the texture histogram.

- **Sobel histogram**: the Sobel operator is one of the most popular image-processing operators for edge detection [31]. It calculates the intensity difference on the neighborhood of a pixel in the horizontal and vertical directions; this difference may be interpreted as the derivative of a function representing the image in that point. In this implementation, a $3 \times 3$ operator was used to analyze the 8-neighborhood of each pixel. A 512 bins histogram is calculated using the intensity-change measures produced by the operator.

- **Invariant feature histogram**: this technique models features that are invariant under different transformations such as those induced by projecting an object on an image plane. In this work, the distribution of invariant features is calculated according to [50]. Particularly, the integration method based on the kernel function $X(8,0)X(0,16)$ is used; this may be interpreted as a rotation analysis with two circles, with radius 8 and 16 pixels respectively, where the support of the local transformation is calculated.

Figure 4.2 shows some examples of low-level feature extraction, specifically, textures, edges, contrast and directionality. Histograms corresponds to the probability distribution of edges, found by the use of the Sobel operator. This example shows the approximate match between histogram representations and visual perception.

## 4.2 Low-level Metrics

Since low-level features are histograms, they require metrics evaluating differences between probability distributions. The selected features are histograms ranging between 256 and 512 bins, that is high dimensional vector spaces. Similarity measures can be evaluated using distance functions, and it is important to determine the most suitable function for each type of histogram.

For the following definitions, let $H$ and $H'$ be histograms, both with $M$ bins. In addition, let $H_m$ be the $m$-th bin in the histogram $H$. In general, histogram metrics can be classified according to the underlying theory, as follows:

1. To process histograms as vector representations of images or discrete functions
   - **Euclidean distance**: used to evaluate distances in n-dimensional vector spaces
     \[
     L_2(H, H') = \left( \sum_{m=1}^{M} (H_m - H'_m)^2 \right)^{1/2}
     \]
Figure 4.2: Feature extraction results on three example images.
Sobel histogram at the bottom shows an approximate match between visual perceptions and edge features.

Histogram intersection: discrete area under histogram functions.
\[ D_\cap(H, H') = \sum_{m=1}^{M} \min(H_m, H'_m) \]

2. To use information theory to analyze histogram similarity
Jensen-Shannon divergence: a metric based on information theory
\[ D_{JSD}(H, H') = \sum_{m=1}^{M} H_m \log_2 \frac{H_m}{H_m + H'_m} + H'_m \log_2 \frac{H'_m}{H_m + H'_m} \]

3. To apply statistical concepts to calculate proximity between histograms
Relative-Bin Deviation: based on the statistical variation coefficient
\[ D_{rbd}(H, H') = \sum_{m=1}^{M} \frac{\sqrt{(H_m - H'_m)^2}}{1/2 (\sqrt{H_m} + \sqrt{H'_m})} \]

Chi-square distance: adapted from the chi-square test
\[ D_{\chi^2}(H, H') = \sum_{m=1}^{M} \frac{H_m - H'_m}{H_m + H'_m} \]

As it was presented in the previous Subsection, six different features corresponding to some visual perceptions on images have been proposed to represent histopathology image contents. On the other hand, five metrics for histograms
have been presented. The actual problem now is to decide which metric to use for each low-level histogram feature. The most appropriate metric for each feature space was experimentally found, according to the histopathology requirements. Therefore, feature-metric pairs are configured in order to get the better system response. This evaluation is widely detailed in [4] and will be presented in Subsection 6.3.

4.2.1 Metric combination

Many features can be evaluated in an individual metric using a linear combination approach of the feature-metric pairs, according to the equation 3.2. Let $x$ and $y$ be images in the database collection. Let $F_k$ be a function to extract the $k$-th low-level feature and let $D_k$ be a metric able to evaluate differences between a pair of instances of the $k$-th feature. Therefore, a metric to evaluate many low level features is:

$$d(x, y) = \sum_k w_k M_k (F_k(x), F_k(y))$$

where each $w_k$ is a factor that controls the relative importance of each feature-metric pair. The best values for all $w_k$ have been found by exhaustive search as is presented in Subsection 6.3.3. This combination of features produces a representation space in which images are distributed according to the distances of this metric. Figure 4.3 shows the projection of all images in a two-dimensional space using multidimensional scaling. Multidimensional scaling allows the visualization of all images in the database using the information of relative positions between them i.e. distances. In this Figure, images that are nearby share similar visual patterns, such as colors, edges and textures, since the metric used to generate it, was a combination of all of them.
Figure 4.3: Multidimensional Scaling with the low-level metric

*Multidimensional Scaling of distances obtained with the low-level metric in all database. Histopathology images represent medoids of each semantic group.*
Chapter 5

A Semantic Similarity for Histopathology Image Retrieval

Low-level features have been traditionally used in content-based image retrieval to calculate similarity measures and to rank similar images. In the present work, full histograms were used to assess how useful low-level features are in histopathology image retrieval, as it was shown in Chapter 4. Afterward, regarding the semantic gap, a method to link high-level interpretations with the presence of visual patterns is designed. The proposed method is based on machine-learning algorithms, which are trained to recognize histopathology concepts as is described in Subsection 5.2.

Since histograms are by nature high-dimensional representations of a random variable, they are not directly used in the semantic analysis phase. Instead, statistical moments are calculated from histograms, which are useful to describe main characteristics of a probability distribution and to reduce the feature space dimensionality. All selected low-level features are represented by histograms and from each of them the set of statistical moments are calculated. Those moments, herein called meta-features, are calculated per histogram as follows:

Let \( h \) be a given histogram with \( n \) bins and let \( k \) denote the bin index.

- **Mean**: \( \sum_k kh(k) \)
- **Deviation**: \( \sum_k (k - \mu)h(k) \)
- **Skewness**: \( \frac{\mu^3}{\sigma^3} \), the third central moment
- **Kurtosis**: \( \frac{\mu^4}{\sigma^4} - 3 \), the fourth central moment
- **Entropy**: \( -\sum_k h(k)\ln[h(k)] \)
All meta-features are calculated on each of the six histogram features, which amounts to a total of 30 meta-features per image. Images are now represented in a 30-dimensional vector space, in which each dimension represents a visual feature. This space will be called the low-level feature space and will be noted by $F$.

5.1 Semantic Mapper Structure

The main idea in the semantic mapper is to interpret visual image contents into semantic interpretations. 18 main histopathology concepts are used as possible interpretations of image contents as was presented in Section 2.3. Since one image may contain several concepts, a strategy to identify each of those 18 concepts is needed. Hence, a supervised learning approach is followed to decide whether one image contains a concept or not. The semantic mapper structure is composed of 18 classifiers specialized in the recognition of one specific concept.

The semantic analysis process is illustrated in Figure 5.1, where histogram features are first extracted from the original image. Then, meta-features are calculated to obtain an image representation in a 30-dimensional vector space. This representation is the input for the semantic mapper, in which 18 classifiers evaluate the current image. Since one image may contain one or many concepts simultaneously, this mapper does not perform only a classification task, assigning a unique class to an image. Instead, the output of each SVM is used as a score value indicating the presence or absence degree of the related concept in the image. Finally, the output of the semantic mapper is a semantic vector giving a global semantic description of the image.

5.1.1 Individual concept identification

In general, classification algorithms are devised for binary decision problems i.e. classify an instance in one of two classes. The semantic mapper requires to classify an image in many of the 18 different concepts. Thus, to recognize an individual concept from visual features, we need to define a classifier that decides whether the image contains the concept or not. In that way, a class binarization technique has to be followed to allow classifiers identifying one concept from others.

The main class binarization techniques are [19]:

- **Unordered**: it transforms a $n$-class problem into $n$ two-class problems. It produces $n$ classifiers that learn to discriminate their class from all other classes. To make a final decision about the pattern to be classified, this technique requires an external criterion for combining individual predictions into a final prediction.

- **Ordered**: it transforms a $n$-class problem into $n - 1$ binary problems. Classifiers are built using the examples of the current class as positive examples and the examples of the following classes as negative examples.
Figure 5.1: The semantic mapping process.

Original images are first processed to get histogram features. The second step is to calculate metafeatures i.e. statistical moments from histograms. The semantic mapper analyzes metafeatures to produce semantic descriptions.
This technique requires a predefined order on the induced classifiers. To make the final decision for a pattern to be classified, the first classifier is applied. If the pattern belongs to this first class, no other classifiers are evaluated. Otherwise, the pattern is passed to the next classifier until it gets a label.

- **Pairwise**: it transforms a \( n \)-class problem into \( c( c - 1)/2 \) two-class problems. The basic idea is to build a classifier for each pair of classes \( i, j \) using only the examples in those classes. To decode the final decision a voting strategy can be used, selecting the most frequent class.

The semantic mapper does not require to make a final decision for an image, instead, it requires to determine the presence degree of each concept in an image. Hence, ordered class-binarization technique is useless since it would stop the evaluation of other classifiers when it detects one concept in the current image. On the other hand, the pairwise class-binarization technique forces the image to be classified erroneously by some of the binary base classifiers because each classifier must label the image containing one of the concepts it was trained on. Therefore, the unordered class-binarization technique is the best choice to build up the semantic mapper.

However, the unordered class binarization is useful for the semantic mapper only if it does not make a final decision, that is to say, it must preserve all individual predictions that will be interpreted as the presence or absence of the 18 concepts in an image. This predictions are used to construct the semantic feature vector, that is the output of the semantic mapper.

### 5.1.2 Learning algorithms

We assume each concept can be modeled as a complex combination of visual features, capturing the way in which pathologists understand image contents. For example, the elastosis concept could be described only in terms of color and texture but the exact way in which they are mixed is hard to say.

We select Support Vector Machines (SVM) to model a classification strategy for the recognition of histopathology concepts. Although SVM are linear classifiers, by the use of the kernel trick it is possible to apply a nonlinear map to change the representation of images into one that is more suitable for the problem. That means patterns in the low-level feature space are represented as vectors in some dot product space \( \mathcal{H} \), in which \( \mathcal{H} \) is dimensionally greater than \( \mathcal{F} \).

\[
\phi : \mathcal{F} \rightarrow \mathcal{H} \tag{5.1}
\]

The freedom to choose the mapping \( \phi \) will allow to design a large variety of learning algorithms based on SVM. Though the classifier is a hyperplane in the high-dimensional feature space it may be non-linear in the original visual-feature space, resulting in a complex model of the pathologists knowledge. The mapping function is implicitly defined by a kernel function [47].
Figure 5.2: An example SVM hyperplane and the margin. Patterns represented as crosses in the figure are difficult to classify, since they are near the separating hyperplane. Images that clearly contain the concept have a positive distance to the hyperplane, represented by squares. Images that do not contain the concept have a negative distance to the hyperplane, represented by circles.

5.1.3 Semantic feature vector

To build the semantic feature vector, the definitions of the separating hyperplane and geometrical margin are used. Each of the 18 SVMs are binary classifiers, with a separating hyperplane represented by a pair \((w, b) \in \mathcal{H} \times \mathbb{R}\), where \(w\) is the vector that generates the hyperplane. The decision function of such binary classifiers is a function from the augmented space \(\mathcal{H}\) to a binary decision:

\[
f_{w,b} : \mathcal{H} \rightarrow \{ \pm 1 \} \tag{5.2}
\]

This function is modeled as the sign of the resultant value when the pattern \(x\) is evaluated in the linear hyperplane function.

\[
x \mapsto f_{w,b}(x) = \text{sgn}(\langle w, x \rangle + b) \tag{5.3}
\]

By definition, the value of \(\langle w, x \rangle + b\) is the distance from the separating hyperplane to the pattern \(x\). In the semantic mapper, this distance is used as the presence degree of each concept, instead of the binary decision i.e. distance patterns from the hyperplane have a high presence degree while close patterns have a small presence degree. Since SVM are designed to find a margin between
the separating hyperplane and the support vectors with a distance equals to 1, the output of a SVM in the semantic mapper could be interpreted according to the interval in which the value belongs to, as is shown in Figure 5.2.

5.1.4 Model generalization

The generalization problem consist of build a classification function that may be robust to the variation of the input data. It is well known that SVM generalization performance depends on a good setting of the kernel parameters. The problem of optimal parameter selection is further complicated by the fact that SVM model complexity \(^1\) depends on those parameters. Selecting a particular kernel type and kernel function parameters is usually based on application-domain knowledge and also should reflect distribution of input values of the training data. In this work, the specific configuration for each SVM is experimentally evaluated to reach a good and general model.

5.2 Semantic Mapper Training

The goal of the semantic mapper is to analyze visual image contents and to determine the degree of presence of each histopathology concept. Since this process is not a multiclass task, for each concept a binary classifier has been modeled to recognize it against the other concepts. That means, for a specific instance, each SVM has to recognize between two possible classes: the positive class, if the instance exhibits the concept; and the negative class, if the instance does not contain the concept. This convention applies only for the training phase since the semantic mapper is devised to get the distance to the separating hyperplane, as was mentioned in Section 5.1.

As output, the semantic mapper gives 18 real values indicating the presence degree of each concept in the original image. An additional classifier was incorporated to the semantic mapper to identify images not containing any concept. It was included since some images in the dataset do not contain any differential diagnosis clue. It amounts to 19 classifiers.

The training process of each SVM followed an extensive experimentation to identify the better possible configuration before include it in the semantic mapper. The detailed training process is described in the following Subsections.

5.2.1 Replicate datasets

The original dataset is composed of 1,502 images, with their corresponding meta-features and labels. For each intended classifier, a dataset is entailed with exactly the same attributes except for the class label which can only have two possible values: positive if the example belongs to this group and negative otherwise.

\(^1\)And hence its generalization performance
5.2.2 Stratified sampling

Each dataset is split in two subsets using a stratified sampling approach. 80\% is used for training and 20\% for testing. A 10-fold cross-validation scheme on the training set is used to evaluate the classification rate with different hyperparameters. The test dataset is set aside and used at the end for calculating the final classification error rate.

5.2.3 Oversampling

Each dataset contains only information about two classes: the positive and negative. The number of the positive samples is usually lower than the negative class and should be balanced in order to improve the classifier performance. Each dataset has a 7\% of instances in average, which corresponds to the positive example class that is attempting to classify. This difficulty is known in machine learning as the class imbalance problem, and there are several approaches in the literature about how to solve it. In this work an oversampling approach has been followed, including new samples of the minor class with the same attributes of an existing one. The final training datasets were approximately composed of 2,200 instances, i.e 180\% of the initial dataset.

5.2.4 SVM classifiers

A SVM is a linear classifier that could be configured with a kernel function to induce a new feature space with higher dimensionality. In order to select the best kernel function for each SVM classifier and to generalize the nature of the associated concept, six different kernel functions were studied: 4 using the Gaussian kernel and 2 using the polynomial kernel.

- **Polynomial Kernel.** (inhomogeneous) \( k(x, x') = (x \cdot x' + 1)^e \), with exponent \( e \) set to 5 and 20

- **Gaussian Kernel.** \( k(x, x') = \exp(-\gamma \|x - x'\|^2) \), with four different values for \( \gamma \): 0.001, 0.1, 0.5 and 1

Each kernel configuration represents a different learning model, thus, the following steps in the training process will work with six different models per concept. Regarding that the semantic mapper needs 19 classifiers, it sums up to 114 different SVM evaluated configurations.

5.2.5 Regularization parameter

The parameter \( \lambda \) is the regularization parameter to manage the complexity of a SVM classifier. This parameter was evaluated for each learning model in the interval \((0, 50]\) with steps \( \delta = 0.1 \). For each value of \( \lambda \), a 10-fold cross validation is performed to estimate the error rate with this complexity value. Then, an error vs. complexity plot is drawn to select the best value for \( \lambda \). An example
error-complexity plot is presented in Figure 5.3, that shows how the error rate falls while complexity increases. The training error rate falls to zero, overfitting the training set, and the test error, measured on non-seen instances\(^2\), is not improved after some complexity value, that can be selected as the best choice for \(\lambda\) in this model. The best complexity value for such model is found as the minimum complexity that outputs the minimum error on the test set.

### 5.2.6 Classifier evaluation

For each concept in the semantic mapper, six different learning models have been evaluated, and for each model the best parameter \(\lambda\) has been identified. Now, the best learning model per concept can be selected identifying the minimum error rate among all six models. So that, the six models with their best \(\lambda\) parameter are trained again, using all instances in the training dataset. The obtained classifier is assessed using the test dataset, with non-seen instances in any previous step. This model is applied 30 times to the test dataset using 10-folding cross validation to assess the real error with unknown instances. The obtained error rate between models allows to select the best classifier among the 6 possible ones.

Figure 5.4 shows the error rate of each model per concept\(^3\). The Figure

\(^2\)This non-seen instances come from the 10-folding partition. They do not come from the test dataset.

\(^3\)Although 6 models were evaluated, Figure 5.4 shows only the error rates for 5 of them.
shows that some concepts are difficult to recognize, even using different SVM models, e.g. the concept number 18, corresponding to noisy images that do not contain any concept. On the other hand, there are some other concepts relatively easier to identify, for which almost all models output low error rates, e.g. the concept number 2 associated to cystic change, gets a maximum error rate lower than 7.5%. Figure 5.4 also shows that the SVM model with a RBF kernel configured with $\gamma = 1$ gets the minimum error rate for all histopathology concepts.

Finally, a confidence interval for the best model and a final error-rate using the test dataset are calculated and reported. The final configuration of each SVM in the semantic mapper is reported in Table 5.1.

The not-included model corresponds to the polynomial kernel with exponent value equals to 5, which outputs the higher error rates among all models and does not allow to compare others in this plot.
Table 5.1: Final configuration of the SVM classifiers

<table>
<thead>
<tr>
<th>Concept</th>
<th>Lambda</th>
<th>Test Error</th>
<th>Deviation</th>
<th>Conf. Int.</th>
<th>Train. Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pilosebaceous annexa</td>
<td>25</td>
<td>11.3%</td>
<td>0.66</td>
<td>0.54</td>
<td>1.99%</td>
</tr>
<tr>
<td>Cystic change</td>
<td>4</td>
<td>4.65%</td>
<td>0.62</td>
<td>0.51</td>
<td>0.69%</td>
</tr>
<tr>
<td>Elastosis</td>
<td>15</td>
<td>10.3%</td>
<td>1.31</td>
<td>1.07</td>
<td>2.13%</td>
</tr>
<tr>
<td>Eccrine glands</td>
<td>8</td>
<td>16.61%</td>
<td>1.03</td>
<td>0.84</td>
<td>5.65%</td>
</tr>
<tr>
<td>Lymphocyte infiltrate</td>
<td>12</td>
<td>18.27%</td>
<td>1.79</td>
<td>1.45</td>
<td>4.12%</td>
</tr>
<tr>
<td>Perineural invasion</td>
<td>16</td>
<td>0%</td>
<td>0.48</td>
<td>0.39</td>
<td>0.09%</td>
</tr>
<tr>
<td>Lesion with fibrosis</td>
<td>15</td>
<td>10.3%</td>
<td>0.72</td>
<td>0.58</td>
<td>1.71%</td>
</tr>
<tr>
<td>Micronodules</td>
<td>5</td>
<td>7.64%</td>
<td>1.05</td>
<td>0.85</td>
<td>1.85%</td>
</tr>
<tr>
<td>Necrosis</td>
<td>17</td>
<td>3.99%</td>
<td>0.55</td>
<td>0.45</td>
<td>0.23%</td>
</tr>
<tr>
<td>N-P-C, elastosis</td>
<td>12</td>
<td>5.98%</td>
<td>0.73</td>
<td>0.59</td>
<td>0.6%</td>
</tr>
<tr>
<td>N-P-C, fibrosis</td>
<td>15</td>
<td>5.98%</td>
<td>0.95</td>
<td>0.77</td>
<td>1.16%</td>
</tr>
<tr>
<td>N-P-C, infiltrate</td>
<td>10</td>
<td>18.27%</td>
<td>2.70</td>
<td>2.2</td>
<td>5.09%</td>
</tr>
<tr>
<td>N-P-C, pilosebaceous</td>
<td>15</td>
<td>7.97%</td>
<td>0.68</td>
<td>0.56</td>
<td>0.32%</td>
</tr>
<tr>
<td>N-P-C, trabeculae</td>
<td>5</td>
<td>2.33%</td>
<td>0.31</td>
<td>0.25</td>
<td>0.42%</td>
</tr>
<tr>
<td>Morpheaform pattern</td>
<td>15</td>
<td>2.99%</td>
<td>0.63</td>
<td>0.51</td>
<td>0.32%</td>
</tr>
<tr>
<td>Rod trabeculae</td>
<td>10</td>
<td>12.96%</td>
<td>0.75</td>
<td>0.61</td>
<td>1.85%</td>
</tr>
<tr>
<td>Ulceration</td>
<td>15</td>
<td>1.99%</td>
<td>0.50</td>
<td>0.41</td>
<td>0.09%</td>
</tr>
<tr>
<td>Noise</td>
<td>25</td>
<td>30.23%</td>
<td>2.52</td>
<td>2.05</td>
<td>8.79%</td>
</tr>
<tr>
<td>Sanguineous vessel</td>
<td>10</td>
<td>13.29%</td>
<td>1.59</td>
<td>1.3</td>
<td>3.15%</td>
</tr>
</tbody>
</table>

5.3 Semantic Metric

In previous Sections the structure and configuration of the semantic mapper has been introduced. Using the semantic mapper, the system is able to analyze the image content to extract the semantic features. All images in the database have been processed using the semantic mapper, and semantic features have been stored. In addition, when a new image is archived in the system, the semantic mapper also process it. This recently archived image may be simply stored or may be used as query image. Now, the actual problem is how to compare those semantic features.

Semantic features are represented by vectors in $\mathbb{R}^d$, with $d = 19$, in which each dimension represents a presence degree of a concept. This semantic vector provides information about the content and an interpretation of the overall image. To compare images in a semantic way, the Tanimoto coefficient was selected, which is a generalization of the Jacquard coefficient. In this problem, Tanimoto coefficient can estimate how many positions in the semantic vectors are showing coincidences, emphasizing the similarity between concepts shared by both images. Given two semantic vectors $A$ and $B$, the Tanimoto coefficient
gives a distance measure between both images as follows:

\[ T(A, B) = 1 - \frac{A \cdot B}{\|A\|^2 + \|B\|^2 - A \cdot B} \]  

(5.4)

This semantic metric generates a representation space in which images are distributed according to its distances, in the same way as the low-level metric. Figure 5.5 shows the projection of all images in a two-dimensional space using multidimensional scaling. This Figure shows blue points to represent images according to distances in the space induced by this metric, and also shows 19 histopathology images in that space.

As was mentioned in Chapter 4, images presented in a multidimensional scaling are closer according to a feature-metric interpretation. In Figure 5.5, images are distributed in a semantic space, in which those that are closer, share histopathology concepts.
Figure 5.5: Multidimensional scaling with the semantic metric

Multidimensional scaling from the distances obtained with the semantic metric. Histopathology images represent medoids of each semantic group.
Chapter 6

Effectiveness Evaluation

6.1 Performance Measures for Image Retrieval

When evaluating effectiveness of information retrieval systems, it is important to define what a perfect system response would be like, named ground truth [39]. Many approaches to define a ground truth have been used in the literature, including: user assessment of relevancy; the automatic definition of classifications from available image annotations; and manual definition of user information needs. In this work a ground truth was defined through the analysis, annotation and coding of a set of images, performed manually by pathologists.

The available image collection to evaluate the system is a database of 6,000 histopathology images, from which a subset of 1,502 images was selected as ground truth. The ground truth, created by pathologists, is composed of 18 general concepts, corresponding to possible information needs in pathology. In average, each concept has 75 relevant images in the collection. In addition, one image may contain many concepts, that is to say, an image should be relevant for different information needs and should appear in the results of those different queries.

For the experimental test, each image in the result set is evaluated against the ground truth to see whether it is relevant or not. Müller et al [22] presents a framework to evaluate CBIR systems in order to report comparable results from different research centers in a standardized way. The following performance measures are used to evaluate the behavior of image retrieval systems, regarding the user information needs:

- $\text{Rank}_1$, $\overline{\text{Rank}}$ and $\overline{\text{Rank}}$: In general, rank is defined as the position in which a relevant image is presented to the user. So $\text{Rank}_1$ is the average position in which the first relevant image appears, $\overline{\text{Rank}}$ is the average position of all relevant images, and $\overline{\text{Rank}}$ is the normalized average rank of all relevant images independently of their number.

- Precision: It is defined as the percentage of relevant images that the
system found among all the images delivered to the user.

- **Recall**: It is defined as the percentage of relevant images that the system has found among all the relevant images that it should retrieve.

- **\(N_R\)**: Average number of relevant images for one query.

- **\(P(1)\)**: Average precision for the first retrieved image.

- **\(P(20), P(50), P(N_R)\)**: System precision when it retrieves 20, 50 and \(N_R\) images respectively.

- **\(R(0.5)\) and \(R(100)\)**: Recall value when the system has reached a precision of 0.5 and when the system has retrieved exactly 100 images.

- Query response time.

The most representatives of those performance measures are precision and recall, that are more formally defined as:

\[
\text{precision} = \frac{\text{number of relevant retrieved images}}{\text{number of all retrieved images}}
\]

\[
\text{recall} = \frac{\text{number of relevant retrieved images}}{\text{number of all relevant images}}
\]

Precision can be measured for different values of recall and the average precision of the \(n\)-th result is reported as \(P(n)\) to compare experiments. A most appropriate way to report different values of \(P(n)\) is using a precision vs recall graph, which provides information about the behavior of the system in many points.

Different evaluation measures for information retrieval have been proposed. Many of those measures assess the quality of the retrieved results since different viewpoints. In general, all evaluation measures in information retrieval are highly correlated and the use of any in particular is a choice that depends on the specific application. In [14] a systematic empirical study of such evaluation measures was done for content-based image retrieval. This study corroborates the high correlation between measures and suggests the use of \(P(1)\) as a basic measure to compare results.

### 6.2 Experimental design

The experimental framework uses the subset of 1,502 annotated images which allows to determine whether results are relevant or not. Each experiment is composed of 30 queries (images) randomly selected from the annotated collection. When results are delivered, the evaluation procedure verifies if each result belongs to the same query topic of the example image, marking it as relevant or irrelevant. The obtained results for each random query are averaged with other results to get a unique measurement per experiment.
Through the remain of this document, all analysis are based on this experimental framework, and the specific results for each approach is discussed in future sections reporting the most relevant measures. In particular, $P(1)$ and the precision vs recall graph has been selected to analyze and compare models for histopathology image retrieval.

Different experiments have been performed in order to evaluate the system effectiveness. Since the system has three main strategies to search images, it is important to compare their performance. The three main strategies are based on the type of features that they use to calculate similarity between images: individual low-level features, a combination of low-level features, and semantic features. The following is the outline of the experimentation made in each case:

1. **Best metric for each low-level feature**: In Section 4.1 six different low-level histogram features were introduced. In Section 4.2 five different histogram metrics were presented. To identify the best metric for each low-level feature, an experimental evaluation was made using different information retrieval performance measures. Results are presented in Subsection 6.3.1.

2. **Best individual feature-metric pair**: Once the best feature-metric pairs have been identified, they are evaluated to identify the better performance using only one low-level histogram feature. Results are presented in Subsection 6.3.2.

3. **Combination of low-level features**: An evaluation of a linear combination of low-level histogram features has been done. Individual low-level features evaluate only one aspect of image contents, and mixing different features may improve the system response. Results are presented in Subsection 6.3.3.

4. **Semantic features**: The semantic mapper produces features that the system can use to calculate the conceptual proximity between images. This strategy is expected to have better performance than low-level features since it compares images using domain knowledge, and results may better correspond with user information needs. This results are presented in Section 6.4.

### 6.3 Low-level Metric Evaluation

Histopathology Image Retrieval based on low-level features has been described in Chapter 4. In general, image retrieval based on low-level features, requires a set of visual features and metrics to identify the most similar results. This Section presents different experiments to analyze the effectiveness of low-level features to retrieve histopathology images.
6.3.1 Best metrics for low-level features

Since the system has six different low-level histogram features and five different histogram metrics, the first evaluation aims identifying the best metric for each feature. One can says that a good metric for a specific feature space is such that, given a histopathology image, is able to determine the closer image set according to the application domain knowledge. Hence, using the performance measures described in Section 6.1, a *effectiveness rank* was designed. This strategy averages the relative position of the metric among performance measures.

The following is the *effectiveness rank* strategy. Consider a feature space $F$ and $n$ metrics $D_1$ to $D_n$. We want to know what metric is better to calculate the similarity between any pair of features in $F$ for the image retrieval task:

1. First, $n$ experiments are performed using each metric $D_j$ in a feature space and a set of some information-retrieval (IR) measures are registered and averaged.

2. Then, we can sort the obtained results for each metric according to each IR measure in separated lists. For example, average results for all metrics are sorted according to their $P(1)$ performance in a feature space. In the top of the $P(1)$ list is the metric that better maximize this measure in that feature space.

3. For each metric $M_j$ we calculate the average position in which this metric was ranked among all IR measures. For example, supose that three IR measures are used to evaluate the Jensen-Shannon Divergence metric in a fixed feature space (may be color): $P(1)$, $\overline{\text{Rank}}$ and $R(0.5)$. The metric gets the first position according to the $P(1)$ measure, the second position according to $\overline{\text{Rank}}$ and the third position according to $R(0.5)$. In average, this metric would get the second position $((1 + 2 + 3)/3)$ among all the experimentation for the evaluated feature space.

4. At last, the average position is used to sort metrics again in each feature space.

5. In the top of the list is the metric that got the best position most frequently, according to all selected IR measures.

This strategy was applied for all features combined with all metrics, and results are presented in Table 6.1. This Table shows that the metric $D_{rbd}$ (Relative Bin Deviation) got a good performance in four of the six evaluated feature spaces. The Sobel histogram is better compared using the $D_{JSD}$ (Jensen-Shannon Divergence) metric and Tamura textures got good results using $L_2$ (Euclidean Distance).
Table 6.1: Effectiveness rank for each metric

Effectiveness rank for each metric in all feature spaces. In bold, the best average position per feature.

<table>
<thead>
<tr>
<th>Feature</th>
<th>$D_{\chi^2}$</th>
<th>$D_{JSD}$</th>
<th>$D_{\cap}$</th>
<th>$L_2$</th>
<th>$D_{rub}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gray scales</td>
<td>3.77</td>
<td>2.31</td>
<td>4.62</td>
<td>2.62</td>
<td>1.69</td>
</tr>
<tr>
<td>RGB</td>
<td>4.00</td>
<td>2.15</td>
<td>4.51</td>
<td>2.85</td>
<td>1.46</td>
</tr>
<tr>
<td>Invariant features</td>
<td>4.08</td>
<td>2.23</td>
<td>4.46</td>
<td>2.69</td>
<td>1.54</td>
</tr>
<tr>
<td>Local Binary Partition</td>
<td>3.92</td>
<td>1.77</td>
<td>4.62</td>
<td>2.92</td>
<td>1.77</td>
</tr>
<tr>
<td>Sobel</td>
<td>3.92</td>
<td>1.69</td>
<td>4.62</td>
<td>2.69</td>
<td>20.8</td>
</tr>
<tr>
<td>Tamura</td>
<td>3.85</td>
<td>2.38</td>
<td>4.69</td>
<td>1.92</td>
<td>2.15</td>
</tr>
</tbody>
</table>

Table 6.2: Average precision values for the best feature-metric pairs

<table>
<thead>
<tr>
<th>Feature</th>
<th>Metric</th>
<th>P(1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sobel</td>
<td>Jensen-Shannon Divergence</td>
<td>61%</td>
</tr>
<tr>
<td>RGB Histogram</td>
<td>Relative Bin Deviation</td>
<td>53%</td>
</tr>
<tr>
<td>Local Binary Partition</td>
<td>Relative Bin Deviation</td>
<td>53%</td>
</tr>
<tr>
<td>Gray Histogram</td>
<td>Relative Bin Deviation</td>
<td>50%</td>
</tr>
<tr>
<td>Tamura Texture</td>
<td>Euclidean Distance</td>
<td>39%</td>
</tr>
<tr>
<td>Invariant Feature Histogram</td>
<td>Relative Bin Deviation</td>
<td>36%</td>
</tr>
</tbody>
</table>

6.3.2 Individual feature-metric pairs

The previous Subsection presented the way in which the best metric was identified for each feature. Once feature-metric pairs have been configured, it is important to evaluate and compare them analyzing the precision rate in ranking relevant images. This evaluation was made under the experimental framework.

Results are shown in Table 6.2, where features are associated with the metric that outputs the best performance in that space. Reported values correspond to the average precision of the 30 random queries at the first result, e.g. the Sobel-JSD metric returns a relevant image in the first position in the 61% of the cases. Table 6.2 also shows features sorted by their average precision rate for the first image retrieved. This test shows that edges, codified into the Sobel Histogram, performs better than any other histogram feature, and this suggest that edges is an important feature for differential diagnosis in pathology. Figure 6.2 presents the precision vs. recall graph for this individual feature evaluation comparing the effectiveness with more detail.

This evaluation was extended to each individual concept, i.e. each feature-metric pair was evaluated to identify which is the best per concept. Results are presented in Table 6.3. This experimentation shows that results for some query topics are more difficult to retrieve than others. For example, when the user is searching for images with Perineural Invasion, the best feature-metric pair
Figure 6.1: Precision vs. Recall graph for individual feature-metric pairs

Table 6.3: Query topics with the percentage of relevant images

<table>
<thead>
<tr>
<th>Concept</th>
<th>Relevant</th>
<th>Feature-metric</th>
<th>P(1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pilosebaceous annexa</td>
<td>10.3%</td>
<td>LBP-$D_{rbd}$</td>
<td>60%</td>
</tr>
<tr>
<td>Cystic change</td>
<td>4.5%</td>
<td>LBP-$D_{rbd}$</td>
<td>81%</td>
</tr>
<tr>
<td>Elastosis</td>
<td>8.3%</td>
<td>RGB-$D_{rbd}$</td>
<td>58%</td>
</tr>
<tr>
<td>Eccrine glands</td>
<td>9.9%</td>
<td>RGB-$D_{rbd}$</td>
<td>58%</td>
</tr>
<tr>
<td>Lymphocytic infiltrate</td>
<td>9.3%</td>
<td>RGB-$D_{rbd}$</td>
<td>47%</td>
</tr>
<tr>
<td>Perineural invasion</td>
<td>0.4%</td>
<td>LBP-$D_{rbd}$</td>
<td>17%</td>
</tr>
<tr>
<td>Lesion with fibrosis</td>
<td>6.0%</td>
<td>Sobel-$D_{JSD}$</td>
<td>64%</td>
</tr>
<tr>
<td>Micronodules</td>
<td>2.7%</td>
<td>LBP-$D_{rbd}$</td>
<td>37%</td>
</tr>
<tr>
<td>Necrosis</td>
<td>2.1%</td>
<td>RGB-$D_{rbd}$</td>
<td>31%</td>
</tr>
<tr>
<td>N-P-C, elastosis</td>
<td>3.5%</td>
<td>RGB-$D_{rbd}$</td>
<td>73%</td>
</tr>
<tr>
<td>N-P-C, fibrosis</td>
<td>3.3%</td>
<td>RGB-$D_{rbd}$</td>
<td>64%</td>
</tr>
<tr>
<td>N-P-C, infiltrate</td>
<td>11.7%</td>
<td>Sobel-$D_{JSD}$</td>
<td>59%</td>
</tr>
<tr>
<td>N-P-C, pilosebaceous annexa</td>
<td>2.5%</td>
<td>Sobel-$D_{JSD}$</td>
<td>42%</td>
</tr>
<tr>
<td>N-P-C, trabeculae</td>
<td>0.9%</td>
<td>Tamura-$D_{rbd}$</td>
<td>57%</td>
</tr>
<tr>
<td>Morpheaform pattern</td>
<td>2.5%</td>
<td>Sobel-$D_{JSD}$</td>
<td>77%</td>
</tr>
<tr>
<td>Rod trabeculae</td>
<td>4.0%</td>
<td>Sobel-$D_{JSD}$</td>
<td>47%</td>
</tr>
<tr>
<td>Ulceration</td>
<td>1.0%</td>
<td>LBP-$D_{rbd}$</td>
<td>60%</td>
</tr>
<tr>
<td>Sanguineous vessel</td>
<td>8.1%</td>
<td>Sobel-$D_{JSD}$</td>
<td>64%</td>
</tr>
</tbody>
</table>
is LBP-$D_{rbd}$\(^1\) that reaches an initial precision of 17%. This could be partially explained by the fact that, in the experimental collection, only the 0.4% of images exhibits this concept. However, other concepts with larger examples in the collection are also difficult to retrieve. For example, \textit{Lymphocyte infiltrate}, the fourth largest in the collection reaches a maximal initial precision of 47% with RGB-$D_{rbd}$\(^2\). Other concepts such as \textit{Cystic Change} and \textit{Morpheaform Pattern} got higher precision rates (81% and 77% respectively) although example images are not very large (4.5% and 2.5% respectively).

### 6.3.3 Low-level feature combination

Experiments using many low-level features combined in a single metric were also done. According to the Definition 4.1, a linear combination of features requires the use of different importance factors. The identification of those factors was achieved by exhaustive search, using different information retrieval measurements through the \textit{performance rank}. The performance rank strategy is the same as that described in Subsection 6.3.1. The search for linear-combination weights, finds that 50% for Local Binary Partition, 30% for Sobel Histogram and 20% for RGB Color Histogram is a good combination of low-level features.

---

\(^1\)Local Binary Partition - Relative Bin Deviation
\(^2\)Color Histogram - Relative Bin Deviation
Testing the combination of different features, an average precision rate of 67% was achieved for the first image retrieved. In the precision vs recall graph this configuration outperforms the individual feature-metric pairs previously tested. This tendency is better shown in Figure 6.3, in which the combination approach is compared with the best-three individual metrics. The linear combination of features shows a better performance than individual features i.e. precision-recall pairs are the best for the combination of features in almost all the cases.

6.4 Semantic Metric Evaluation

The semantic similarity model was introduced in Chapter 5. This strategy uses a semantic mapper to extract conceptual features, which can be interpreted as the presence degree of histopathology concepts. In such way, similar images under this framework are expected to share concepts rather than only visual features.

6.4.1 General Evaluation

To analyze the contribution of the semantic similarity approach, several experiments were run and compared with the effectiveness of the best known model
using low-level features i.e. the linear combination of histogram features, that got a maximum initial precision of 67% as was presented in Subsection 6.3.3. Table 6.4 shows the performance of the CBIR system using low-level features in one case, and semantic features in the other case. All performance measures for image retrieval have shown that the semantic similarity outperforms the low-level similarity. Specifically, Rank1 indicates in average, which position was gotten by the first relevant image. Results show that the low-level similarity locates the first useful image for the user in the 8th position in the result list. On the other hand, the semantic similarity is able to locate it in the second position, in average.

In addition, the initial precision with the best low-level similarity indicates that, in average, the first image in the result list is relevant in the 67% of the cases, while the semantic similarity does it in the 80% of the cases. Precision decreases as the number of retrieved images increases in both low-level and semantic similarities. However, the decreasing rate using the semantic similarity is smoother. For example, from initial precision $P(1)$ to precision when 20 images have been retrieved $P(20)$ the low-level similarity falls from 67% to 30%, that means 37 precision points. On the other hand, the semantic similarity, decreases from $P(1) = 80\%$ to $P(20) = 59\%$, i.e. 21 precision points.

A smoother decrease in precision is specially desirable in the top-retrieved results. In general, when users are searching for something in particular, they want to get relevant results in the first page to save time, one of the main goals of a retrieval system. Since this view point, the proposed semantic similarity not only outperforms the output of the low-level similarity, but also remains higher while other images are retrieved. This tendency is better shown in the precision vs. recall graph presented in Figure 6.4. Results have shown that the proposed model to represent histopathology domain knowledge in a semantic feature space, greatly improves the performance of the system.

### 6.4.2 Topic Evaluation

As it was mentioned before, the semantic approach has improved the system effectiveness compared with the low-level feature approach. This Subsection presents another evaluation of the semantic image retrieval method, detailing its performance by query topic. This evaluation allows to analyze what is expected to get when some user is searching for images with specific contents.

Table 6.5 presents all histopathology concepts (also called query topics) and several information retrieval measures. The first column presents the average

<table>
<thead>
<tr>
<th>Model</th>
<th>Rank1</th>
<th>Rank</th>
<th>P(1)</th>
<th>P(20)</th>
<th>P(50)</th>
<th>P(100)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low-level similarity</td>
<td>8.22</td>
<td>0.28</td>
<td>0.67</td>
<td>0.30</td>
<td>0.21</td>
<td>0.16</td>
</tr>
<tr>
<td>Semantic similarity</td>
<td>1.96</td>
<td>0.07</td>
<td>0.80</td>
<td>0.59</td>
<td>0.51</td>
<td>0.45</td>
</tr>
</tbody>
</table>
Figure 6.4: Precision vs Recall graph using two similarities

*Precision vs Recall graph comparing the system performance using two similarities: combination of low-level features and the semantic similarity*

Table 6.5: Performance measures for image retrieval grouped by search topic

<table>
<thead>
<tr>
<th>Topic</th>
<th>Rank</th>
<th>P(1)</th>
<th>P(20)</th>
<th>P(50)</th>
<th>P(100)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pilosebaceous annexa</td>
<td>6.83</td>
<td>0.09</td>
<td>0.70</td>
<td>0.67</td>
<td>0.62</td>
</tr>
<tr>
<td>Cystic change</td>
<td>1.23</td>
<td>0.04</td>
<td>0.90</td>
<td>0.87</td>
<td>0.75</td>
</tr>
<tr>
<td>Elastosis</td>
<td>7.57</td>
<td>0.13</td>
<td>0.77</td>
<td>0.64</td>
<td>0.52</td>
</tr>
<tr>
<td>Eccrine glands</td>
<td>3.40</td>
<td>0.10</td>
<td>0.77</td>
<td>0.57</td>
<td>0.54</td>
</tr>
<tr>
<td>Lymphocyte infiltrate</td>
<td>1.87</td>
<td>0.12</td>
<td>0.70</td>
<td>0.52</td>
<td>0.45</td>
</tr>
<tr>
<td>Perineural invasion</td>
<td>2.83</td>
<td>0.02</td>
<td>0.33</td>
<td>0.10</td>
<td>0.04</td>
</tr>
<tr>
<td>Lesion with fibrosis</td>
<td>1.37</td>
<td>0.10</td>
<td>0.80</td>
<td>0.64</td>
<td>0.54</td>
</tr>
<tr>
<td>Micronodules</td>
<td>4.33</td>
<td>0.10</td>
<td>0.43</td>
<td>0.35</td>
<td>0.26</td>
</tr>
<tr>
<td>Necrosis</td>
<td>4.70</td>
<td>0.10</td>
<td>0.67</td>
<td>0.37</td>
<td>0.25</td>
</tr>
<tr>
<td>N-P-C, elastosis</td>
<td>1.43</td>
<td>0.07</td>
<td>0.83</td>
<td>0.62</td>
<td>0.42</td>
</tr>
<tr>
<td>N-P-C, fibrosis</td>
<td>1.53</td>
<td>0.05</td>
<td>0.77</td>
<td>0.55</td>
<td>0.45</td>
</tr>
<tr>
<td>N-P-C, infiltrate</td>
<td>1.50</td>
<td>0.14</td>
<td>0.77</td>
<td>0.67</td>
<td>0.58</td>
</tr>
<tr>
<td>N-P-C, pilosebaceous annexa</td>
<td>2.67</td>
<td>0.12</td>
<td>0.67</td>
<td>0.39</td>
<td>0.24</td>
</tr>
<tr>
<td>N-P-C, trabeculae</td>
<td>1.36</td>
<td>0.05</td>
<td>0.71</td>
<td>0.26</td>
<td>0.15</td>
</tr>
<tr>
<td>Morpheaform pattern</td>
<td>1.27</td>
<td>0.02</td>
<td>0.83</td>
<td>0.79</td>
<td>0.55</td>
</tr>
<tr>
<td>Rod trabeculae</td>
<td>1.27</td>
<td>0.11</td>
<td>0.83</td>
<td>0.53</td>
<td>0.37</td>
</tr>
<tr>
<td>Ulceration</td>
<td>35.47</td>
<td>0.14</td>
<td>0.60</td>
<td>0.27</td>
<td>0.15</td>
</tr>
<tr>
<td>Sanguineous vessel</td>
<td>1.97</td>
<td>0.08</td>
<td>0.70</td>
<td>0.67</td>
<td>0.64</td>
</tr>
</tbody>
</table>
Rank \(_1\) i.e. the position in which the first relevant image appears in the result list. For 9 of the 18 concepts, the first image retrieved appears in the first or second position in the result list, and for other 5 concepts more, the first image retrieved is before the fifth position. In addition, precision values show a good system response for almost every query topic. There are some concepts such as perineural invasion and micronodules that have a poorer performance in terms of precision, suggesting that these query topics may be difficult to retrieve.

This evaluation also compares the system response against the performance of the low-level similarity as is presented in Table 6.6. Compared results are based on the \(P(1)\) measure, since in general, this measure is the maximum value in the precision vs. recall graph. In average, the increment in the initial precision is 16% approximately. For some concepts, the semantic similarity improves the low-level similarity by up to 36%. This improvement may be explained by the fact that the low-level similarity relates visual features using a linear combination approach, while the semantic similarity is based on a complex model of the histopathology knowledge. By the use of SVMs in the semantic mapper, visual features are combined in a non-linear way, using the kernel trick approach.

The error rate of each SVM in the semantic mapper is also shown in Table 6.6. Intuitively, one can relate the error rate of each SVM with the precision rate of the semantic similarity. However, there is no a direct relationship between them, because one image may contain one or several concepts, and the semantic similarity evaluates how many concepts are shared by two images. In such way, the precision rate of the semantic similarity is in fact related with the error rate of the semantic mapper, but taking into account that the semantic similarity evaluates the presence or absence of all concepts simultaneously instead of searching only one of them.

6.5 A Medical Image Retrieval Challenge

6.5.1 Background

The research community in Content-based Medical Image Retrieval has been working for several years. In the beginning, some isolated works were reported proposing different approaches. However, after reviewing the literature, different models are difficult to compare because each one makes use of different datasets and reports different performance measures. Many computer science disciplines have been using a common experimental configuration to solve a specific problem, and that idea has motivated the organization of the Cross Language Evaluation Forum (CLEF). This forum promotes the investigation of search models for data collections in different languages and different modalities\(^3\), in order to be able to compare strategies, models, results and ideas.

Specifically, the ImageCLEFmed is a retrieval challenge in a collection of medical images [36], which is organized yearly to stimulate the development of

\(^3\)Also known as multimedia: text, audio, images and video among others.
Table 6.6: Comparative results per topic

Comparative results per topic of the initial precision on both semantic and low-level similarity. Last column shows the classification error of the SVM in the semantic mapper associated to each concept.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Semantic $P(1)$</th>
<th>Low-level $P(1)$</th>
<th>SVM error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pilosebaceous annexa</td>
<td>70%</td>
<td>60%</td>
<td>11.3%</td>
</tr>
<tr>
<td>Cystic change</td>
<td>90%</td>
<td>81%</td>
<td>4.65%</td>
</tr>
<tr>
<td>Elastosis</td>
<td>77%</td>
<td>58%</td>
<td>10.3%</td>
</tr>
<tr>
<td>Eccrine glands</td>
<td>77%</td>
<td>58%</td>
<td>16.61%</td>
</tr>
<tr>
<td>Lymphocyte infiltrate</td>
<td>70%</td>
<td>47%</td>
<td>18.27%</td>
</tr>
<tr>
<td>Perineural invasion</td>
<td>33%</td>
<td>17%</td>
<td>0%</td>
</tr>
<tr>
<td>Lesion with fibrosis</td>
<td>80%</td>
<td>64%</td>
<td>10.3%</td>
</tr>
<tr>
<td>Micronodules</td>
<td>43%</td>
<td>37%</td>
<td>7.64%</td>
</tr>
<tr>
<td>Necrosis</td>
<td>67%</td>
<td>31%</td>
<td>3.99%</td>
</tr>
<tr>
<td>N-P-C, elastosis</td>
<td>83%</td>
<td>73%</td>
<td>5.98%</td>
</tr>
<tr>
<td>N-P-C, fibrosis</td>
<td>77%</td>
<td>64%</td>
<td>5.98%</td>
</tr>
<tr>
<td>N-P-C, infiltrate</td>
<td>77%</td>
<td>59%</td>
<td>18.27%</td>
</tr>
<tr>
<td>N-P-C, pilosebaceous annexa</td>
<td>67%</td>
<td>42%</td>
<td>7.97%</td>
</tr>
<tr>
<td>N-P-C, trabeculae</td>
<td>71%</td>
<td>57%</td>
<td>2.33%</td>
</tr>
<tr>
<td>Morpheaform pattern</td>
<td>83%</td>
<td>77%</td>
<td>2.99%</td>
</tr>
<tr>
<td>Rod trabeculae</td>
<td>83%</td>
<td>47%</td>
<td>12.96%</td>
</tr>
<tr>
<td>Ulceration</td>
<td>60%</td>
<td>60%</td>
<td>1.99%</td>
</tr>
<tr>
<td>Sanguineous vessel</td>
<td>70%</td>
<td>64%</td>
<td>30.23%</td>
</tr>
</tbody>
</table>
new retrieval models for heterogeneous document collections containing medical images as well as text.

The ImageCLEFmed challenges researchers in image retrieval on medical applications to find the correct answer for a set of queries in a given collection of medical images. When registration opens, organizers deliver a collection of heterogeneous medical images composed of radiographies, computer tomographies, microscopy images, ultrasounds, among others. In 2007 the collection contains a set with more than 66,000 images of all modalities taken from real medical environments. Together with images is a set of textual annotations corresponding to medical descriptions in English, German and French languages. Some weeks after registration closes, a set of queries are delivered without the correct answer, and researchers have to use their systems to obtain the most relevant results. Each group finds results according to the used model and sends them as the approximate answer to the challenge. The organizers collect all answers and evaluate them according to a predefined gold standard, ranking groups that have participated.

### 6.5.2 Participation

The general topic of this research is Medical Image Retrieval, specially focused on histopathology images. The proposed framework to calculate image similarity can be easily adapted to work with different image retrieval tasks. We considered to participate in this competition applying our image retrieval model based on low-level features to compare the effectiveness of this approach with strategies used by other research groups.

The used methods in this competition were adapted from those introduced in Chapter 4. Mainly, all images in the CLEF database were processed using low-level feature extraction. To retrieve relevant images, low-level metrics were used and they were combined using a *Cross Category Feature Importance* scheme as is described in [7]. In addition, an index to identify related images by modality was introduced using machine learning techniques. This index allows to filter non-relevant images in order to calculate the visual similarity only between images with the same modality. ImageCLEFmed 2007 edition had three query topics, based on visual information, textual information and a mixture. Our participation was only based on visual information, together with other five research groups.

Results of the competition are reported in [37]. To compare image retrieval systems, the ImageCLEFmed uses different evaluation measures of the system response, for example, *Mean Average Precision* (MAP), number of relevant images retrieved, precision at 20, 50 and 100 images, among others. In the track using only visual information, this work took the fourth position among six groups. Results of this challenge are presented in Table 6.7.
Table 6.7: ImageCLEFmed results

ImageCLEFmed results in the category based on visual information. The first column shows research group name, $N_R$ is the number of relevant images retrieved and at last the $MAP$ measure.

<table>
<thead>
<tr>
<th>Group</th>
<th>$N_R$</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aachen University, Germany</td>
<td>1613</td>
<td>0.2328</td>
</tr>
<tr>
<td>Concordia University, Canada</td>
<td>630</td>
<td>0.0333</td>
</tr>
<tr>
<td>Spain University Consortium</td>
<td>532</td>
<td>0.0186</td>
</tr>
<tr>
<td>Universidad Nacional de Colombia</td>
<td>644</td>
<td>0.0082</td>
</tr>
<tr>
<td>Dokuz Eylul University, Turkey</td>
<td>292</td>
<td>0.0041</td>
</tr>
<tr>
<td>University of Geneva, Switzerland</td>
<td>277</td>
<td>0.0028</td>
</tr>
</tbody>
</table>
Part III

System Efficiency
Chapter 7

Search Algorithms

Chapters 4 and 5 have introduced two similarity strategies to identify the most relevant images in the database. Those strategies are based on a set of features and metrics that allow to induce a representation space in which distances between images can be calculated. A two-dimensional projection of those spaces can be viewed in Figures 4.3 and 5.5. The first one shows all images in a space in which one object is close to another mainly because they are similar in visual terms. The second Figure shows all images in a space in which one object is close to another because they share similar concepts.

The key aspect in such spaces is that a similarity function is available to calculate how close is an image to another one. So the problem of searching the most similar images is in fact the problem of identifying the nearest neighbors in the representation space. In general, this problem is defined as to process a database of $n$ objects so that given a query object, one can effectively determine its nearest neighbors in the database. This problem is common to many computer science problems such as information retrieval, machine learning and data mining, as well as different applications such as recommendation systems, spelling corrections and MPEG compression, among others. Through this Chapter the main features of a nearest neighbor search algorithm are introduced, as well as some aspects of the sequential scan and a metric indexing method.

7.1 Introduction

Similarity measures induce an order relationship in image contents in a database. When the system is queried, the actual problem is to identify the subset of more similar images. Therefore, we need to consider the problem of accessing a set of images using a similarity measure. Formally, the search space is an image representation space $\mathcal{I}$, induced by any of the feature extraction methods. Let $\mathcal{D}$ be the image database in which elements $i_j$ are images actually stored in the system. Let $\sigma$ be a similarity function defined over the search space. The problem of retrieving the most similar images from the database is reduced
to the problem of identifying the nearest neighbors [35]. The following is the outline of these definitions:

- **Search space**: image representation space $\mathcal{I}$
- **Similarity function**: $\sigma : \mathcal{I} \times \mathcal{I} \rightarrow \mathbb{R}$
- **Image database**: the input dataset $\mathcal{D} = \{i_1, ..., i_n\} \subset \mathcal{I}$
- **Query object**: given by the user $q \in \mathcal{I}$
- **Task**: find $\text{argmax}_{i_j \in \mathcal{D}} \sigma(i_j, q)$

A nearest neighbor search algorithm has to conduct the previously defined task efficiently. The main problem of the required task is that the input dataset $\mathcal{D}$, in our case the image database, may be very large. In real applications the amount of objects in $\mathcal{D}$ is permanently growing, and there can be many users searching for something concurrently. Different tree-based data structures have been proposed to speed up the search task when a vector representation is at hand, such as R-tree, k-d tree and quadtree, among others. Some comprehensive surveys on the problem of search in metric spaces can be found in [13, 23]. In general, indexing methods in metric spaces take advantage of the triangular inequality property of a metric function.

Through this work, the terms *similarity* and *metric* have been used indistinctly to define a function that compares the content of two images. Nevertheless, it is important to point out that they are, in fact, different. Specially in this context, a metric function is useful to define a metric space as follows:

A metric space is a tuple $(\mathcal{I}, d)$, with $\mathcal{I}$ being the search space and $d$ being a metric $d : \mathcal{I} \times \mathcal{I} \rightarrow \mathbb{R}$, such that $d$ satisfies:

1. **Non-negativity**: $\forall x, y \in \mathcal{I} : \; d(x, y) \geq 0$
2. **Identity** $d(x, y) = 0$, if and only if $x = y$
3. **Symmetry** $\forall x, y \in \mathcal{I} : \; d(x, y) = d(y, x)$
4. **Triangle inequality**: $\forall x, y, z \in \mathcal{I} : \; d(x, z) \leq d(x, y) + d(y, z)$

The main difference between a similarity and a metric is that similarity spaces generally does not satisfies the triangle inequality. Although similarity and metric are different things, sometimes, one can change a function in order to transform one into the other. For example, the Tanimoto Coefficient introduced in Section 5.3 to compare semantic features, is originally defined as a similarity function. However, it is easy to change its behavior to a metric function, knowing the basic properties of a metric. In this work, functions to compare low-level and semantic features have been enabled as metrics.
Inputs: QueryImage \( x \), SimilarityMeasure \( sm \), ImageDatabase \( d \)
Output: SimilarityResults \( sr \)

Begin
    foreach \( y \) in \( d \)
    
    score = \( sm.\text{evaluate}(x,y) \)
    \( sr.\text{add}( \text{score} ) \)
    
end
sr.\text{sort()}

End

Algorithm 1: Simple image retrieval algorithm

7.2 Sequential Scan

The sequential scan is a simple algorithm that compares all images in the database with the example image. The general definition of the sequential scan is shown in Algorithm 1, which receives as input a query image \( x \in I \), the similarity measure \( \sigma \) and the image database \( D \). This algorithm returns as output the distance measure between the image \( x \) and all images into the database, sorted by closeness. This algorithm is regarded by two run-time characteristics that make it computationally expensive: the number of required similarity computations and the sorting cost, both depending on the size of the database.

The sequential scan is widely applied in experimental settings, to test mainly the precision of similarity measures more than the computational performance of the database. It is actually not usable in real scenarios, because large amounts of images in the database will generate prohibitive response times.

Suppose that a system is required in a hospital to store 100,000 images and to provide access to 10 users. The system is configured in some hardware platform with a unique processor and following a sequential scanning approach, i.e. comparing the query object with all images in the database. In addition, the system has a metric function that takes 10 milliseconds calculating the similarity between two images. In the worst case, all users perform a similarity search concurrently, and then, the user that has to wait more time, may remain seated approximately \( 10,000 \text{ sec} \approx 166 \text{ min} \approx 2.7 \text{ hours} \) before getting its results. This is a very useless system even if the hardware platform is improved, and regarding that the number of images and users in the example is still very small.

7.3 Metric indexing

The purpose of an indexing method is to avoid the evaluation of the metric on many non-relevant images as possible. In particular, an algorithm to search the nearest neighbors may help to avoid the similarity computation between the query image and non-relevant images. A special requirement in the proposed
system is that this search method works in metric spaces, and does not require vector representation of the data.

Several algorithms have been developed to perform the indexing task on vector spaces, which take advantage of geometric and coordinate information (e.g. KD-trees, R-trees). These methods work under the assumption that the Euclidean distance is enough to represent the similarities between objects [13]. However, in this work, the proposed similarity measures interpret features as objects instead of as vectors. For example, the Sobel features may be processed as vectors in $\mathbb{R}^{512}$. However, since they are histograms, it was experimentally found that the Jensen-Shannon Divergence is better to evaluate a meaningful similarity between them, as was shown in Subsection 6.3.1. Therefore, an indexing method that takes advantage of the similarity function instead of the space structure is better for this application.

### 7.3.1 LAESA Algorithm

The algorithm used in this system is the Linear Approximating and Eliminating Algorithm (LAESA) [32], which is based on the triangle-inequality property to eliminate irrelevant data. Figure 7.1 shows the basic idea behind the LAESA algorithm: to avoid the computation of distances between the stored element $u$ and the query object $q$, using distances to another reference object in the metric space. The reference object $p$ is in a small subset of the database $D$ called the base prototypes $B$. Reference objects in $B$ are called index pivots or base prototypes.

The algorithm estimates how close the objects $q$ and $u$ are, instead of calculate the actual similarity value. This estimation is based on a lower bound for the distance between the stored image and the query image, which is calculated as:

$$g_u(q) = \max_{p \in B} \{|d(p, u) - d(p, q)|\}$$  \hspace{1cm} (7.1)

The lower bound $g_u$, calculated with respect to a stored image $u \in D$, is
useful to predict whether \( u \) is relevant (very close to the query) or irrelevant (very far from the query). The lower bound is also based on some precalculated distances \( d(p, u) \forall p \in B \), i.e. the distances between all stored images and the base prototypes. This is the index structure that will be described in the next Subsection.

The search procedure of the LAESA algorithm uses a branch and bound strategy to discard non-relevant images with two main steps: approximation and elimination. The approximation step selects a new image closer to the query, while the elimination step decides whether one image is near enough to be relevant or not as is presented in Algorithm 2. Both steps use the information of the lower bound.

LAESA is based on the Approximating and Eliminating Algorithm (AESA). LAESA was proposed by the observation that AESA was quadratic in time and space. The new algorithm introduced some modifications that make it to be linear in both, time and space, as the \( L \) in the name indicates. This property has been analytically evaluated and different experimental results have demonstrated it [33].

7.3.2 A Modified LAESA algorithm

Although LAESA was originally designed to search the nearest neighbor, it is used to solve \( k \)-nearest neighbor queries by being stopped when approximately \( k \) images have not been discarded yet, as was proposed in [34]. This heuristic has some problems:

- It is highly dependant on the order in which the images are evaluated.
- When the algorithm is stopped the images that have not been evaluated yet are taken as relevant though their distances to the query are unknown.
- The error rate increases with \( k \).

In this work, a modified LAESA algorithm has been proposed, which does not discard elements based on the distance to the current nearest neighbor but based on an estimation of the distance to the \( k \)-nearest neighbor. It can be proven that if we have a perfect estimation of this value the retrieved elements will be exactly the \( k \)-nearest neighbors. Let \( e \) be the value for the estimated \( k \)-nearest neighbor distance. At each iteration of the algorithm, lower bounds are computed for all the elements in the database, choosing randomly a pivot from \( B \) and those with a lower bound greater than \( e \) are pruned. If the number of retrieved elements is far from \( k \), \( e \) is adjusted by an approximation factor \( f \) and the process is repeated. Finally the results are sorted and retrieved. Algorithm 2 presents pseudo code for this procedure.

For estimating the distance to the \( k \)-nearest neighbor \( e \) a random sample \( S \) of images from the database was taken. Using sequential scan the distances to the \( k \)-nearest neighbor for each image in \( S \) was obtained and the 90th percentile of these values was chosen. The approximation factor \( f \) was determined by
Algorithm 2: The modified LAESA algorithm scheme

exhaustive evaluation of values in (0, 1), such that the precision of the system was maximized while keeping the querying time low.

The complexity of the modified algorithm is also linear in the size of the database. Notice that, there are no differences in terms of complexity between the sequential scan, the LAESA algorithm and the modified LAESA algorithm. In general, the time saving of the LAESA and its modified version is due mainly because of, in most of the cases, algorithms do not calculate the actual distance between images. Instead, they proceed with a pruning process using an estimate bound based on distances to the prototypes. Therefore, since the amount of time required to search a set of objects depends on the metric cost, it is important to experimentally evaluate the behavior of the algorithms, mainly because all algorithms are linear in run time, from an analytical point of view.

On the other hand, since the proposed algorithm uses an estimate value for the distance to the $k$-nearest neighbor, it is clear that in some cases the algorithm does not locate exactly the $k$-nearest neighbors. This lack of precision has been evaluated as is shown in Section 8.2, to determine a trade off between speed and precision.
7.3.3 Index structure

One important aspect of LAESA is the index structure, that is to say, the table with previously calculated distances between the base prototypes and all images in the database. The main idea with the selection of these base prototypes is to find representative points in the search space. Moreno-Seco et al. [34] suggest to select a set of base prototypes that are maximally separated, because it has been shown that the number of distance computations is reduced in the following steps. In this work, the base prototypes were selected by the calculation of the medoid in each of the 18 semantic groups identified by pathologists.

To compute the medoid of each semantic group, the distance between images is used to determine which image has the minimum distance sum with respect to the others in the same group. Let \( D_s = \{i_1,...,i_k\} \subset D \) be the subset of \( k \) images in the database that correspond to the concept \( s \). The medoid of the subset \( D_s \) is defined as follows:

\[
b_s = \arg\min_{i \in D_s} \sum_{j \in D_s} \sigma(i,j)
\]

Where \( \sigma \) is a metric function between two images. Since the database has 18 concepts, the base prototype set \( B \) contains 18 pivots to build up the index structure. Prototypes \( b_s \) are shown in Figures 4.3 and 5.5 using multidimensional scaling in the database of 1,502 images. In the first case, the low-level metric has been used. In the second case, the Figure shows the semantic representation with its base prototypes.

The construction of the index consists in calculate the distance between all images in the database and the 18 selected base prototypes. In runtime, the LAESA algorithm scans the index instead of calculate the actual similarity function, under the assumption that the computational cost of the metric is higher than evaluate some arithmetic operations on the index.
Chapter 8

Efficiency Evaluation

This evaluation aims to analyze the general system performance comparing the sequential scanning algorithm vs the metric indexing. The measure for system efficiency is based on the response time taken to retrieve a result set. This system was implemented in the Java programming language, and all experiments were run under GNU Linux, in a standard PC with an Intel Pentium D processor and 1 GB of RAM memory.

The feature extraction phase is not included in this performance evaluation, since these computations are made only when images are first added to the database. Once images are archived, features are also stored to use them later in a retrieval algorithm, and the time spent in the feature extraction phase does not impact the response time in the retrieval phase.

Reported times are in milliseconds and correspond only to the time required to execute the retrieval algorithm. An efficiency experiment is comprised of 30 random queries, in each query the run time is measured and then averaged with others to get a mean value. First, experiments to evaluate the sequential scanning algorithm were made, in which every available similarity measures were tested to identify their general computational load. Then the metric indexing was evaluated using the semantic metric and the low-level features combination to compare its performance against the sequential scan.

Since the metric indexing retrieves a subset of images in the database, the precision is also measured to analyze what is the error introduced by an approximate search. The precision analysis compares if the output of the sequential scan is the same as the metric indexing algorithm.

8.1 Performance Results

The required time for identifying a set of similar images using sequential scan is $O(n)$, with $n$ the number of images in the database. This is corroborated by plots in Figure 8.1: while images in the database increases linearly, the response time also increases linearly. However, an important aspect in this plot
is that the linear combination of low-level features requires more time than both, individual low-level features and semantic features. Therefore, this plot shows that the retrieval algorithm has a computational load that depends on the cost of the underlying metric.

This is consistent with the fact that the combination of low-level features computes many distances between many histograms to finally deliver similarity results. Although individual features require less time than a combination, they are in fact more expensive than the semantic metric, because histograms are represented in a higher dimensional space. The semantic vector belongs to a relatively low-dimensional space, and the semantic metric is based on the computation of some dot products, so it is a fast-to-calculate metric. Figure 8.1 shows that, in a database with 1,500 images, the sequential scan takes less than 10 ms to deliver the results. However, a linear growing is also the trend. Table 8.1 shows the average response times of the sequential scan algorithm when the database has 1,502 images, using different underlying metrics.

In general, sequential scan ranks all images in the database, but the user only needs the subset of the more similar ones. In the following subsections the parameter $k$ is used to evaluate the number of required images in the result list. So, if the user needs to find $k$ relevant images, the system does not need to calculate the similarity between all images in the database. Experiments in the following subsections use a database with 1,502 images in order to evaluate the response times with different values of the $k$ parameter.
Table 8.1: Average response times

<table>
<thead>
<tr>
<th>Metric</th>
<th>Average time</th>
<th>Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metric based on low-level feature combination</td>
<td>610 ms</td>
<td>23 ms</td>
</tr>
<tr>
<td>Metrics based on individual low-level features</td>
<td>172 ms</td>
<td>14 ms</td>
</tr>
<tr>
<td>Semantic metric</td>
<td>9.4 ms</td>
<td>1.3 ms</td>
</tr>
</tbody>
</table>

8.1.1 Low-level metric performance

Figure 8.2 shows the response time trend of the indexing method, using as underlying metric the low-level feature combination. According to the Table 8.1 the sequential scan takes 610 ms in average to retrieve the most similar results using the low-level metric in a database of 1,502 images. Results show an important performance increase when the system is using the metric indexing. In particular, when 50 images are needed as result, the LAESA algorithm takes 57 ms in average, demonstrating a performance improvement of 90%. When \( k \) takes a value of 100, the 88% of time is saved. To retrieve datasets under 250 images, this improvement still remains up to 80%.

According to the plot, the metric indexing outperforms the sequential scan with a 60% of improvement, in average. It is important to point out that this average improvement was calculated through different \( k \) values. In this
Table 8.2: Featured response times of the LAESA algorithm

Featured response times of the LAESA algorithm using the low-level metric. $k$ is the number of required images. Average time is the required time to get $k$ results, together with the Standard Deviation. The Improvement column presents the performance outperforming regarding sequential scan.

<table>
<thead>
<tr>
<th>$k$</th>
<th>Average time</th>
<th>Std Deviation</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>52.7 ms</td>
<td>6.2 ms</td>
<td>90%</td>
</tr>
<tr>
<td>100</td>
<td>76.4 ms</td>
<td>8.4 ms</td>
<td>88%</td>
</tr>
<tr>
<td>150</td>
<td>79.0 ms</td>
<td>7.3 ms</td>
<td>87%</td>
</tr>
<tr>
<td>200</td>
<td>116.7 ms</td>
<td>12.1 ms</td>
<td>81%</td>
</tr>
</tbody>
</table>

experimentation the number of retrieved objects goes to a maximum value of 1,000 images. However, the number of images to deliver for real users, should range between 50 and 200 similar images. For fixed $k$ values the Table 8.2 shows the most featured response times, taking into account that a user needs a manageable result set.

Figure 8.2 also shows how the LAESA algorithm requires more time while the number of retrieved images grows. This is because of the number of filtered images using the index structure is each time smaller. Results on the low-level metric shows that the metric indexing effectively avoids many distance computations, and this difference is better shown here because of the high cost of the low-level metric.

### 8.1.2 Semantic metric performance

As it was mentioned before, the semantic metric is a fast computation because semantic features are in a low-dimensional space and the similarity function is based on a few dot products. This situation makes that a sequential scan takes a very low time computing similarities and sorting the whole database. However, the metric indexing shows again a performance improvement with regard to sequential scan. Figure 8.3 presents the response time tendency when search algorithms use the semantic metric as the underlying similarity function. Retrieving a subset of 50 images takes near to 9.4 ms with the sequential scan while the metric indexing does the same only in 3.8 ms, 60% faster. For a subset of 100 images, the performance improvement is about 42% and for datasets under 250 images approximately the 40% of time is saved. Featured response times are shown in Table 8.3.

In general, the metric indexing is 21% faster than the sequential scan, regarding that this value has been calculated with $k$ values from 50 to 1,000. When the $k$ value is near to the database size it is logical that response times tends to have the same behavior as the sequential scan. However, in a real environment the system delivers the subset of most similar results that allows the user to explore interesting images. Compared with the obtained improvement in the low-level case, this is a lower improvement because the semantic metric is
Table 8.3: Featured response times of the LAESA algorithm using the semantic metric

Featured response times of the LAESA algorithm using the semantic metric. $k$ is the number of required images, Average time is the required time to get $k$ results, together with the Standard Deviation. The Improvement column presents the performance outperforming with respect to sequential scan.

<table>
<thead>
<tr>
<th>$k$</th>
<th>Average time</th>
<th>Std Deviation</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>3.83 ms</td>
<td>0.4 ms</td>
<td>59%</td>
</tr>
<tr>
<td>100</td>
<td>4.97 ms</td>
<td>0.5 ms</td>
<td>42%</td>
</tr>
<tr>
<td>150</td>
<td>4.57 ms</td>
<td>0.9 ms</td>
<td>48%</td>
</tr>
<tr>
<td>200</td>
<td>5.37 ms</td>
<td>0.8 ms</td>
<td>39%</td>
</tr>
</tbody>
</table>
not as expensive as the low-level metric. However, the advantage of the metric indexing is also clear in the semantic metric case. In addition, using a system configuration that includes the semantic metric, the LAESA algorithm, and a powerful hardware platform, users may expect an effective and efficient Image Retrieval System.

8.2 Impact on Precision

The metric indexing helps to save time in identifying the $k$ most similar images, but an important question is if those images are really the $k$-nearest-neighbor images in the search space. Taking into account that the sequential scan retrieves exactly the $k$ most similar images, it makes sense to compare the metric-indexing precision with the sequential-scan precision. This experimentation aims to identify which of the nearest neighbors are avoided by the modified LAESA algorithm.

The experimental setup makes 30 runs using the sequential scan and the modified LAESA algorithm. In each run a list of the most similar results are delivered by both algorithms. The list retrieved by the sequential scan is herein used as the ground truth, since it exactly has the $k$-nearest neighbors according to the metric. We evaluate differences between both lists counting how many times the $n$th-nearest-neighbor was not included in the modified LAESA results on a list of $k$ results. In particular, this experimentation evaluates the case in which a user is looking for the most $k = 100$ similar images.

In the experimentation, the modified LAESA algorithm shows an average lack of precision of 9%. That means, when a user requires 100 similar images, there is expected to get the 91-nearest-neighbors $^1$ and other 9 images. An important aspect to analyze is which positions in the ranking are affected by this lack of precision. Figure 8.4 shows the expected lack of precision in groups of 5 positions in the ranking list. The error value is interpreted as the probability of getting a wrong nearest neighbor in the $n$-th position of the list. For example, the probability of getting a wrong image between the first position and the fifth position is 0.03%. On the other hand, the probability of get a wrong result between the 80th position and the 85th position is 0.87%, the higher among all. In general, those errors are smaller than 1%.

Figure 8.4 shows higher error probabilities for the most distance neighbors, i.e. the probability of getting an error in the first position is much smaller than the probability of getting an error in the 100th position. Hence, considering that the introduced lack of precision of the modified LAESA algorithm is very small and the most interesting results for the user will remain in the ranking list, it can be said that the impact of the metric indexing on the system effectiveness is not significant.

$^1$According to the metric
Figure 8.4: Average error in the raking list

*Average error in the raking list that is introduced by the modified LAESA algorithm. This plot shows an expected error per each 5 positions in the result list.*
Part IV

Summary
Chapter 9

Conclusions and Future Work

This research has presented the design, development and evaluation of a medical image database system. The proposed system exhibits some particular features that distinguish it from traditional image management systems: its architecture is multi-tier, it provides web access to image collections, and it allows effective and efficient content-based retrieval. The content-based-retrieval module provides a search-by-example capability, i.e. the user can retrieve images that are similar to a given reference image. This chapter presents the main conclusions of this work and the future work to be developed.

9.1 Discussion

In this research a system to retrieve histopathology images has been designed, implemented and evaluated. This system has an extensible architecture, allowing the implementation and use of many features, metrics and search algorithms. To face the problem of histopathology image retrieval, a model for semantic feature extraction has been proposed, based on machine learning algorithms. In addition, the problem of an efficient system has been approached with the use of a metric indexing technique. Results have shown the effectiveness and efficiency of the proposed model.

9.1.1 Global Features vs. Region Modeling

The low-level feature extraction process is completely based on global descriptors of the image content, i.e. no local features are modeled or identified such as regions or objects. Moreover, global histograms are reduced with meta-features to be used in the semantic mapper. The purpose of feature extraction is the identification of key characteristics that represent image contents, and many methods have been proposed such as region extraction [11] and hierarchical
representations [17]. However, the main problem is what the system does with the extracted features, because all of them represent visual perceptions instead of conceptual interpretations. The proposed approach in this work is an effort to take great advantage of low-level features by configuring machine learning algorithms in an effective way to produce semantic features. A detailed design and evaluation of such algorithms have been achieved, and the experimental results support this approach.

9.1.2 Automatic Annotation vs. Semantic Feature Extraction

The semantic feature extraction in this work is closer to an image annotation process. Image annotation has been previously used for image retrieval, and multiple binary-class SVM have also been used to generate keywords [16]. However, assigning deterministic keywords to an image makes that some information about visual appearance may be lost. Specifically, when the concept is represented as a keyword there are only two possibilities: the concept is present or absent, the concept cannot be partially present in the image. The proposed approach is able to keep certainty information about the decisions made by the classifiers in a semantic feature vector, allowing the representation of the presence degree of each concept. Furthermore, the proposed semantic metric generates a metric space in which conceptual distances can be calculated, obtaining a finer mechanism to compare images.

9.1.3 Spatial-based Indexing vs. Metric-based Indexing

Several algorithms have been developed to perform the indexing task on vector spaces, which take advantage of geometric and coordinate information (e.g. kd-trees, R-trees). These methods work under the assumption that the Euclidean distance is enough to represent similarities between objects [35]. However, in this work image features are histograms and it was experimentally found that other metrics such as Jensen-Shannon Divergence and Relative Bin Deviation give a better similarity notion. Therefore, an indexing method based on similarity functions was chosen and evaluated. Specifically, a modified LAESA algorithm has been used to take advantage of the triangle inequality in metric spaces. This algorithm can identify the most similar images in an efficient way, avoiding irrelevant computations. This is another important characteristic of the proposed system that makes it different from other CBIR systems reported in the literature.

9.2 System effectiveness

The content-based-retrieval module was evaluated using a collection of annotated histopathology images. Images were annotated by an expert establishing a ground truth for the system response. The effectiveness evaluation was made
in two different scenarios: using low-level features and using semantic features. In the initial evaluation with low-level features, results show that the combination of some features can approximate the differential diagnosis criteria used by pathologists until certain level.

This research presented a novel approach to represent visual histopathology knowledge, which is naturally included into a CBIR system. The approach allows to bridge the semantic gap between low-level visual features and human interpretation. This is accomplished by a semantic mapper based on SVM classifiers. This mapper defines a new semantic feature space in which a metric is used to calculate the conceptual similarity between images. The experimental results show that this approach can effectively model the histopathology knowledge, that is to say, images are automatically interpreted and compared as pathologists do. This strategy provides a semantic analysis of image contents, allowing a highly improved operation of the CBIR system, in terms of precision.

Content-based retrieval on general collections of medical images is still a challenging task that needs new and clever methods to implement useful and effective systems. The evaluation in the ImageCLEFmed presents a two-phase strategy to build an image category index and to rank relevant images. This strategy is completely based on low-level visual information and does not make use of textual data. In general, obtained results were as expected, not only because of the well known semantic gap, but because of the consistency in feature combination and filtering quality. The semantic gap in this case, was pointed by the fact that only low-level visual features were not enough to get an effective image retrieval. The feature combination shows a slightly improvement over the single visual features. And the filtering quality shows that the more effective the classification is, the better the retrieval results are.

9.3 System efficiency

A method to efficiently retrieve images was implemented. The LAESA algorithm, mainly used to classification tasks, was adapted to an image retrieval framework to find the $k$-nearest neighbors of a query image. This modified algorithm take advantage of properties in metric spaces in which images are represented, both in the low-level case and the semantic case. The experimental results show that this indexing scheme can make important improvements in terms of system efficiency. Since the interactive responses are a requirement for image retrieval, defined as response times lower than one second, the evaluation shows that this system is in essence interactive.

The algorithm shows important improvements in time responses with regard to the sequential scan, when the metric calculation is expensive. However, when the metric computation is also fast, the algorithm still shows better performance than the sequential scan, because it effectively prunes irrelevant images. The branch-and-bound strategy in this algorithm, based on the triangular inequality, has been useful in feature metric spaces by the use of semantic pivots. In both situations, low-level metric and semantic metric, the medoids of each conceptual
group were identified and used as base prototypes to calculate lower bounds in order to prune non-relevant images.

In addition, this modified algorithm has shown a very low impact in the precision of the retrieved results. When identifying the 100-nearest neighbors, it got a general error up to 9%. Further experimentation showed that the introduced lack of precision has a higher probability in the most distance neighbors i.e. the error is not in the top results. For specific positions in the result list, the error probability is always lesser than 1%, with a minimum value of 0.03% and a maximum value of 0.87%. In that sense, this algorithm has an impact in the retrieval precision that is not significant.

9.4 Future work

Since this work has been supported by a global visual description of image contents, the future work includes other visual representations with specific structure, i.e. object recognition or hierarchical representations, guided by the work in the image interpretation group of the BioIngenium lab. Furthermore, another approach will be followed with textual annotations to complement structured visual representations. Since textual annotations are not always available and they are not enough descriptions of visual image contents, the visual feature extraction is an important approach. But, when they are available, even with some few keywords, they provide important semantic clues about image context. Therefore, the fusion of textual and visual information will be an important part of the future work.

The design of a more general model to increase medical-image retrieval precision in general-medical image collections will be also studied. This is motivated by the fact that in hospitals and health centers image databases archive lots of images of different modalities. The ImageCLEFmed evaluation allows the development of an initial frame to face this problem, but more effective approaches have to be modeled. On the other hand, the design of powerful retrieval algorithms faster than the sequential scan will be considered.
Figure 9.1: Use case diagram for the prototype system

Figure 9.2: Entity-Relationship diagram for the system core platform
Bibliography


