VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

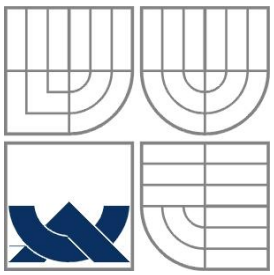
MPEG-7 MULTIMEDIA METADATA MANEGMENT

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

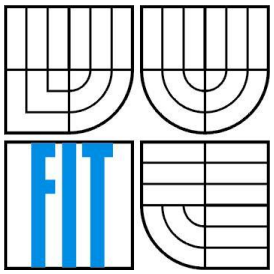
AUTOR PRÁCE
AUTHOR

ANTONIO CASERO PALMERO

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

NÁZEV DIPLOMOVÉ PRÁCE ČESKY

NÁZEV DIPLOMOVÉ PRÁCE ANGLICKY

DIPLOMOVÁ PRÁCE
MASTER‘S THESIS

AUTOR PRÁCE
AUTHOR

ANTONIO CASERO PALMERO

VEDOUCÍ PRÁCE
SUPERVISOR

ING. PETR CHMELA CHMELAŘ

BRNO 2007

Abstract

This MSc. Thesis deals with multimedia databases and it is focused on analysis of pictures with MPEG7 using Oracle InterMedia. Also, we create a model to perform a classification and prediction of images introduced in the database. This is useful to know and classify of a huge database, and later query the database for images we want to see.

I worked with the eXperimentation Model (XM) innovative tool to analyze pictures, creating a MPEG7 documents with the information of color and texture of the picture. These are stored in the database and processed like XML documents.

Also the created application has been developed with the aim of knowing and testing the last versions of interMedia technology provided by Oracle.

At last this application uses the coefficients of the images to create a classifier model that can be used by the program RapidMiner. We can use this tool to realize predictions about the images description.

Keywords

MPEG-7, XML, Color Descriptors, Texture Descriptors, XMLType, Database, SQL, Oracle InterMedia, UML, Classification, Crossvalidation, Java.

Citation

CASERO PALMERO, Antonio. MPEG-7 Multimedia Metadata Management. Master Thesis. Brno University of Technology, Brno, 2008.

MPEG-7 MULTIMEDIA METADATA MANAGEMENT

Declaration

I declare that I have solved this Master Thesis by myself.
I have mentioned all information resources used in the thesis.

.....
Name
Date

Acknowledgement

I would like to thank my supervisor **Ing. Petr Chmelař**, for his great supervision, help and patience with this MSc. Thesis.

Thanks to everybody in Czech Republic who made this experience unforgettable for me, specially to my new friends which made me enjoy every minute of this Erasmus year. Thanks also to my friends from Spain which always let me know they didn't forget me.

Finally, I will always be grateful to my parents, **Antonio Casero** y **Natalia Palmero** for their great work developed in the difficult field of the real life, during the last 26 years, with me and my big family: **Emilia, Maria, Ana, Carmen, Juan, Henar, Rocio y Pablo**.

To all of them. Thanks.
Antonio Casero Palmero.
Brno, 2007-2008

CONTENT INDEX

Chapter 1. Introduction and Objectives 8

1.1 Goals 8
1.2 Structure of the memory 9
1.3 MPEG-7 Introduction 10
1.4 Others MPEG 11
1.5 MPEG-7 Objectives 12
1.6 Possible Applications 13
1.7 Main elements of MPEG-7 Standard 14
1.8 Parts of MPEG-7 15
1.8.1 Visual Descriptor 16
1.8.2 Color Descriptors 17
1.8.3 Color Space Descriptor 17
1.8.4 Color Quantization Descriptor 17
1.8.5 Dominant Color Descriptor 17
1.8.6 Scalable Color Descriptor 18
1.8.7 Color Layout Descriptor 18
1.8.8 Color Structure Descriptor 19
1.8.9 GOF & GOP Color Descriptor 19
1.9 MPEG-21 20
1.9.1 What is MPEG-21? 20
1.10 MPEG-7 and XML 21
1.10.1 XML and Oracle 21
1.10.2 XMLType 22

Chapter 2. Similarity Search 24

2.1 Introduction 24
2.1.1 Mathematical Theory 24
2.2 Distance Measure 25
2.2.1 Mikowsky Distances 25
2.2.2 Quadratic for Distance 25
2.2.3 Edit Distance 26

2.3 Similarity Queries 26
2.3.1 Range Query 26
2.3.2 Nearest Neighbor Query 27

Chapter 3. Classification and Prediction 29

3.1 Introduction 29
3.2 Definitions 29
3.2.1 How the classification work? 29
3.2.2 How is Prediction different from Classification? 30
3.3 Classification by Decision Tree Induction 30
3.4 Bayesian Classification 31
3.5 Naïve Bayesian Classification 31
3.6 Support Vector Machine (SVM) 31
3.6.1 The case when the data are linearly separable 32
3.6.2 The case when the data are linearly inseparable 33
3.7 Accuracy and error measure 33
3.7.1 Classifier Accuracy measure 34
3.8 Evaluating the accuracy of a Classifier 35
3.8.1 Cossvalidation 35

Chapter 4. Oracle InterMedia 37

4.1 Introduction 37
4.2 Procedure to insert Images and Characteristics 38
4.3 Retrieve of Images 39
4.4 Metadata Extraction 40
4.5 Image Processing 40
4.6 InterMedia and Images 40
4.6.1 Digitized Images 40
4.6.2 Image Component 41
4.7 Image support for Java 42
4.8 Example of InterMedia & Java 44
4.8.1 Insert an Image 44
4.8.2 Download a image from the database 47

Chapter 5. Used Technologies 49

5.1 Java 49
5.1.1 Java Introduction 49
5.1.2 Brief History of Java 50
5.1.3 Main Characteristics 50
5.1.4 Advantages 52
5.1.5 Disadvantages 54
5.2 RapidMiner 54

Chapter 6. Analysis and Design 57

6.1 Architecture 57
6.2 Use Cases 58
6.3 Database Design 67
6.3.1 Description of Tables 69
6.4 Classes Description 72

Chapter 7. The Experiments 82

7.1 Experiment of Similarity Search 82
7.1.1 Finding the twin images 82
7.1.2 Finding the similarity image 83
7.2 Experiment of Classification 85

Chapter 8. Conclusions 88

8.1 Introduction 88
8.2 Success of achieving the objectives 88
8.3 Surged Inconveniences 89
8.4 Conclusion of MPEG-7 89
8.5 Future Work 90

References 92

Chapter 9. Appendixes & Attachments 95

9.1 User Manual 95
9.2 Select Images 96
9.2.1 Analysis of Images 97
9.2.2 Search on the Database 98
9.2.3 The experiments 100
9.3 RapidMiner Manual 104
9.3.1 General Information 104
9.3.2 Setting up a Process 104
9.3.3 Validating your process definition 108
9.3.4 Running your process 108
9.3.5 Evaluating the results 108
9.4 CD Content 110

Figure Index

Figure 1.1 - MPEG-7 Description from [9] 11
Figure 1.2 - DDL Schema from [9] 15
Figure 1.3 - Overview of a visual descriptor tools [4] 16
Figure 1.4 - Table of pros and cons of XML storage from [11]. 22
Figure 2.1 - Example of Range Query [17] 27
Figure 3.1 - Sample of decision tree induction [16]. 30
Figure 3.2 - Data linearly Separable (a) (b) [16] 32
Figure 3.3 - RBF function [13] 33
Figure 3.4 - Confusion Matrix 34
Figure 4.1 - Oracle Multimedia Architecture [11] 38
Figure 4.2 - ORDImage Type Structure [12] 41
Figure 4.3 - Relation between ORDSYS.ORDImage and Java ORDImage Object [12] 43
Figure 4.4 - Insertion a new Row 44
Figure 4.5 - Selection a new row for up date 45
Figure 4.6 - Creation of proxies linked to new Images 45
Figure 4.7 - Upload Image File to proxy 45
Figure 4.8 - Signature creation and Thumbnail creation 46
Figure 4.9 - Last step to insert a new picture 46
Figure 4.10 - Download a image from the database 47
Figure 5.1 - Compilation in Java [5] 51
Figure 6.1 - Two-tier Architecture 57
Figure 6.2 - Three-tier architecture 57
Figure 6.3 - Uses Case 58
Figure 6.4 - UC Select Image 59
Figure 6.5 - UC02 – Analysis of Image 60
Figure 6.6 - UC03 – Insert into the database 61
Figure 6.7 - UC04 – Search database 62
Figure 6.8 - UC05 – See the XML image 63
Figure 6.9 - UC06 – Similarity search Images 64
Figure 6.10 - UC07 – Create a training data 65
Figure 6.11 - UC08 – Insert concepts 66
Figure 6.12 - Compact view of the Database 67

Figure 6.13 - Design of the Database 68
Figure 7.1 - Example of experiment, twin Images 82
Figure 7.2 - Twin Images 83
Figure 7.3 - Similar Images 83
Figure 7.4 - Example of experiment, similar Images 84
Figure 7.5 - First tab of Example, Experiment two. 85
Figure 7.6 - Second tab of the Example, experiment two. 85
Figure 7.7 - Using Crossvalidation in RapidMiner 86
Figure 7.8 - Confusion Matrix 86
Figure 9.1 - The main frame 95
Figure 9.2 - Choosing a Image 96
Figure 9.3 - Load Successful 97
Figure 9.4 - MPEG-7 Window 97
Figure 9.5 - The images in the database 99
Figure 9.6 - Preview of the Image 99
Figure 9.7 - Experiments menu. 100
Figure 9.8 - Experiment 1 Interface 100
Figure 9.9 - Image Preview 101
Figure 9.10 - The first tab of the experiment two 102
Figure 9.11 - Second tab of experiment two. 103
Figure 9.12 - Tree View 105
Figure 9.13 - Attribute Editor 107
Figure 9.14 - Results 109

Chapter 1. Introduction and Objectives

This chapter explains the project goals and a little description of each chapter and introduction about the main topic of this project, MPEG-7.

1.1 Goals

This work is about the management, storage and classification of images. To do this, I created an application with an intuitive and easy interface, where all users can use it without knowledge of the theory concepts.

At the beginning of this Master Thesis I met the following goals:

- The study of MPEG-7 focuses on the visual part: To be able to analyze images with eXperimentation Model (XM), and do an application that can do this.

- Study support of Oracle Database for XML data: The application analyzes images and inserts them into Oracle database, to do this I have studied the next fields:
  - Study of XML: The application has to parser XML documents. The storage in database is possible with the type XMLType.
  - Oracle InterMedia: Study the functions and operations of this technology. The application has to storage, download and manage images.
  - Study of Similarity search, to use the algorithms for measure the distance between different images and be able to compare them.

- Classification of Images:
  - Study of classification and prediction methods and how do they work.
  - The application has to create classifier model using RapidMiner program.

- Explain all the theory concepts in the memory and document the experiments and the process of development of the application.

1.2 Structure of the memory

- *Introduction*: In the Chapter-1 is about the goals of project, summary of memory, and an introduction of MPEG-7.

- *Theoretic Background*:
  - The Chapter-1 is about MPEG-7 and the others MPEG.
  - The Chapter-2 is about similarity search and mathematical algorithm.
  - The Chapter-3 is about Classification and prediction.

- *Used Technologies* :
  - The Capter-1 contain information about XML .
  - The Chapter-4 is about Oracle, so it is very important in this project so it have a one chapter for himself. You can see examples and theoretic concepts.
  - The Chapter-5 is about the used technologies like Java and RapidMiner, explaining the history and functions of these technologies.

- *Analysis and Design*: Chapter-6 contains this information, Uses Case, Design of database, descriptions of classes.

- *Experiments*: In the Chapter-7 is the documentation about two experiments. The first one is similarity search using MPEG-7 and Oracle InterMedia. The second one is about classification and the use of RapidMiner

- *Conclusion and References*: In the chapter-8 are the conclusions of my work, the future works and references used in this thesis.

- *Appendices*: At the end of this memory, chapter-9. You can read the user manual of my application and manual of RapidMiner (where I explain how use it in my application) and the CD content.

1.3 MPEG-7 Introduction

MPEG-7 is an ISO/IEC standard developed by MPEG (Moving Picture Experts Group), the committee that also developed the successful standards known as MPEG-1 (1992) and MPEG-2 (1994), and the MPEG-4 standard (Version 1 in 1998, and version 2 in 1999). The MPEG-1 and MPEG-2 standards have enabled the production of widely adopted commercial products, such as Video CD, MP3, digital audio broadcasting (DAB), DVD, digital television (DVB and ATSC), and many video-on-demand trials and commercial services. MPEG-4 is the first real multimedia representation standard, allowing interactivity and a combination of natural and synthetic material coded in the form of objects (it models audiovisual data as a composition of these objects). MPEG-4 provides the standardized technological elements enabling the integration of the production, distribution and content access paradigms of the fields of interactive multimedia, mobile multimedia, interactive graphics and enhanced digital television.

The MPEG-7 standard, formally named "Multimedia Content Description Interface", provides a rich set of standardized tools to describe multimedia content. Both human users and automatic systems that process audiovisual information are within the scope of MPEG-7.

MPEG-7 has been developed by experts representing broadcasters, electronics manufacturers, content creators and managers, publishers, intellectual property rights managers, telecommunication service providers and academia.

MPEG-7 offers a comprehensive set of audiovisual Description Tools (the metadata elements and their structure and relationships, that are defined by the standard in the form of Descriptors and Description Schemes) to create descriptions (i.e., a set of instantiated Description Schemes and their corresponding Descriptors at the users will), which will form the basis for applications enabling the needed effective and efficient access (search, filtering and browsing) to multimedia content. This is a challenging task given the broad spectrum of requirements and targeted multimedia applications, and the broad number of audiovisual features of importance in such context.

The new standard helps the tools of indexation to create big bases of audiovisual material (fixed images, graphics, three-dimensional models, audio, discourses, video and information be more than enough how those elements are combined in a multimedia presentation) and looking up in these manual bases of materials or automatically.

However, a description in XML can be very voluminous. It is a problem for the applications in which the space of storage or the width of band of transmission is insufficient (records with limited

capability, transmission for modem, etcetera). The BIM (Binary Format for MPEG 7) has unrolled the compressor itself for these cases.

This thesis is compound of several chapters that will describe all this new technology to analysis of images MPEG-7, the up to date of multimedia data bases, the algorithm of similarity search and an approximation of Classification of images.

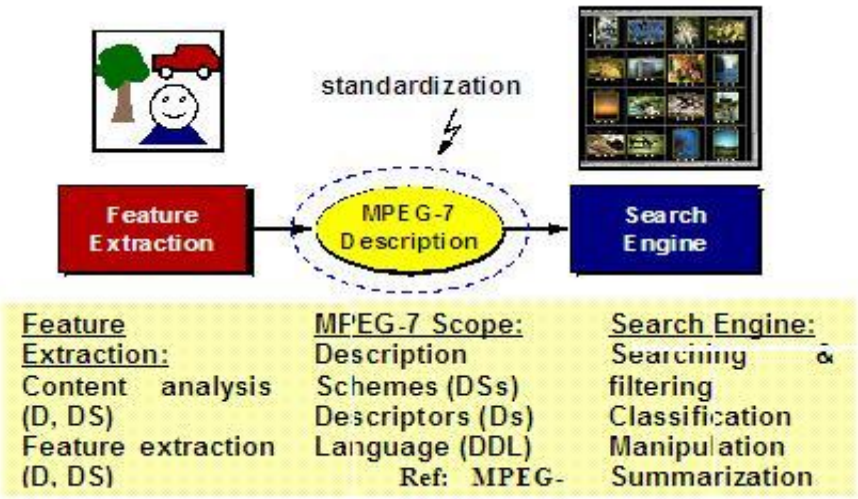


Figure 1.1: MPEG-7 Description from [9]

1.4 Others MPEG

The Moving Picture Coding Experts Group (MPEG) is a group of the organization ISO/IEC in charge of the development of international standards for compression, decompression, processing and performance encoded of moving images (video), audio and both combination. This group has created the following standards with video:

- **MPEG-1**: Initial standard of compression of audio and video. Used after like the standard for video CD, popular audio layer 3 format of compression include (MP3).

- **MPEG-2**: Standards of audio and video for diffusion of quality of television. Used for services of TV for satellite like Direct TV (North American Chain of television road satellite of direct diffusion), signs of cable digital TV and (with light modifications) for the DVD video records.

- **MPEG-4**: To support objects expands MPEG audio video, contents 3D, encoding of low binary velocity and support for step of digital rights (protection of copyright).

See [4] or [6] for more information concerning these standard.

1.5 MPEG-7 Objectives

In October 1996, MPEG started a new work item to provide a solution to the questions described above. The new member of the MPEG family, named "Multimedia Content Description Interface" (in short MPEG-7), provides standardized core technologies allowing the description of audiovisual data content in multimedia environments. It extends the limited capabilities of proprietary solutions in identifying content that exist today, notably by including more data types.

Audiovisual data content that has MPEG-7 descriptions associated with it may include: still pictures, graphics, 3D models, audio, speech, video, and composition information about how these elements are combined in a multimedia presentation (scenarios). A special case of these general data types is facial characteristics.

MPEG-7 descriptions do, however, not depend on the ways the described content is coded or stored. It is possible to create an MPEG-7 description of an analogue movie or of a picture that is printed on paper, in the same way as of digitized content.

MPEG-7 allows different granularity in its descriptions, offering the possibility to have different levels of discrimination. Even though the MPEG-7 description does not depend on the (coded) representation of the material, MPEG-7 can exploit the advantages provided by MPEG-4 coded content. If the material is encoded using MPEG-4, which provides the means to encode audio-visual material as objects having certain relations in time (synchronization) and space (on the screen for video, or in the room for audio), it will be possible to attach descriptions to elements (objects) within the scene, such as audio and visual objects.

Because the descriptive features must be meaningful in the context of the application, they will be different for different user domains and different applications. This implies that the same material can be described using different types of features, tuned to the area of application. To take the example of visual material: a lower abstraction level would be a description of e.g. shape, size, texture, color, movement (trajectory) and position (where in the scene can the object be found?); and for audio: key, mood, tempo, tempo changes, position in sound space. The highest level would give semantic information: ‘This is a scene with a barking brown dog on the left and a blue ball that falls

down on the right, with the sound of passing cars in the background.' Intermediate levels of abstraction may also exist.

The level of abstraction is related to the way the features can be extracted: many low-level features can be extracted in fully automatic ways, whereas high level features need (much) more human interaction.

Next to having a description of what is depicted in the content, it is also required to include other types of information about the multimedia data:

- The form - An example of the form is the coding format used (e.g. JPEG, MPEG-2), or the overall data size. This information helps determining whether the material can be 'read' by the user terminal;
- Conditions for accessing the material - This includes links to a registry with intellectual property rights information, and price;
- Classification - This includes parental rating, and content classification into a number of pre-defined categories;
- Links to other relevant material - The information may help the user speeding up the search;
- The context - In the case of recorded non-fiction content, it is very important to know the occasion of the recording (e.g. Olympic Games 1996, final of 200 meter hurdles, men).

1.6 Possible Applications

Granted that you have in phase of development, we will limit ourselves to encounter some examples that the potentiality of MPEG7 and some of his areas of application can illustrate:

- Multimedia bookstores: Quest and location of sequences of audio or video with determined characteristics (Examples: Operators of audio video in television. For the fast location of sound effects image in the realization of you program live).
- Biomedical applications: Location of echographies, X-ray pictures, scanners or any other type of how-to elements to the intervening diagnosis her specification of determined characteristic parameters (age, rhythm cardiac ...).
- Means of diffusion great public: Selection of canals attending to preferences of diverse nature (for example, differentiation between films of action and Interviews).
- Security System: Capable system to acknowledge to and to store to people, objects, situations.

The list is as extensive as we are able to imagine.

1.7 Main elements of MPEG-7 Standard

- Description Tools: Descriptors (D), that define the syntax and the semantics of each feature (metadata element).
- Description Schemes (DS) that specify the structure and semantics of the relationships between their components that may be both Descriptors and Description Schemes.
- *A Description Definition Language* (DDL) to define the syntax of the MPEG-7 Description Tools and to allow the creation of new Description Schemes and, possibly, Descriptors and to allow the extension and modification of existing Description Schemes.
- *Classification Schema* (CS), defines a list of typical terms used in many applications together with the meanings. For instance, it allow the definitions of file formats in a standardize way. MPEG-7 provides many predefined CSs for characterizing roles, formats, and so forth. New CSs have to be registered.
- *Extensibility*, supported through MPEG-7 schema extensions mechanism (new DSs and Ds)
- *System tools*, to support binary coded representation for efficient storage and transmission, transmission mechanisms (both for textual and binary formats), multiplexing of descriptions, synchronization of descriptions with content, management and protection of intellectual property in MPEG-7 descriptions, etc.

Therefore, MPEG-7 Description Tools allows to create descriptions (i.e., a set of instantiated Description Schemes and their corresponding Descriptors at the users will), to incorporate application specific extensions using the DDL and to deploy the descriptions using System tools.

The MPEG-7 descriptions of content that may include:

- Information describing the creation and production processes of the content (director, title, short feature movie).
- Information related to the usage of the content (copyright pointers, usage history, and broadcast schedule).
- Information of the storage features of the content (storage format, encoding).
- Structural information on spatial, temporal or spatio-temporal components of the content (scene cuts, segmentation in regions, region motion tracking).
- Information about low level features in the content (colors, textures, sound timbres, melody description).
- Conceptual information of the reality captured by the content (objects and events, interactions among objects).

- Information about how to browse the content in an efficient way (summaries, variations, spatial and frequency subbands).
- Information about collections of objects.
- Information about the interaction of the user with the content (user preferences, usage history).

All these descriptions are of course coded in an efficient way for searching, filtering, etc.

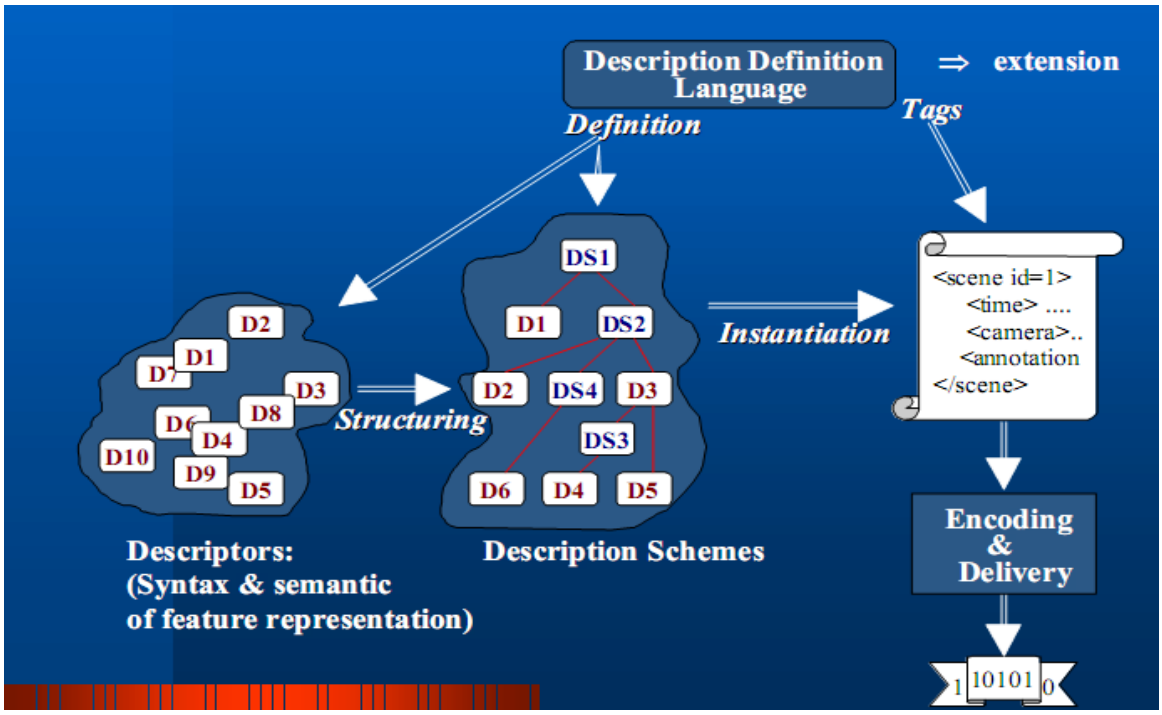


Figure 1.2: DDL Schema from [9]

1.8 Parts of MPEG-7

The MPEG-7 Standard consists of the following parts:

1. *MPEG-7 Systems* – the tools needed to prepare MPEG-7 descriptions for efficient transport and storage and the terminal architecture.
2. *MPEG-7 Description Definition Language* - the language for defining the syntax of the MPEG-7 Description Tools and for defining new Description Schemes.
3. *MPEG-7 Visual* – the Description Tools dealing with (only) Visual descriptions.
4. *MPEG-7 Audio* – the Description Tools dealing with (only) Audio descriptions.
5. *MPEG-7 Multimedia Description Schemes* - the Description Tools dealing with generic features and multimedia descriptions.

6. *MPEG-7 Reference Software* - a software implementation of relevant parts of the MPEG-7 Standard with normative status.
7. *MPEG-7 Conformance Testing* - guidelines and procedures for testing conformance of MPEG-7 implementations
8. *MPEG-7 Extraction and use of descriptions* – informative material (in the form of a Technical Report) about the extraction and use of some of the Description Tools.
9. *MPEG-7 Profiles and levels* - provides guidelines and standard profiles.
10. *MPEG-7 Schema Definition* - specifies the schema using the *Description Definition Language.* [5]

1.8.1 Visual Descriptor

MPEG-7 visual descriptor tools consist mainly of descriptors that cover color, texture, shape, motion and face recognition. They mainly use a histogram-based approach of representation. That is, they compute a vector (histogram) of elements each representing the number of pixels (regions) in a given image, which have similar characteristics. In this work we explain only the color descriptors if you want to know more about the others use [8], [9] and [10].

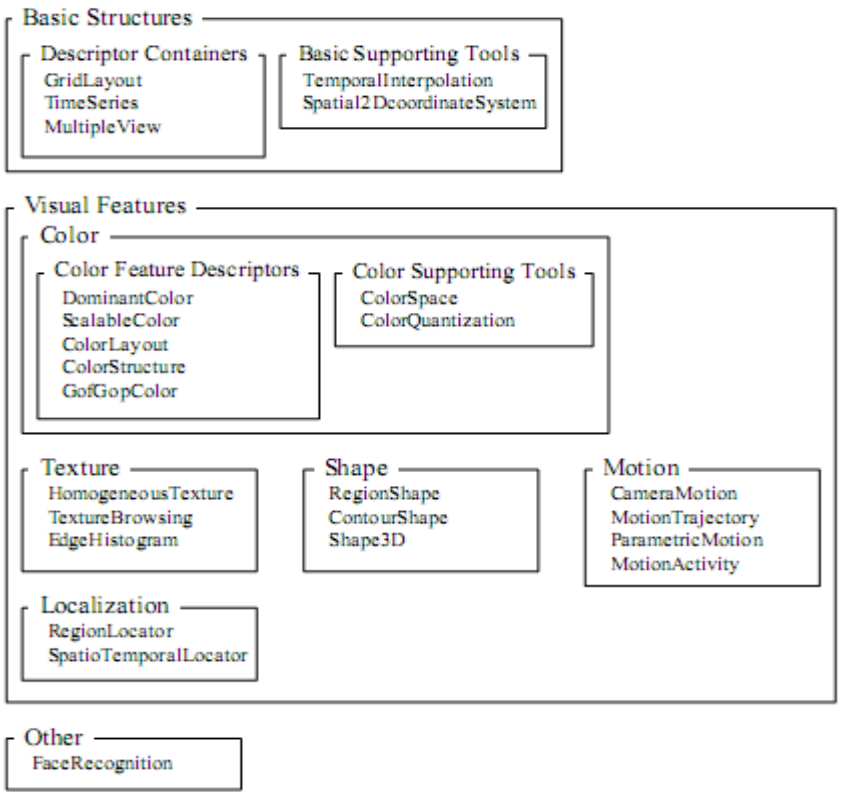


Figure 1.3 Overview of a visual descriptor tools [4]

1.8.2 Color Descriptors

There are seven color descriptors available for description in MPEG-7: *Color space, Color Quantization, Dominant Colors, Scalable Color, Color Layout, Color-Structure, and GoF/GoP Color (group of frames and group of pictures)*. Here we give a brief overview of these descriptors (including their DLL) and show how descriptions are created.

1.8.3 Color Space Descriptor

This feature is the color space that is be used. The following main color spaces are supported for description:

- R,G,B Color Space.
- Perceptual Color Spaces available for descriptions are:
    - Y, Cr, Cb Color Space.
    - H,S,V Color Space.
    - HMMD.

1.8.4 Color Quantization Descriptor

This descriptor defines a uniform quantization of a color space. The number of bins which the quantizer produces is configurable, such that great flexibility is provided for a wide range of applications. For a meaningful application in the context of MPEG-7, this descriptor has to be combined with dominant color descriptors, e.g. to express the meaning of the values of dominant colors.

1.8.5 Dominant Color Descriptor

This color descriptor is most suitable for representing local (object or image region) features where a small number of colors are enough to characterize the color information in the region of interest. It also applicable on whole images, for example flag images or color trademark images. Color quantization is used to extract a small number of representing colors in each region/image. The percentage of each quantized color in the region is calculated correspondingly. A spatial coherency on the entire descriptor is also defined, and is used in similarity retrieval.

1.8.6 Scalable Color Descriptor

The Scalable Color Descriptor is a Color Histogram in HSV Color Space, which is encoded by a Haar transform. Its binary representation is scalable in terms of bin numbers and bit representation accuracy over a broad range of data rates. The Scalable Color Descriptor is useful for image-to-image matching and retrieval based on color feature. Retrieval accuracy increases with the number of bits used in the representation.

1.8.7 Color Layout Descriptor

This descriptor effectively represents the spatial distribution of color of visual signals in a very compact form. This compactness allows visual signal matching functionality with high retrieval efficiency at very small computational costs. It provides image-to-image matching as well as ultra high-speed sequence-to-sequence matching, which requires so many repetitions of similarity calculations. It also provides very friendly user interface using hand-written sketch queries since this descriptors captures the layout information of color feature. The sketch queries are not supported in other color descriptors.

The advantages of this descriptor are:

- That there are no dependency on image/video format, resolutions, and bit-depths. The descriptor can be applied to any still pictures or video frames even though their resolutions are different. It can be also applied both to a whole image and to any connected or unconnected parts of an image with arbitrary shapes.
- That the required hardware/software resource for the descriptor is very small. It needs as law as 8 bytes per image in the default video frame search, and the calculation complexity of both extraction and matching is very low. It is feasible to apply this descriptor to mobile terminal applications where the available resources is strictly limited due to hardware constrain.
- That the captured feature is represented in frequency domain, so that users can easily introduce perceptual sensitivity of human vision system for similarity calculation.
- That it supports scalable representation of the feature by controlling the number of coefficients enclosed in the descriptor. The user can choose any representation granularity depending on their objectives without interoperability problems in measuring the similarity among the descriptors with different granularity. The default number of coefficients is 12 for video frames while 18 coefficients are also recommended for still pictures to achieve a higher accuracy.

1.8.8 Color Structure Descriptor

The Color structure descriptor is a color feature descriptor that captures both color content (similar to a color histogram) and information about the structure of this content. Its main functionality is image-to-image matching and its intended use is for still-image retrieval, where an image may consist of either a single rectangular frame or arbitrarily shaped, possibly disconnected, regions. The extraction method embeds color structure information into the descriptor by taking into account all colors in a structuring element of 8x8 pixels that slides over the image, instead of considering each pixel separately. Unlike the color histogram, this descriptor can distinguish between two images in which a given color is present in identical amounts but where the structure of the groups of pixels having that color is different in the two images. Color values are represented in the double-coned HMMD color space, which is quantized non-uniformly into 32, 64, 128 or 256 bins. Each bin amplitude value is represented by an 8-bit code. The Color Structure descriptor provides additional functionality and improved similarity-based image retrieval performance for natural images compared to the ordinary color histogram.

1.8.9 GOF & GOP Color Descriptor

The Group of Frames/Group of Pictures color descriptor extends the ScalableColor descriptor that is defined for a still image to color description of a video segment or a collection of still images. Additional two bits allow to define how the color histogram was calculated, before the Haar transform is applied to it: by average, median or intersection. The average histogram, which refers to averaging the counter value of each bin across all frames or pictures, is equivalent to computing the aggregate color histogram of all frames and pictures with proper normalization. The Median Histogram refers to computing the median of the counter value of each bin across all frames or pictures. It is more robust to round-off errors and the presence of outliers in image intensity values compared to the average histogram. The Intersection Histogram refers to computing the minimum of the counter value of each bin across all frames or pictures to capture the "least common" color traits of a group of images. Note that it is different from the histogram intersection, which is a scalar measure. The same similarity/distance measures that are used to compare scalable color descriptions can be employed to compare GoF/GoP color Descriptors.

1.9 MPEG-21

The appetite for consuming content and the accessibility of information continues to increase at a rapid pace. Access devices, with a large set of differing terminal and network capabilities, continue to evolve, having a growing impact on peoples' lives. Additionally, these access devices possess the functionality to be used in different locations and environments: anywhere and at anytime. Their users, however, are currently not given tools to deal efficiently with all the intricacies of this new multimedia usage context.

Solutions with advanced multimedia functionality are becoming increasingly important as individuals are producing more and more digital media, not only for professional use but also for their personal use. All these "content providers" have many of the same concerns: management of content, re-purposing content based on consumer and device capabilities, protection of rights, protection from unauthorized access/modification, protection of privacy of providers and consumers, etc.

Such developments are pushing the boundaries of existing business models for trading physical goods and require new models for distributing and trading digital content electronically. For example, it is becoming increasingly difficult for legitimate users of content to identify and interpret the different intellectual property rights that are associated with the elements of multimedia content. Additionally, there are some users who freely exchange content with disregard for the rights associated with content and rights holders are powerless to prevent them. The boundaries between the delivery of audio (music and spoken word), accompanying artwork (graphics), text (lyrics), video (visual) and synthetic spaces are becoming increasingly blurred. New solutions are required for the access, delivery, management and protection processes of these different content types in an integrated and harmonized way, to be implemented in a manner that is entirely transparent to the many different users of multimedia services.

The need for technological solutions to these challenges is motivating the MPEG-21 Multimedia Framework initiative that aims to enable the transparent and augmented use of multimedia resources across a wide range of networks and devices.

1.9.1 What is MPEG-21?

The MPEG-21 standard, from the Moving Picture Experts Group aims at defining an open framework for multimedia applications. ISO 21000.

Specifically, MPEG-21 defines a "Rights Expression Language" standard as means of sharing digital rights/permissions/restrictions for digital content from content creator to content consumer. As an XML-based standard, MPEG-21 is designed to communicate machine-readable license information and do so in a "ubiquitous, unambiguous and secure" manner.

Among the aspirations for this standard that the industry hopes will put an end to illicit file sharing is that it will constitute: "A normative open framework for multimedia delivery and consumption for use by all the players in the delivery and consumption chain. This open framework will provide content creators, producers, distributors and service providers with equal opportunities in the MPEG-21 enabled open market."

MPEG-21 is based on two essential concepts: the definition of a fundamental unit of distribution and transaction, which is the Digital Item, and the concept of users interacting with them. Digital Items can be considered the kernel of the Multimedia Framework and the users can be considered as who interacts with them inside the Multimedia Framework. At its most basic level, MPEG-21 provides a framework in which one user interacts with another one, and the object of that interaction is a Digital Item. Due to that, we could say that the main objective of the MPEG-21 is to define the technology needed to support users to exchange, access, consume, trade or manipulate Digital Items in an efficient and transparent way.

1.10 MPEG-7 and XML

The syntax for the associated Description Definition Language (DDL) is XML. "The Description Definition Language (DDL) is the language which allows the creation of MPEG-7 Description Schemes and Descriptors. A DDL schema (a DDL file) specifies the constraints that a valid MPEG-7 description should respect. It is encoded in XML. The DDL is used by MPEG-7 groups when they have to define any structured data model. For instance the MDS, audio and video groups are using it to define their descriptors and description schemes".

In this work we won’t talk about how xml works, only the tools to put into the database.

1.10.1 XML and Oracle

Oracle has a new type to manipulate data was introduced XML, called XMLType. This type of piece of information can be used to define columns of tables and views, arguments for stored procedures and other places where the type of native piece of information used. XMLType defines operators' rich game XML to extract, to transform and to validate data XML. However, the types of data not indicate the option used of storage for the data XML. In effect, this is designed for

accommodating a selection varied of storages, from once completely not a storage was structured highly structured.

Table : Pros and cons of XML storage options in Oracle XML DB

Feature	LOB Storage	Structured Storage (object-relational storage)
Database schema flexibility	Very flexible when schemas change.	Limited flexibility for schema changes. Similar to the ALTER TABLE restrictions.
Data integrity and accuracy	Maintains the original XML byte for byte-important in some applications.	Trailing new lines, whitespace within tags, and data format for nonstring datatypes is lost. But maintains Document Object Model (DOM) fidelity.
Performance	Mediocre performance for data manipulation language (DML).	Excellent DML performance.
Access to SQL	Some accessibility to SQL features.	Good accessibility to existing SQL features, such as constraints and indexes.
Space needed	Can consume considerable space.	Needs less space than CLOB storage, in particular when used with an Oracle XML DB- registered XML Schema.

Figure 1.4: Table of pros and cons of XML storage from [11].

1.10.2 XMLType

Oracle9i Release 1 (9.0.1) introduced a new data type, XMLType, to facilitate native handling of XML data in the database. The following summarizes XMLType:

- XMLType can be used in PL/SQL stored procedures as parameters, return values, and variables.
- XMLType can represent an XML document as an instance (of XMLType) in SQL.
- XMLType has built-in member functions that operate on XML content. For example, you can use XMLType functions to create, extract, and index XML data stored in Oracle9i database.

Functionality is also available through a set of Application Program Interfaces (APIs) provided in PL/SQL and Java.

With XMLType and these capabilities, SQL developers can leverage the power of the relational database while working in the context of XML. Likewise, XML developers can leverage the power of XML standards while working in the context of a relational database.

Chapter 2. Similarity Search

2.1 Introduction

For this application we used an algorithm to query of database information about the pictures that we have stored. The search problem is constrained in general by the type of data stored in the database, the method of comparing individual data instance, and the specification of the query by which users express their information needs.

We have the distance searching problem, because the exact match retrieval for traditional databases, is neither feasible nor meaningful for data types in the present digital age.

A very useful search paradigm is to quantify the proximity, similarity, or dissimilarity of a query object versus the object stored in the database to be search.

2.1.1 Mathematical Theory

The search problem can be described as follow:

Let D be a domain, d a distance measure on D, and (D, d) a metric space. Given a set X $\subseteq$ *D of* n *elements, preprocess or structure the data so that proximity queries are answered efficiently*.

X can be seen as a file (a dataset or a collection) of objects that takes values from the domain D, with d as the proximity measure. Thought several types of similarity queries exist like *similarity range* and *the nearest neighbor* queries.

The primary reasons for looking at the distance data search problem are the following:

- There are numerous applications where the proximity criteria offer no special proprieties but distance, so a metric search becomes the sole option.
- Many specialized solutions for proximity search perform no better than indexing techniques based on distances. Metric search thus forms a viable alternative.
- If a good solution utilizing generic metric space can be found, it will provide high extensibility.

2.2 Distance Measure

The distance functions of metric spaces represent a way to quantifying the closeness of objects in a given domain. Depending on the character of values returned, distance measure can be divided into two groups:

- Discrete: Return only a small set of values.
- Continuous: Distance functions in which the cardinality of the set of values returned is very large or infinite.

2.2.1 Mikowsky Distances

The Minskowski distance functions form a whole family of metric functions (Lp)

$$Lp\ [(x1,\dots,xn)(y1,\dots,yn)] = \sqrt[p]{\sum_{i=1}^{n} |xi - yi|^p}$$

These functions are defined on n-dimensional vector of real number.

2.2.2 Quadratic for Distance

I will use this function for my application, with this I can manage color histograms of images, where each dimension represents a specific color. The following expression represents a generalized quadratic distance measure d_M:

$$d_M(\vec{x},\vec{y}) = \sqrt{(\vec{x},\vec{y})^T * M * (\vec{x},\vec{y})}$$

Where M is n x n (2-dimensional) positive semi-definite matrix, where the weights $m_{i,j}$ denote how strong the connection between two components I and j of vector $\vec{x}$ and $\vec{y}$ is, respectively. And T is the vector transposition.

Applying such a matrix, the quadratic form distance formula turns out to be as follows, yielding the general formula for the weighted Euclidian distance:

$$d_M(\vec{x},\vec{y}) = \sqrt{\sum_{i=1}^{n} w_i(\vec{x},\vec{y})^2}$$

This is the formula that we will use, with the appropriate data of the database.

2.2.3 Edit Distance

This method is good for the closeness of sequences of symbols (Strings), also called Levenshtein distance. The theory is the next, the distance between two strings *x =x1…xn* and *y=y1…yn is* defined as the minimum number of atomic edit operations (insert, delete, replace) needed to transform string *x* into string *y*.

2.3 Similarity Queries

What is a similarity query? Is defined explicitly or implicitly by a query object q and a constraint on the form and extent of proximity required, typically expressed as a distance. The response to a query returns all objects which satisfy the selection conditions, presumed to be those objects close to the given query object.

2.3.1 Range Query

Probably the most common type of similarity query, R(q,r). The query is specified by a query object q ∈ D, with some query radius r as the distance constrain. The query retrieves all objects found within distance r of q. (Example: Give me all museums within a distance of two kilometers from my hotel.).

If the search radius is zero, the range query is called a *point query* or *exact match*. This case is when we are looking for an identical copy (or queries) of the query object q.

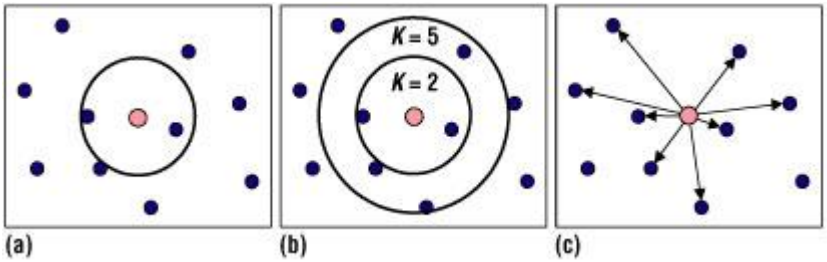


Figure 2.1 - Example of Range Query [17]

2.3.2 Nearest Neighbor Query

Over the last few years, there has been considerable interest in the database community with regard to supporting K-Nearest Neighbor (KNN) queries. The general model of a KNN query is that the user gives a point query in multidimensional space and a distance metric for measuring distances between points in this space. The system is then expected to find it, with regard to this metric, the K closest answers in the database from the query point. Typical distance metrics include Euclidean distance, Manhattan distance, etc.

It is possible that a majority of the answers to a KNN query may be very similar to one or more of the other answers, especially when the data has clusters. In fact, there may even be duplicates the attributes of the multidimensional space. For a variety of applications, such as online restaurant selection, providing homogeneous result sets may not add value to the user. [17]

Chapter 3. Classification and Prediction

3.1 Introduction

Classification and prediction are two forms of data analysis that can be used to extract models describing important data classes or to predict future data trends. Such analysis can help us with a better understanding of the data at large. Classification predicts categorical labels and prediction models continuous valued function.

Many classification and prediction methods have been proposed by researchers in machine learning, pattern recognition and statistics. In this project we will focus on some theoretic aspects for the elaboration of classification of images stored in our data base.

3.2 Definitions

We are going to define concepts that we will use in this chapter:

- **Classification:** Where a model o classifier is constructed to predict categorical labels (such as "yes" or "not"). These categories can be represented by discrete values, where the ordering among values has no meaning.
- **Prediction:** Where the model constructed predicts a continuous valued function, or ordered value, as opposed to a categorical label.

3.2.1 How the classification work?

Data classification is two-steps process:

1. *Learning:* Training data are analyzed by classification algorithm. Here, the class label attribute and the learn model or classifier is represented in the form of classification rules. Because the class label of each training sample (or tuple) is provided, this step is also known as supervised learning.

2. *Classification:* Test data are used to estimate the accuracy of the classification rules. If the accuracy is considered acceptable, the rules can be applied to the classification of new data samples. This is unsupervised learning.

3.2.2 How is Prediction different from Classification?

Data prediction is a two steps process, similar to that of data classification as described before. However, for the prediction, we lose the terminology of "class label attribute" because the attribute for witch values are being predicted is continuous valued (ordered) rather than categorical (discrete valued and unordered). The attribute can be referred to simply as the predicted attribute.

3.3 Classification by Decision Tree Induction

Decision tree induction is the learning of decision trees from class labeled training samples.

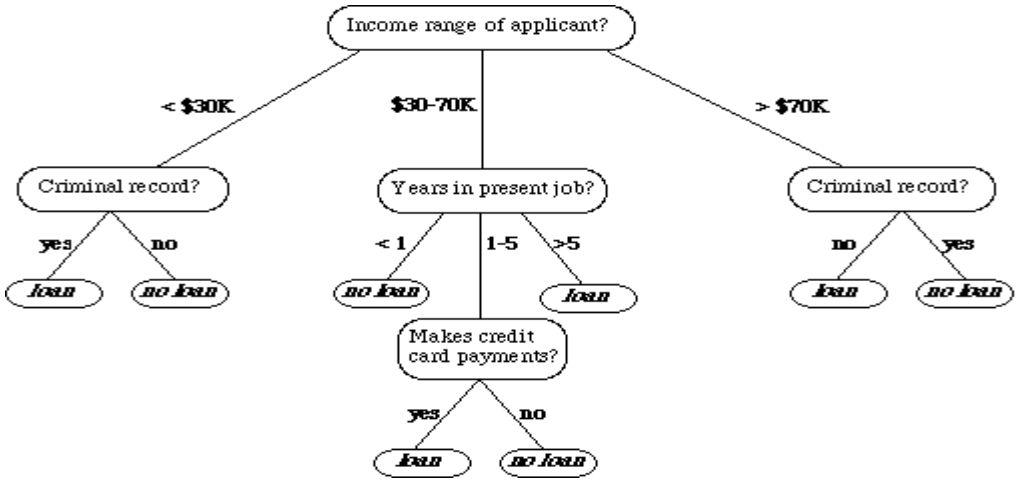


Figure 3.1 - Sample of decision tree induction [16].

Are flow charts where each internal node denotes a test on an attribute, each branch representing an outcome of the test and the terminal node hold a class label.

Given a tuple, X, for which the associated class label is unknown, the attribute values of the tuple are tested against the decision tree. A path is traced from tge root to a leaf node, which holds the class prediction for that tuple. Decision tree can easily be converted to classification rules.

3.4 Bayesian Classification

In this project we will use (normally) this statistical classifier because Bayesian have high accuracy and speed when applied to large databases, they can predict class membership probabilities, such as the probability that a given tuple belongs to a particular class.

This classification is based on Bayes' theorem. This theorem says:

"Let X be a data tuple. In Bayesian terms, X is considered *evidence*. Let H be some hypothesis, such as that the data tuple X belongs to a specified class C. For classification problems, we want to determinate P(H|X), the probability that the hypothesis H holds given the *evidence* or observed data tuple X. " In other words, we are looking for the probability that tuple X belongs to class C, given that we know that attribute description of X.

$$p(H|X) = \frac{\mathrm{P(X|H)P(H)}}{P(X)}$$

3.5 Naïve Bayesian Classification

Studies comparing classification algorithms have found a simple Bayesian classifier know as the naïve Bayesian classifier to be comparable in performance with decision tree and selected neural network classifier.

Naïve Bayesian classifier assume that effect of an attribute value on a given class is independent of the values of the others attributes. This assumption is called class conditional independence. It is made to simplify the computations involved and, in this sense is considered "naïve."

3.6 Support Vector Machine (SVM)

It's a new method for the classification of both linear and nonlinear data. SVM is an algorithm that uses a nonlinear mapping to transform the original training data into a higher dimension. Within this new dimension, it searches for the linear optimal separating hyperplane (that is, separating the tuples of one class from another).

The SVM find this hyperplane using support vectors ("essential" training tuples) and margins (defined by the support vectors).

3.6.1 The case when the data are linearly separable

To explain this section of the chapter, look this simple sample figure 6.1. From the graph we see that 2-D data are *linearly separable* (or linear) because a straight line can be drawn to separate all of the tuples of class + from all the tuples of class -. There are an infinite number of separating lines that could be drawn. We want to find the best one, if we generalizing to n dimensions we want to find the best hyperplane.

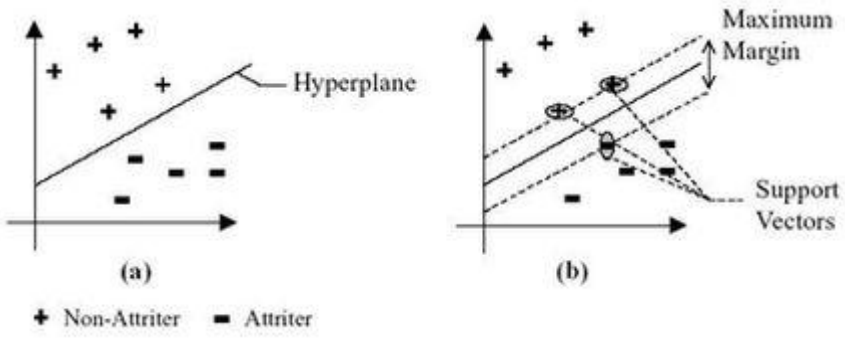


Figure 3.2 - Data linearly Separable (a) (b) [16]

So, how we can find the best line or hyperplane? An SVM approaches this problem by searching for the maximum marginal hyperplane (consider Figure 6.1 b).

The SMV finds the maximum separating hyperplane, that is, the one with maximum distance between the nearest training tuples. The support vectors are show with a thicker border.

Figure 6.1b we can see the support vectors, they are equally close to the (separating) MMH. Essentially, the support vectors are the most difficult tuples to classify and give the most information regarding classification.

3.6.2 The case when the data are linearly inseparable

If the data are not linearly separable, no straight line can be found that would separate the classes.

SMVs can be extended to create anonlinear SVMs for the classification of *linearly inseparable data* (or *nonlinear data*). Such SMVs are capable of finding nonlinear decision boundaries in input space.

We obtain a nonlinear SMV by extending the approach for linear SMVs. We can use diferents kernel fuctions to be used:

- Polygonal Kernel of degree h.
- Sigmoid kernel.
- Gaussian radial basis function kernel.

We will use RBF function to make run the experiment. A radial basis function (RBF) is a real-valued function whose value depends only on the distance from the origin, so that $\phi(\mathbf{x}) = \phi(||\mathbf{x}||)$; or alternatively on the distance from some other point c, called a center, so that $\phi(\mathbf{x}, \mathbf{c}) = \phi(||\mathbf{x} - \mathbf{c}||)$. Any function φ that satisfies the property φ(x)=φ(||x||) is a radial function. The norm is usually Euclidean distance.

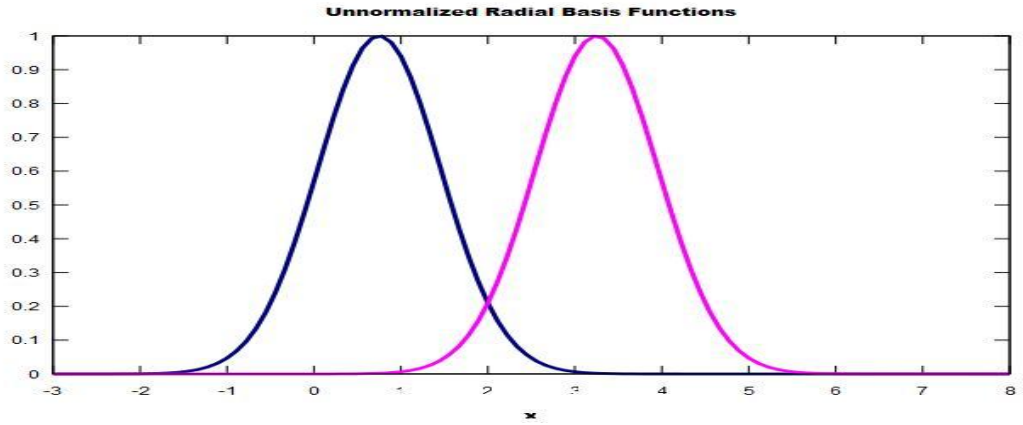


Figure 3.3 - RBF function [13]

3.7 Accuracy and error measure

What is accuracy? How can I estimate it? Are there strategies for increasing the accuracy of a learned model?, tha answers are in this section .

3.7.1 Classifier Accuracy measure

The accuracy of classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier.

We can also speak of the error rate or misclassification rate of a classifier M, we can define as 1-Accuracy(M).

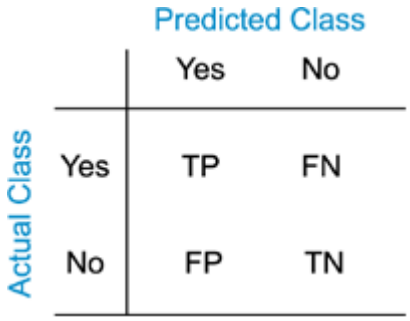


Figure 3.4 - Confusion Matrix

The confusion matrix is a useful tool for analyzing how well your classifier can recognize tuples of a different classes. A confusion matrix for two classes is shown in Figure 7.1, we can talk in terms of *positive tuples* (tuples of the main class of interest) versus *negative tuples*. *True positives* refer to the positive tuples were correctly labeled by the classifier, while *true negatives* are the negative tuples that were correctly labeled by the classifier. False positive are the negative tuples that were incorrectly labeled. Similarly false negative are the positive tuples that were incorrectly labeled.

The alternatives to the accuracy measure are the sensitive, specificity and precision. Sensitive is also refered to as the true positive (recognition) rate (that is, the proportion of positive tuples that are correctly identified), while specificity is the true negative rate (that is, the proportion of negative tuples that are correctly identified) :

- Sensititivity = $\frac{t\_pos}{pos}$
- Specificity = $\frac{t\_neg}{neg}$
- Precision = $\frac{t\_pos}{(t_{pos} + f\_pos)}$

Where t_pos is the number of true positives, t_neg is the number of true negatives, neg is the number of negatives tuples and f-pos is the number of false positives.

$$\text{Accuracy} = sensitivity \frac{pos}{(pos+neg)} + specifitivy \frac{neg}{(pos+neg)}$$

3.8 Evaluating the accuracy of a Classifier

How can we use the above measures to obtain a reliable estimate of classifier accuracy? We will use to this project cross-validation, the use of this technique to estimate accuracy increases the overall computation time, yet is useful for the model selection.

3.8.1 Cossvalidation

Cross-validation, sometimes called rotation estimation, is the statistical practice of partitioning a sample of data into subsets such that the analysis is initially performed on a single subset, while the other subset(s) are retained for subsequent use in confirming and validating the initial analysis.

The initial subset of data is called the training set; the other subset(s) are called validation or testing sets.

In K-fold cross-validation, the original sample is partitioned into K subsamples. Of the K subsamples, a single subsample is retained as the validation data for testing the model, and the remaining K−1 subsamples are used as training data. The cross-validation process is then repeated K times (the folds), with each of the K subsamples used exactly once as the validation data. The K results from the folds then can be averaged (or otherwise combined) to produce a single estimation.

Chapter 4. Oracle InterMedia

4.1 Introduction

Oracle Multimedia (Formerly Oracle *inter*Media) is a feature of Oracle Database that is included in both Standard Edition and Enterprise Edition.

- It is a general purpose feature that enables the efficient management and retrieval of image, audio, and video data.
- Oracle Multimedia has knowledge of the most popular multimedia formats and can automate metadata extraction and basic image processing.
- Oracle Multimedia increases programmer productivity when developing multimedia applications using JSPs, Servlets, PL/SQL or when using tools such as JDeveloper (ADF/UIX) and Oracle Portal.

Oracle Multimedia manages multimedia data within Oracle Database under transaction control. Alternately, Oracle Database can store and index meta-information together with external references that enable efficient access to media content stored outside the database.

Oracle Multimedia supports storage of the popular file formats, including desktop publishing images, and streaming audio and video formats in databases. Oracle Multimedia provides the means to add audio, image, and video, or other heterogeneous media columns or objects to existing tables, and insert and retrieve multimedia data. This enables database designers to extend existing databases with multimedia data, or to build new end-user multimedia database applications. Oracle Multimedia developers can use the basic functions provided here to build specialized multimedia applications.

Oracle Multimedia is extensible. It supports a base set of popular audio, image, and video data formats for multimedia processing that also can be extended, for example, to support additional formats, new digital compression and decompression schemes (codecs), data sources, and even specialized data processing algorithms for audio and video data.

The Oracle Multimedia architecture defines the framework through which media-rich content as well as traditional data are supported in the database. This content and data can then be securely shared across multiple applications written with popular languages and tools, easily managed and

administered by relational database management and administration technologies, and offered on a scalable database that supports thousands of users.

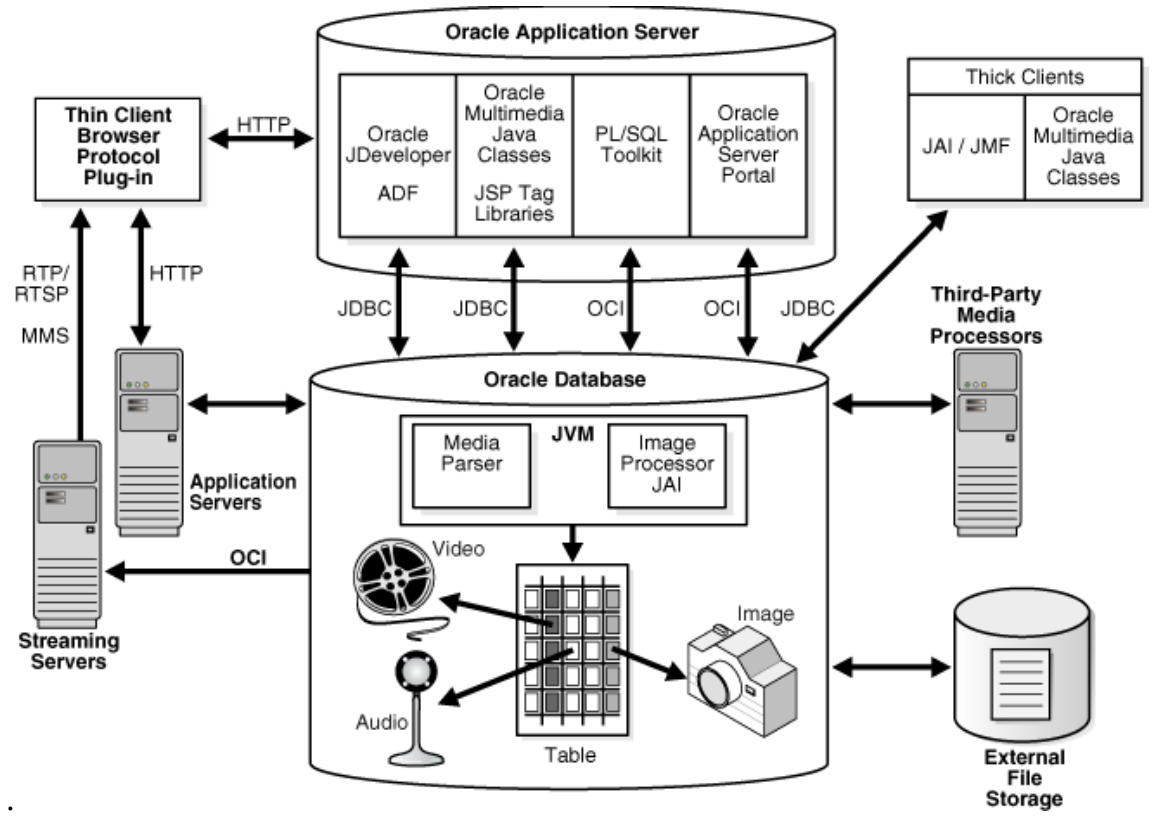


Figure 4.1 – Oracle Multimedia Architecture [11]

4.2 Procedure to insert Images and Characteristics

When ORDImage in addition to support the image, various metadata of information extract themselves, generates a type's object itself. For example:

- Information on the storage of the piece of information, including the kind of origin, location and name. Time of bringing up to date of the piece of information and format.

- MIME media type (used in applications Web and mail).
- Height and width of the image (for the case of images), and I deliver of the contents of the image format and type of compression.
- Metadata selected of the application right after the insertion.

Besides, ORDImageSignature's instances contain methods to grab of automatic way the low-level characteristics of the image stored in an object ORDImage, for it uses the method generateSignature( ). This will allow accomplishing consultations based in intrinsic contentses like, for example, in than measure the color and the form the images stored in the data base correspond with another one in particular.

4.3 Retrieve of Images

The images and metadata of information can store in objects ORDImage themselves and besides ORDImageSignature can store the identity tag of an image in objects itself.

Objects ORDImage and ORDImageSignature supply several methods and operators for recuperation with information for contents. For example, if you want finding a particular image the characteristics of the images using methods can compare and or operators defined for the type's object ORDImageSignature.

You intermediate in order to compare images, is assigned to a weight the visual attributes, then he calculates the score for them, besides a threshold to indicate until score tolerates itself in the answer should be supplied. Threshold to indicate until score tolerates itself in the answer should be supplied.

Once the metadata of information were stored they can be consulted and prosecuted easily by means of shows supplied especially for types XMLType.

The advantage of storing the metadata of information in format XML, the fact that it is very simple from storing is, but at the same time like the data they are structured, the recuperation for semantic contents is not limited. If they store the metadata of information in schemata of tie, the operations of storage will leave a lot of files once tie of data were separated from full nesses, that they will have a superior cost.

4.4 Metadata Extraction

Oracle Multimedia provides the ability to extract content and format metadata from media sources (audio and video files), and collects and organizes this metadata as an XML formatted CLOB. Once metadata has been extracted and stored, you can index the metadata for powerful full text and thematic media searches using Oracle Text. Thus, the database can be queried to locate the media data based on the metadata extracted from the media.

4.5 Image Processing

Oracle Multimedia supports image processing, such as image format transcoding, image cutting, image scaling, and generating thumbnail images. In addition, specifically when the destination image file format is RAW Pixel (RPIX) format or Microsoft Windows Bitmap (BMPF) image format, Oracle Multimedia supports a variety of operators for changing the format characteristics.

4.6 InterMedia and Images

This section contains information about digitized image concepts and using the ORDImage object type to build image applications or specialized ORDImage objects.

4.6.1 Digitized Images

ORDImage integrates the storage, retrieval, and management of digitized images in a database. ORDImage supports two-dimensional, static, digitized raster images stored as binary representations of real-world objects or scenes. Images may be produced by a document or photograph scanner, a video source such as a digital camera or VCR connected to a video digitizer or frame grabber, other specialized image capture devices, or even by program algorithms. Capture devices take an analog or continuous signal such as the light that falls onto the film in a camera, and convert it into digital values on a two-dimensional grid of data points known as pixels. Devices involved in the capture and display of images are under application control

4.6.2 Image Component

Digitized images consist of the image data (digitized bits) and attributes that describe and characterize the image data. Image applications sometimes associate application-specific information, such as the name of the person pictured in a photograph, description of the image, date photographed, photographer, and so forth, with image data by storing this descriptive text in an attribute or column in the database table.

The image data (pixels) can have varying depths (bits per pixel) depending on how the image was captured, and can be organized in various ways. The organization of the image data is known as the data format. ORDImage can store and retrieve image data of any data format. ORDImage can process and automatically extract properties of images of a variety of popular data formats. We can see in the next picture the aspect of an ORDImage object:

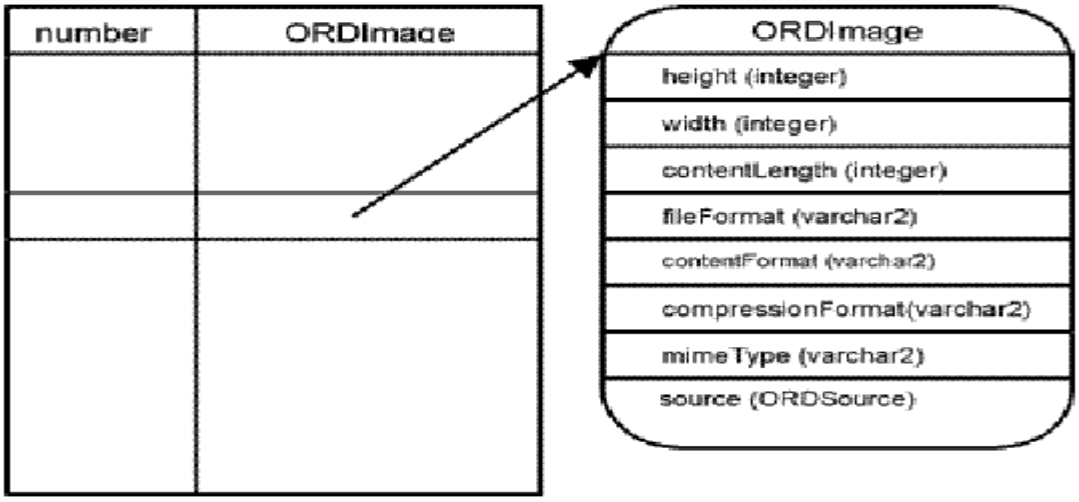


Figure 4.2 - ORDImage Type Structure [12]

- ***Height:*** the height of the image in pixels.
- ***Width:*** the width of the image in pixels.
- ***contentLength:*** the size of the on-disk image file in bytes.
- ***fileFormat:*** file type or format in which the image data is stored (TIFF, JIFF...).
- ***contentFormat:*** the type of image (monochrome and so forth).
- ***compressionFormat:*** the compression algorithm used on the image data.
- ***mimeType:*** the MIME type information.

- ***source:*** the source of the stored image data.

In addition, certain foreign images (formats not natively supported by ORDImage) have limited support for image processing.

The storage space required for digitized images can be large compared to traditional attribute data such as numbers and text. Many compression schemes are available to squeeze an image into fewer bytes, thus reducing storage device and network load. Lossless compression schemes squeeze an image so that when it is decompressed, the resulting image is bit-for-bit identical with the original. Lossy compression schemes do not result in an identical image when decompressed, but rather, one in which the changes may be imperceptible to the human eye. As compared with lossless schemes, lossy schemes generally provide higher compression.

Image interchange format describes a well-defined organization and use of image attributes, data, and often compression schemes, allowing different applications to create, exchange, and use images. Interchange formats are often stored as disk files. They may also be exchanged in a sequential fashion over a network and be referred to as a protocol. There are many application subdomains within the digitized imaging world and many applications that create or utilize digitized images within these. ORDImage supports storage and retrieval of all image data formats, and processing and attribute extraction of many image data formats.

4.7 Image support for Java

Oracle provides through its Oracle.Ord.Im package a complete Java API to work with interMedia. Java programmers are intimately familiar with Java objects, but they are often unaware that Oracle Database is an object-relational database, and as such supports storage and retrieval of objects. As we saw before, Oracle interMedia provides the database type ORDImage which is used to store images in a database table just like any other relational data. Some interMedia functionality (such as thumbnail generation) may also be used if images are stored in BLOB (Binary Large Object) columns, but Oracle Corporation recommends storing images in ORDImage columns. The reader can review the structure of ORDImage type in the previous Figure 2.8.2.1 and remember that this data type allows the access to this data so we can work with their values.

Even though the JDBC specification does not support object-relational databases directly, Oracle interMedia database objects can be used in JDBC programs by means of the interMedia Java Client.

The interMedia Java Client contains high performance proxy Java objects that allow quick object property retrieval and convenient upload/download. The proxies forward any requests for the ORDImage object computation back to the database server.

A schematic diagram of how a database ORDSYS.ORDImage object is related to the Java ORDImage object is shown below on Figure 2.9.1. It is easy to see that ORDImage Java objects are merely proxies for database objects — they must be created from a database ORDImage object.

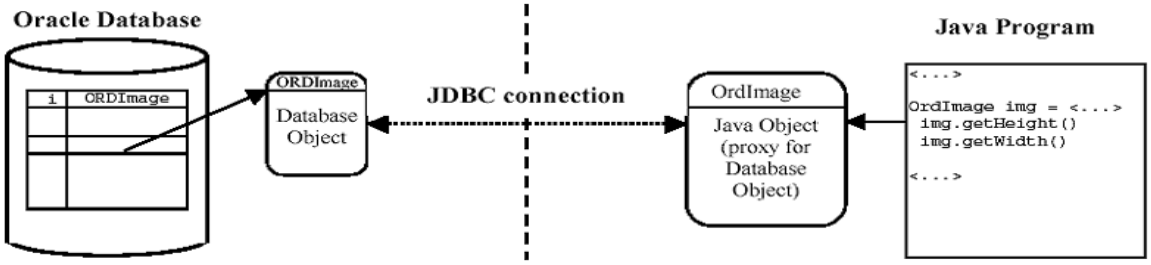


Figure 4.3 - Relation between ORDSYS.ORDImage and Java ORDImage Object [12]

The connection to the database with JDBC is similar to a connection to a normal database unless one detail: Oracle InterMedia uses BLOB columns internally to store data. This implies that if the autoCommit flag must be set to false or any operation that involves BLOB´s will fail. This flag is put to false with the instruction conn.setAutoCommit(false); afterwards create the connection to the database.

4.8 Example of InterMedia & Java

4.8.1 Insert an Image

One of the most important things that we must do in a multimedia database is to insert new data. In this particular case we will insert in the database an image, a copy of it in JPEG format, a thumbnail of the JPEG and the related XML metadata in a separate field this will be his MPEG-7 document. This is the typical problem we will need to solve for the most of the applications with images over a database.

Imagine that we have in our Oracle database a table like this:

ID_IMAGE	IMAGE	THUMB	SIGNATURE
(NUMBER)	(ORDIMAGE)	(ORDIMAGE)	(ORDSINATURE)
…	…	…	…

You can see ID_IMAGE is a numeric value to identify the image that we store. The next fields are special Oracle InterMedia fields, IMAGE is the picture in normal size, THUMB is the thumbnail of the image, and SIGNATURE is the unique mark of the picture, that means each image have a unique identify.

The process of insertion new image data is not very similar to a normal insertion and it consists on three stages:

1. Create a new row in the table. We cannot leave the ORDImages fields empty or subsequent actions will fail. For this reason we must use a creation method provided by Oracle for this data type. A normal insertion statement could be:

```
Insert into IMAGES
        (id, image, thumb, signature)
Values
        (id, ordsys.ordimage.init(), ordsys.ordimage.init(),ordsys.ordimagesignature.init())
```

Figure 4.4 – Insertion a new Row

2. After that we have created a new row but the fields are empty. We must insert or upload the images in the database. For that, first we must obtain the OrdImage field in which we will insert the image executing the instruction:

```
SELECT
        (id, image, thumb, signature)
FROM IMAGES
        WHERE (id = "number_of the image") FOR UPDATE
```

Figure 4.5 – Selection a new row for up date

3. Then, we must create the proxies linked to this images, this operation make us possible to use the pool of methods provided by Oracle for working with images. The proxies is created using:

```
OrdImage imageProxy = (OrdImage)rset.getCustomDatum("image", OrdImage.getFactory());
OrdImageSignature signatureProxy = (OrdImageSignature)rset.getCustomDatum("signature",
OrdImageSignature.getFactory());
```

Figure 4.6 - Creation of proxies linked to new Images

Where rset contains the result of the previous instruction. Then, we have to load the image file using the method of the OrdImage object and the URL of the file as a parameter:

```
imageProxy.loadDataFromFile(nameOfTheImage);
imageProxy.setProperties();
```

Figure 4.7 - Upload Image File to proxy

After that we should use the method setProperties() of the new image. This way interMedia writes the characteristics of the foreign image into the appropriate attribute fields of the OrdImage object, based on a set of characteristics that describes the image properties. With this information, interMedia is able to process certain foreign image formats.

After that we copy and process the image, we generate the signature of the image and we create the thumbnail:

```
signatureProxy.generateSignature(imageProxy);
 //We create the thumbnail
 OrdImage dstimageProxy = imageProxy;
 imageProxy.processCopy("maxscale=200 200 fileformat=jfif", dstimageProxy);
```

Figure 4.7 - Signature creation and Thumbnail creation

Finally, we only need to update the current data in the proxies for the database to be coherent. We can do it with the following code:

```
String updateSQL = "UPDATE images SET image=?, signature=?, thumb=? where (id_image='" +
name_picture+"')";
OraclePreparedStatement opstmt = (OraclePreparedStatement)conn.prepareStatement(updateSQL);
opstmt.setCustomDatum(1, imageProxy);
opstmt.setCustomDatum(2, signatureProxy);
opstmt.setCustomDatum(3,dstimageProxy);
opstmt.executeUpdate();
opstmt.close();
 stmt.close();
```

Figure 4.8 – Last step to insert a new picture

4.8.2 Download a image from the database

The process of downloading an image to a local directory is less complicated than upload to the database. Supposing again that we are working over the table shown in the next Figure, and that we want to download the JPEG image to our file system, the code we should use for this operation is:

```
OraclePreparedStatement stmt;

String query = "select image from IMAGES"+" where id = 1";

 OracleResultSet rset =
                    (OracleResultSet)stmt.executeQuery(query);
 rset.next();
 OrdImage imageProxy =
                    (OrdImage)rset.getCustomDatum(
                                                "image",OrdImage.getFactory());
 rset.close();
 imageProxy.getDataInFile(nameOfTheDownloadedImage);
```

Figure 4.9 – Download a image from the database

Chapter 5. Used Technologies

After all the process concerning to the analysis and design stages, we reach the implementation. As we mentioned before, the chosen language was Java because JDBC (Java Database Connectivity) provides a good API for working with Oracle Databases and this way we can use the Oracle interMedia library for working with the images under Java. Other reason for choosing Java was because provides the Swing library for creating an attractive and easy graphic user interface.

Then, the second decision about the implementation was the development environment. JDeveloper was the chosen IDE (Integrated Develop Environment) and the stronger reasons are because is provided free by Oracle, manages easily a lot of aspects of the database and at the same time has all the features of a Java developer environment (GUI design assistant, debug capability, navigability between classes, code help,...). With this IDE, the programmer can also create all the necessary diagrams from the documentation of a project, following the UML standard.

At last I used the program RapidMiner V4.1 beta 2, to realize the classification model of the pictures.

5.1 Java

5.1.1 Java Introduction

Java is a programming language created by Sun Microsystems and released in 1995 as one of the main components of Sun's Java platform. The language follows the steps of C and C++ in the syntax but has a simpler object model and fewer low-level facilities. Java applications are normally compiled to bytecode. This bytecode can run on any Java virtual machine (JVM) independently what is the computer architecture.

The reference and original implementation Java compilers, class libraries and virtual machines were developed by Sun from 1995. In May 2007, following the specifications of the Java Community Process, Sun made available most of their Java technologies as free software under the GNU General

Public License. Other companies have developed alternative implementations, such as the GNU Compiler for Java and GNU Classpath.

5.1.2 Brief History of Java

In June 1991, James Gosling created the Java language for use in a set top box project. The language was initially called Oak because James Gosling had an oak tree outside his office. After several names, including the name “Green”, it was decided to call the language “Java”, word selected from a list of random words. The goals that gosling wants to obtain were to implement a virtual machine and a language that similar in notation to C and C++.

In 1995 the first implementation of java that appeared in public. That implementation was Java 1.0. It promised "Write Once, Run anywhere" (WORA), providing no-cost runtimes on popular platforms. It was quite secure and its security was configurable: the user was able to restrict network and file access. A short time after, major web browsers incorporated the ability to run secure Java applets within web pages.

In a short time of period, Java became popular. With the arrival of Java 2, new versions had multiple configurations created to answer the new needs of the different types of platforms:

- J2EE was for enterprise applications and the greatly stripped down version.
- J2ME was for mobile applications.
- J2SE was the designation for the Standard Edition.

On 13 November 2006, Sun released much of Java as free software under the terms of the GNU General Public License (GPL). On 8 May 2007 Sun finished this process, making all the code of Java's core open source, except a small portion of code that Sun did not have the copyright.

5.1.3 Main Characteristics

❖ **Platform independence**:

This characteristic lets Java users write programs that must run similarly on any supported hardware/operating-system platform. This target is reached by most Java compilers by compiling the Java language code halfway (Java bytecode). This Java bytecode can be defined as simplified machine instructions specific to the Java platform. After this process, the code is run on a virtual machine (VM). The virtual machine is a program written in native code that interprets and executes generic Java bytecode. This code is written on the host hardware The Java bytecode is interpreted or converted to native machine code by the JIT compiler (look figure 1)

An interpreted virtual machine was used in the first implementations of the language to achieve portability. The disadvantage of these implementations was that produced programs ran more slowly than programs compiled to native executables, like in C or C++ programs. This produced a reputation for poor performance to the language. More recent Java Virtual Machine implementations produce programs that run quite faster than before, using multiple techniques. These techniques are going to be exposed in the next paragraphs.

The first technique that is going to be exposed is the technique known as just-in-time compilation (JIT) with this technique, the Java bytecode is translated into native code at the time that the program is run. This produces a program that executes faster than interpreted code but this has a bad aspect: there is compilation overhead during execution.

To solve this aspect, more sophisticated virtual machines use dynamic recompilation. This technique is based on the idea that the VM can analyze the behavior of the running program and selectively recompile and optimize critical parts of the program. Dynamic recompilation can obtain code more optimized than the one obtained using static compilation.

The third technique that is going to be exposed is the technique commonly known as static compilation. With this technique, the code is directly compiled into native code. This is used by the more traditional compilers. Static Java compilers translate the Java language code to native object code, without the intermediate bytecode stage. One example of computer that uses this technique is GCJ. This method achieves good performance compared to interpretation, but has one weakness: the portability is minor.

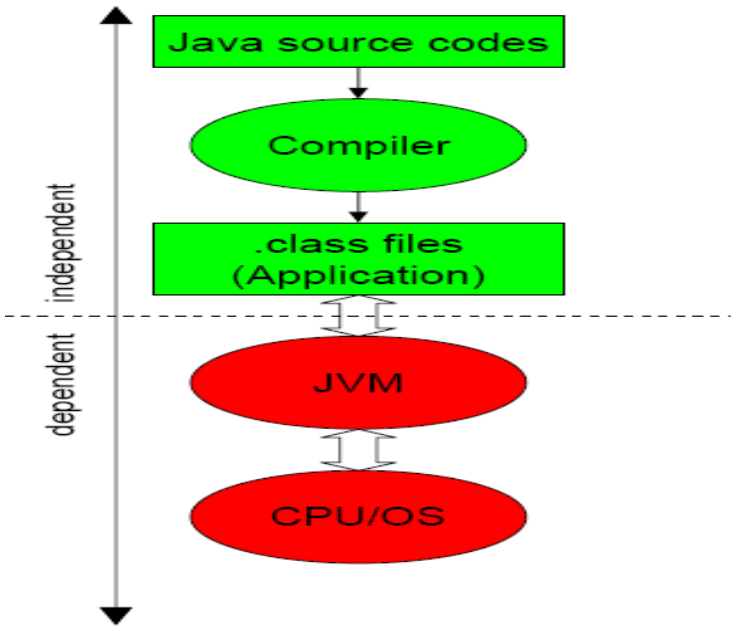


Figure 5.1 - Compilation in Java [5]

❖ **Automatic memory management:**

This aspect let programmers not to be worried about memory management. When the programmer has to allocate memory manually, several "unpleasant" situations can appear: if he forgets to deal locate memory previously allocated or writes code that fails trying to make the task, a memory leak occurs and the program can consume an arbitrarily large amount of memory; if the program attempts to deal locate the region of memory more than once, the result is undefined and the program may become unstable and may crash. Automatic memory management tries to avoid all these kind of problems.

In Java, automatic memory management is realized by automatic garbage collector. The programmer determines when objects are created, and the Java runtime is responsible for managing the object's lifecycle. When there are no references to an object, this one is eligible for release by the Java garbage collector and it may be freed automatically by the garbage collector at any time.

The use of garbage collection in a language can also affect programming paradigms. If, for example, the developer assumes that the cost of allocate memory or free space of memory is low, he may choose to construct objects instead of pre-initializing, holding and reusing them.

Other characteristic about garbage collector in Java is that it is virtually invisible to the developer: developers may have no notion of when garbage collection will take place. This can be an advantage or a disadvantage, depending on the intentation of the application.

To end this aspect, it has to be said that Java does not support pointer arithmetic because the garbage collector may relocate referenced objects. If arbitrary manipulation of pointers is allowed, the safety and security of the automatic memory management wouldn´t be guaranteed.

5.1.4 Advantages

Some of the advantages that the use of Java brings to its users are:

❖ *Java is simple*

- Java has replaced the complexity of multiple inheritances in languages like C++ with a simple structure called interface, and also has eliminated the use of pointers.
- Java uses automatic memory allocation and garbage collection.
- The number of language constructs in Java is small for such a powerful language. The clean syntax makes Java programs easy to write and read.

❖ *Java is Distributed*: this let programmers construct distributed programs in an easy way: writing network programs in Java is like sending and receiving data to and from a file.

- *Java is Portable*
  - The programs implemented in Java can run on any platform without having to be recompiled.
  - There are no platform-specific features on the Java language specification.

- *Java is Architecture Neutral*: this is a consequence of platform independent.

- *Security*: The compiler, interpreter, and Java-compatible browsers contain several levels of security measures to reduce the risk of security compromise, loss of data and program integrity, and damage to system users.

- *Reliability:*
  - Pointers and automatic type conversion, features that are detrimental to program reliability, are avoided in Java.
  - The Java compiler provides several levels of additional checks to identify type mismatches and other inconsistencies.
  - The Java runtime system duplicates many of the checks and performs additional checks to verify that the executable bytecodes form a valid Java program.

- *Multimedia:* Images, Sounds and Animation: JAVA provides extensive multimedia facilities that will enable a programmer to start developing powerful multimedia applications immediately.

- *Networking:* JavaBeans make networking easy to write reassembling components that can be strung together with a minimum of additional coding.

- Java is *Robust:*
  - Early checking for possible errors.
  - Java does not support pointers, which eliminates the possibility of overwriting memory and corrupting data
  - Java has a runtime exception-handling feature to provide programming support for robustness, and can catch and answer to an exceptional situation so that the program can continue its normal execution and terminate successfully when a runtime error occurs.

- Java is *Multithreaded:*

- In Java, multithreaded programming has been integrated into Java, while in other languages, operating system-specific procedures have to be called in order to enable multithreading.
- Multithreading is especially useful in GUI and network programming.

❖ Java is *Dynamic*: A programmer can add to the class new methods and properties without affecting the clients of the class. Also, Java is able to load classes as needed at runtime. [6]

5.1.5 Disadvantages

❖ The main disadvantage is *speed*: An interpreter must first translate the Java binary code into the equivalent microprocessor instruction.

5.2 RapidMiner

RapidMiner (formerly YALE) is the world-wide leading open-source data mining solution due to the combination of its leading-edge technologies and its functional range. Applications of RapidMiner cover a wide range of real-world data mining tasks.

Use RapidMiner and explore your data! Simplify the construction of experiments and the evaluation of different approaches. Try to find the best combination of preprocessing and learning steps or let RapidMiner do that automatically for you.

More than 400 data mining operators can be used and almost arbitrarily combined. The setup is described by XML files which can easily be created with a graphical user interface (GUI). This XML based scripting language turns RapidMiner into an integrated development environment (IDE) for machine learning and data mining. RapidMiner follows the concept of rapid

prototyping leading very quickly to the desired results. Furthermore, RapidMiner can be used as a Java data mining library.

The development of most of the RapidMiner concepts started in 2001 at the Artificial Intelligence Unit of the University of Dortmund. Several members of the unit started to implement and realize these concepts which led to a first version of RapidMiner in 2002. Since 2004, the open-source version of RapidMiner (GPL) is hosted by SourceForge. Since then, a large number of suggestions and extensions by external developers were also embedded into RapidMiner. Today, both the open-source version and a close-source version of RapidMiner are maintained by Rapid-I.

Although RapidMiner is totally free and open-source, it offers a huge amount of methods and possibilities not covered by other data mining suites, both open-source and proprietary ones. You should read the Features page to learn more about the main features of RapidMiner or the Operator Overview for a summary of available operators. It you want to get an idea of the graphical user interface you should have a look at the Screenshots page. Or you can simply download RapidMiner and test it yourself.

The modular operator concept of RapidMiner (formerly YALE) allows the design of complex nested operator chains for a huge number of learning problems in a very fast and efficient way (rapid prototyping). The data handling is transparent to the operators. They do not have to cope with the actual data format or different data views - the RapidMiner core takes care of all necessary transformations. Read here about the most important features of RapidMiner.

RapidMiner (formerly YALE) and its plugins provide more than 400 operators for all aspects of Data Mining. Meta operators automatically optimize the experiment designs and users no longer need to tune single steps or parameters any longer. A huge amount of visualization techniques and the possibility to place breakpoints after each operator give insight into the success of your design - even online for running experiment. On this page we discuss the main groups of operators and give operator examples for each of the groups.

Chapter 6. Analysis and Design

6.1 Architecture

The architecture of the system will be very easy and based on the client-server schema. The innovative characteristics are that, as was explained in the previous page, the program could run as stand-alone or applets fitting the two-tier or three-tier architectures respectively as the pictures show below:

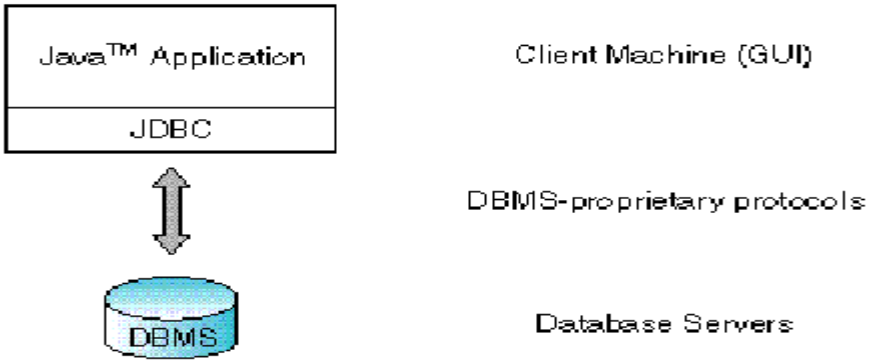


Figure 6.1 – Two-tier Architecture

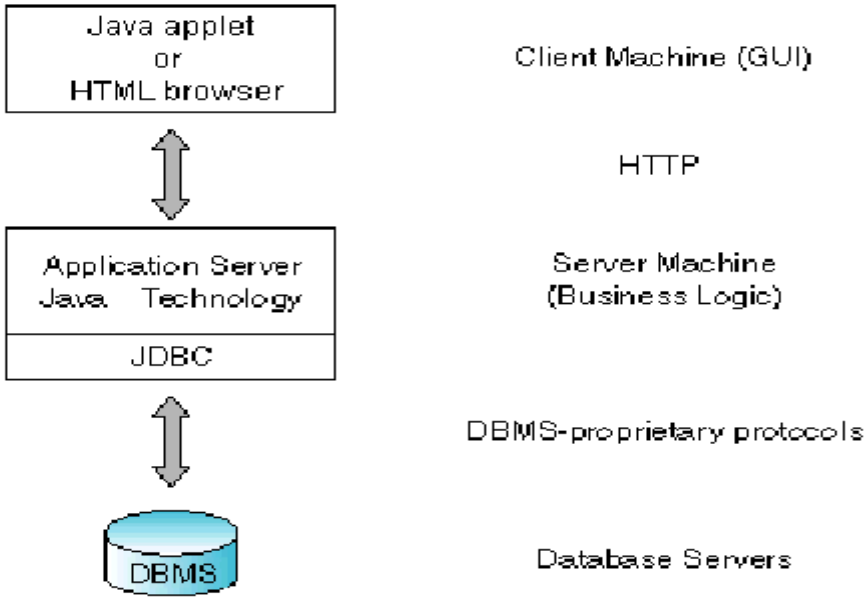


Figure 6.2 – Three-tier architecture

The connection with the database is trough the API provided by Oracle to Java called Java Database Connectivity (JDBC). This API is the industry standard for database-independent connectivity between the Java programming language and a wide range of databases.

6.2 Use Cases

For demonstration purposes of the sample, the application only considers an actor called User as we can see in the following diagram (Figure 6.3). That is because only one person be able to do all the application options.

Depending that the user wants to realize, we can find within the application different experiments such like:

- ***Analysis of images:*** Extract the information of one image in XML documents, where inside of the documents we have the MPEG-7 information of the picture.
- ***Search inside of the data base.***
- ***Two different experiments:*** The experiment-1 are related to the use of MPEG7 and the second one are related to classification of the images.

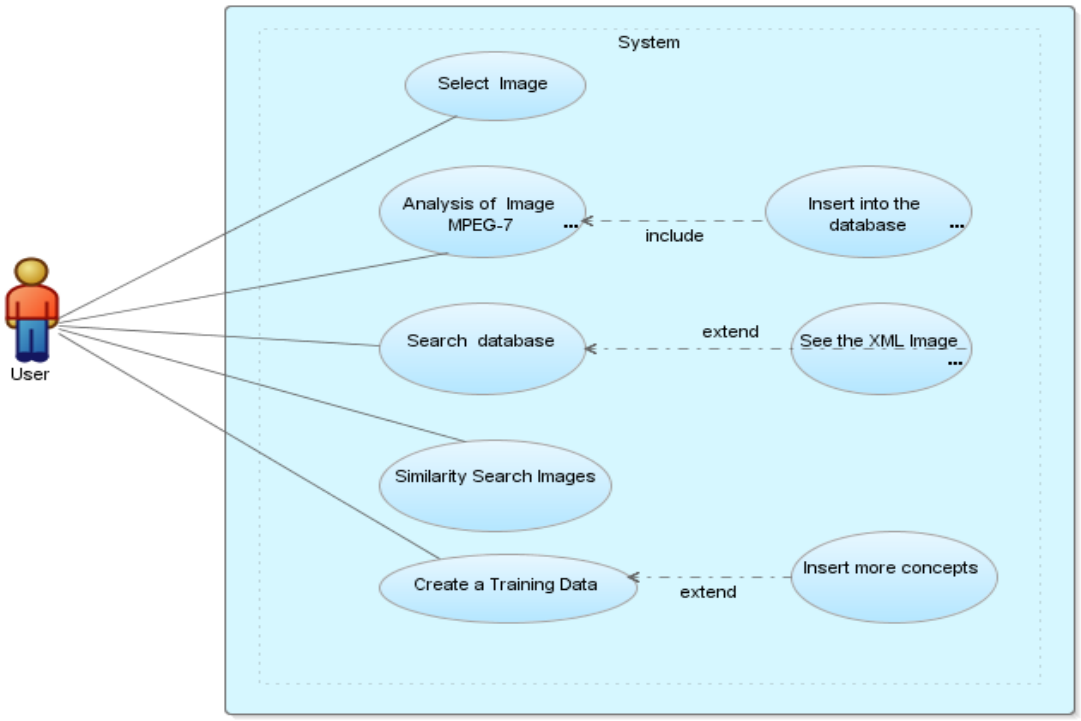


Figure 6.3 - Uses Case

For understand easily how each use case works and the relations between them, the following pages are dedicated to the formal description for all of them.

UC01 – Select Image

Use case -01	**Select Image**	
Description	The user selected a picture from the data set, to be processed by the application	
Normal Sequence	**Step**	**Action**
	1	The user presses the “Image” button in the main frame from the application.
	2	The system shows a File Chooser, with a preview of images.
	3	The user select one image file.
	4	The system take this picture and put in the main frame of the application
Exceptions	**Step**	**Action**
	3‘	If the user selected an image that was in the data base, the system will warn the user
Frequency	Medium	
Significance	Vital	
Urgency	Medium	

Figure 6.4 – UC Select Image

UC02 – Analysis of Image

Use case -02	**Analysis of Image**	
Description	The user choose what kind of analysis need to the picture selected previously and the system create this information and the thumbnail to be put in the data base	
Normal Sequence	**Step**	**Action**
	1	The user press the “MPEG-7“ button in the main frame of the application
	2	The system shows a window with the analysis possibilities.
	3	The user chooses between “Color Descriptors“and “Texture Descriptor“. The user can choose all the options. And press the button “Ok“.
	4	The system will call XM (eXperimentation Model) and will do an xml documents with the MPEG-7 of the picture.
	5	The System go to the UC -03
Exceptions	**Step**	**Action**
	3‘..5‘	If the application doesn’t have an image selected, the system will show to the user an error. UC -01. And the user will have to start again in the UC -01
Frequency	High	
Significance	Vital	
Urgency	Medium	

Figure 6.5 – UC02 – Analysis of Image

UC03 – Insert into the database

Use case -03	**Insert into the database**	
Description	The user after press “Ok“ button from the UC -02, the system will put all the information of the image selected in the data base	
Normal Sequence	**Step**	**Action**
	1	The user press “Ok“ button in the MPEG-7 interface.
	2	The system will prepared the picture and the XML documents to put in the data base. Also, the application will create a thumbnail of this picture.
	3	The system warns us that the picture and the information are in the data base.
Exceptions	**Step**	**Action**
	2‘	If the system has a problem to create the documents or the thumbnail, it will warn us with a message.
	3‘	If the image is already in the data base, the system will warn us and will abort the process.
Frequency	High	
Significance	Vital	
Urgency	Medium	

Figure 6.6 – UC03 – Insert into the database

UC04 – Search database

Use case -04	**Search database**	
Description	The user has the possibilities to search an image and see his xml documents.	
Normal Sequence	**Step**	**Action**
	1	The user press the “Search“ button in the main frame of the application
	2	The system shows to the user a window with a table. This table have the information of all the images in the data base
	3	The user chooses one row of the table. This row means one picture on the database.
	3.1	The user can do double click on the row, the system will show a window with the thumbnail of this image.
	3.2	The user can press the button “XML“ to see the documents MPEG7 of this image. <<Extension Point UC-05>>
	3.2.1	The system will show a window to see the XML document and with different options to see the different types of documents
	3.2.2	The user will select some or all option, and press the button “ok“.
	3.2.3	The system will show to the user the document XML for the picture selected.
Exceptions	**Step**	**Action**
Frequency	Low	
Significance	Medium	
Urgency	Low	

Figure 6.7 - UC04 – Search database

UC05 – See the XML image

Use case -05	**See the XML image**	
Description	The user wants to see a XML document from an image.	
Normal Sequence	**Step**	**Action**
	1	The user can press the button “XML“ to see the documents MPEG7 of this image.
	2	The system will show a window to see the XML document and with different options to see the different types of documents
	3	The user will select some or all option, and press the button “ok“.
	4	The system will show to the user the document XML for the picture selected.
Exceptions	**Step**	**Action**
	1‘	If the user didn’t select an image from the table, the system will show a warning.
	3‘	If the user doesn’t select any option, the system will show nothing to the user.
Frequency	Medium	
Significance	Low	
Urgency	Medium	

Figure 6.8 - UC05 – See the XML image

UC06 – Similarity search Images

Use case -06	**Similarity Search Images**	
Description	The user, choose an image, so that the system apply one algorithm to tell the user which one of the photos that are the data base approaches to the elected image.	
Normal Sequence	**Step**	**Action**
	1	The user press “Experiment 1“ in the menu of the main frame.
	2	The system will show an “Experiment 1“ window.
	3	The user chooses one image from the data base.
	3.1	The user press “Preview“ button.
	3.2	The system shows to the user a window with all the thumbnails from the images in the data base.
	3.3	The user select one image with double click
	4	The user press the “Search“ button
	5	The system put all the information in two tables, one is the results using oracle tools and the other is using MPEG-7 coefficients.
	5.1	The user can see the pictures, if he do double click on the row of the table,
	5.2	The system will show a selected image.
Exceptions	**Step**	**Action**
	2‘	
	3‘	
Frequency	High	
Significance	Vital	
Urgency	Medium	

Figure 6.9 - UC06 – Similarity search Images

UC07 – Create a training data

Use case -07	**Create a training data**	
Description	The application will create a classifier model with the user parameters	
Normal Sequence	**Step**	**Action**
	1	The user presses the “Experiment 2“ button in the menu of the main frame.
	2	The system shows the “Experiment 2“ window.
	3	The user selects the tab “Get Document“.
	4	The user choose one concept to do the training data
	5	The user press the button “Create“.
	6	The system will create a document with the data.
Exceptions	**Step**	**Action**
	2‘	
	3‘	
Frequency	High	
Significance	Vital	
Urgency	Medium	

Figure 6.10 - UC07 – Create a training data

UC08 – Insert concepts

Use case -08	**Insert concepts**	
Description	A tab to put more information on one picture, to create later a classifier model.	
Normal Sequence	**Step**	**Action**
	1	The user presses the “Experiment 2“ button in the menu of the main frame.
	2	The system shows the “Experiment 2“ window.
	3	The user selects the tab “Training data“.
	4	The user has to select an image in the combobox and select the attribute to add to the image.
	5	The user press the button “Add Attribute“.
	6	The system storage the attribute in the data base and show the attributes previously on the picture.
Exceptions	**Step**	**Action**
	2‘	
	3‘	
Frequency	High	
Significance	Vital	
Urgency	Medium	

Figure 6.11 - UC08 – Insert concepts

6.3 Database Design

Now we can define the structure created for the database. We need to store mainly the data about the images, the descriptions and the features with their images. The compact view of the entity-relationship diagram is as follows in Figure 6.10:

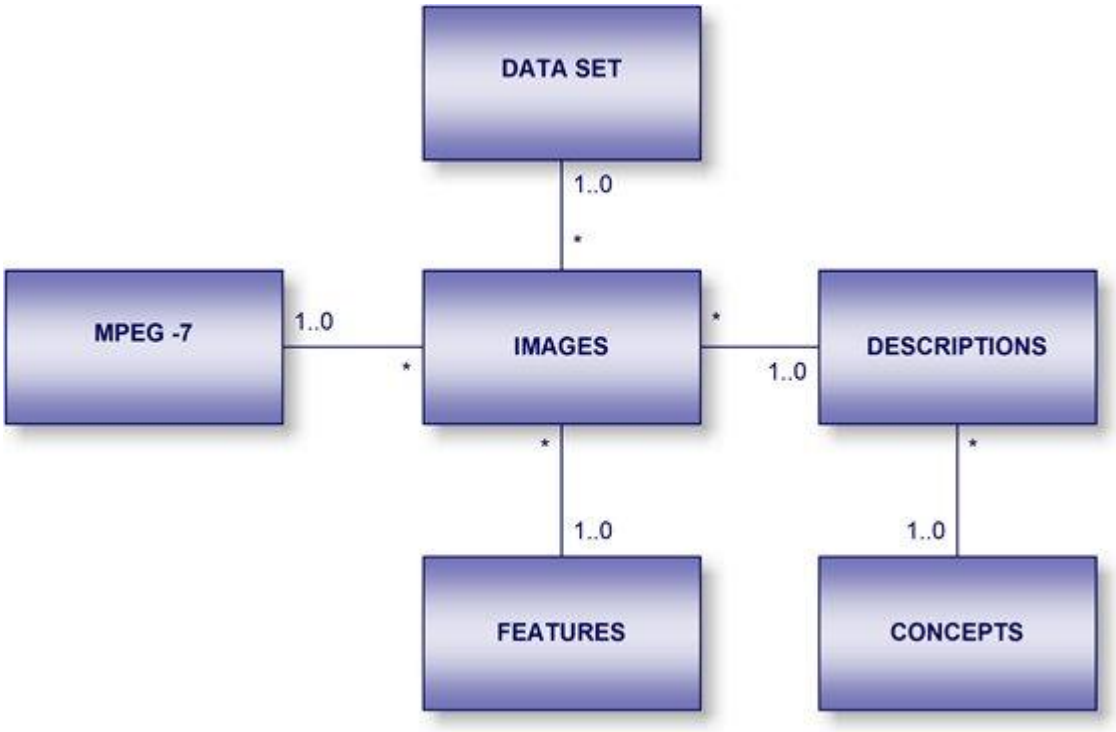


Figure 6.12 – Compact view of the Database

The main idea is when the user load a image in the program, the application get this image and know that data set comes. Each image has different concepts and you can see this relation in descriptions. Also, each image has different MPEG-7 documents about color and texture. The coefficients of the documents are store in features.

The main entity is image, there we will put the image and the thumbnail with Oracle Intermedia. We can see a more detailed view of this database structure with all the attributes and their types in the next diagram (Figure 6.13). We can take a look also to the primary and foreign keys explained later.

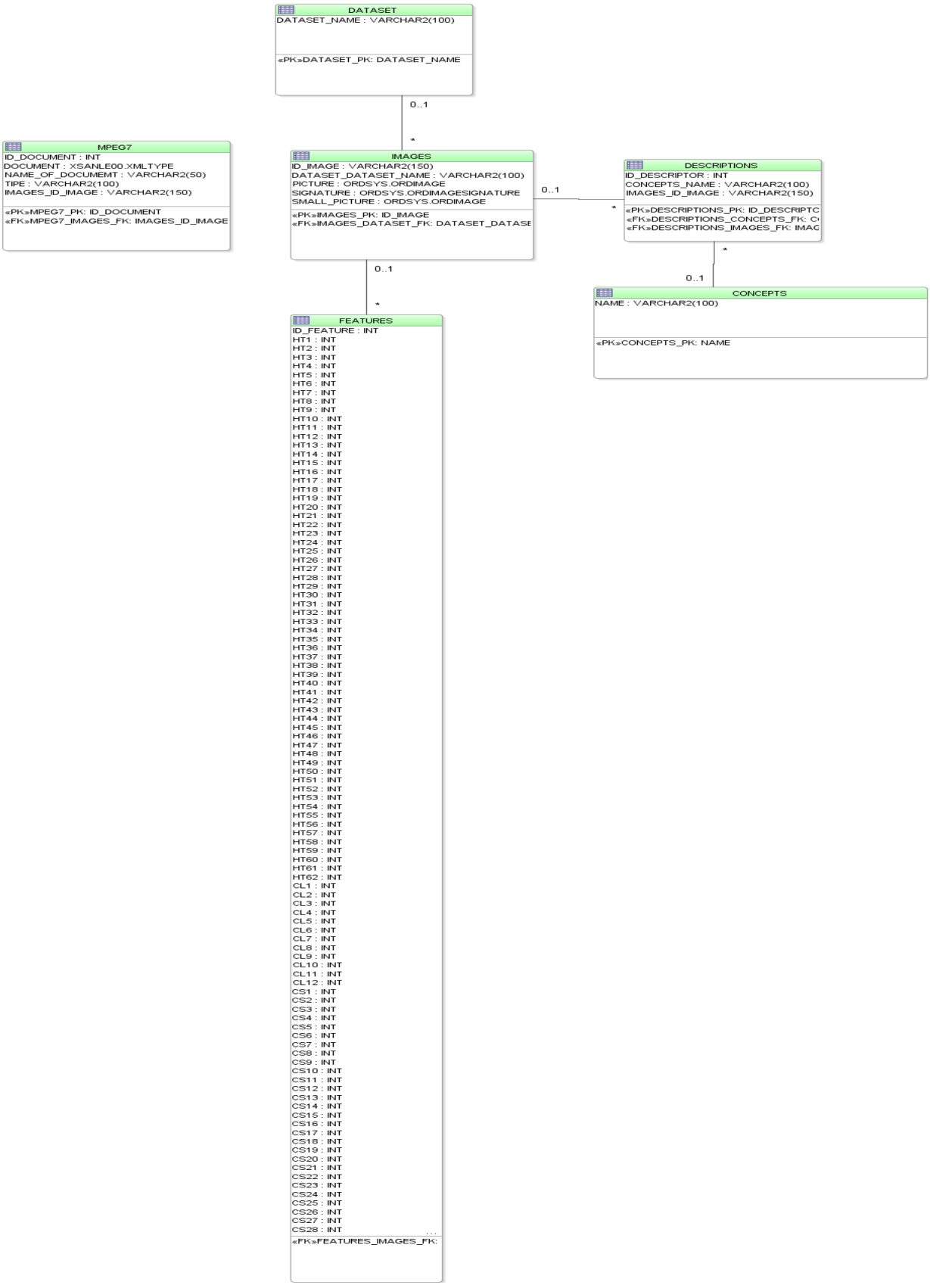


Figure 6.13 – Design of the Database

6.3.1 Description of Tables

I am going to explain the function of each table and why I elected that primary key, a little description of each one:

- The tables DataSet contain the datas of the dataset where we put the images, to have a classification of our images.

Table: DATASET			
Attribute	**Type SQL**	**Long**	**Domain**
DATASET_NAME	VARCHAR	100	ALPHANUMERIC
Primary Key	**DATASET_NAME**		

- The table Images contain the picture and his thumbnail, the primary key is the name (path) of the picture, and the last attribute "signature" is necessary to use the oracle tools, means like a primary key of the picture.This table has a foreign key is dataset.
-

Table: IMAGES			
Attribute	**Type SQL**	**Long**	**Domain**
ID_IMAGE	VARCHAR	150	ALPHANUMERIC
DATASET	VARCHAR	100	ALPHANUMERIC
PICTURE	ORDSYS.ORDIMAGE		
SIGNATURE	ORDSYS.ORDSIGNATURE		
SMALL_PICTURE	ORDSYS.ORDIMAGE		
Primary Key	**ID_IMAGE**		

- The table MPEG7 contain the xml documents that we created, to use this documents we used XMLType. The primary key is id_document this integer is manipulate by the system, and the foreign key reference the image in "images".

Table: MPEG7			
Attribute	**Type SQL**	**Long**	**Domain**
ID_DOCUMENT	INT		NUMERIC
DOCUMENT	XMLType		
NAME_OF_DOCUMENT	VARCHAR	50	ALPHANUMERIC
TYPE	VARCHAR	100	ALPHANUMERIC
IMAGES_ID_IMAGE	VARCHAR	150	ALPHANU;ERIC
Primary Key	**ID_DOCUMENT**		

- The table concepts contain all the possible descriptions of the pictures. The primary key is the name because is only one.

Table: CONCEPTS			
Attribute	**Type SQL**	**Long**	**Domain**
NAME	VARCHAR	100	ALPHANUMERIC
Primary Key	**NAME**		

- The table Descriptions is the relation between images and concepts, one picture can have a lot of concepts on the picture, this table have two foreign keys, the first one is for images, and the second is for concepts. The primary key is a integer is that manipulate by the system

Table: DESCRIPTIONS			
Attribute	**Type SQL**	**Long**	**Domain**
ID_DESCRIPTOR	INT		NUMERIC
CONCEPTS_NAME	VARCHAR	100	ALPHANUMERIC
IMAGES_ID_IMAGES	VARCHAR	150	ALPHANUMERIC
Primary Key	**ID_DESCRIPTOR**		

- The table features is the biggest table here, has 108 attributes, 62 coefficients from HomoTexture, 12 coefficients for ColorLayout and 28 coefficients from ColorStructure. The primary key is a identifier of document, and has one foreign key to reference images.

Table: DESCRIPTIONS			
Attribute	**Type SQL**	**Long**	**Domain**
62, HT	INT		NUMERIC
12, CL	INT		NUMERIC
28, CS	INT		NUMERIC
ID_FEATURE	INT		NUMERIC
IMAGES_ID_IMAGE	VARCHAR	150	ALPHANUMERIC
Primary Key	**ID_FEATURE**		

6.4 Classes Description

In this chapter it is going to be exposed the different elements that works together to manage the goals expressed in Chapter 1, based on the theoretic aspects developed in the next chapters. This section will be divided in several sections, one per class existing class of the system. On the other hand, all this subsections has 3 parts: Overview, where the main function and the role of the described class is exposed; Attributes, where the attributes of the actual class are developed; and Methods, section in which the methods of the class are explained. I only explain the most important classes and the main attributes to be easier for the reader.

- ✓ **Main Frame:** It is the main interface of my application, this IU have the responsibility of show historical of the operations, show the chosen image, and management of the menu and buttons.
  - Attributes:
    - *DefaultComboBoxModel dcm :* Is the model of ComboBox, where appear how many datasets we have.
    - *DefaultListModel dcm:* Is the model list of Jlist, where appear the historical of the program.
    - *String name_picture:* We put here, the path of the image selected.
  - Methods:
    - *void load_image(String direccion,ImageIcon img):* This method put on the main frame the selected image.
    - *void information(String info):* This method put on the jList the operations done. To create a historical of the operations.

- ✓ **DBAccess:** With this class we can connect to the data base and do all the operations against it.
  - Attributes:
    - *String typeConexion :* The conexion driver. In our case is jdbc.
    - *String server :* The conexion address.
    - *String user :* The user of the database.
    - *String pass:* And the password to use the database.
  - Methods:

- *ResultSet conect(String query) :* To realize a query against the data base.
- *void close_conexion(Connection conn):* To close the data base session.
- *boolean exist_concepts() :* This method is to init the data base when is our first time using the application
- *void init_dataset()*: When the data base is empty we use this method to do the initialization.
- *void insertElement(String name_picture, String XMLLayout,String XMLColor,String XMLTexture,int number,String values_CS,String values_CL,String valuesHT,int ctrl)* : This method insert into the database the picture and all the mpeg7 information. The values are ColorLayout,ColorStructure and HomogeneousTexture.
- *void insert_image (String name_picture):* To insert the image on the database with the Oreacle InterMedia tools.
- *void insert_descriptions(String name):* To insert the description of each picture.
- *void insert_concepts(String word):* To insert the possible descriptions of the images.
- *void insert_features(String name_picture,String values_SC,String values_CL,String values_HT):* This method insert the coefficients of the mpeg7 documents in the data base.
- *ImageIcon createImageIcon(OrdImage ImgObj):* This method gives me back an ImageIcon when have an OrdImage.
- *OracleResultSet GetImageBD():* I used this method when I need to get one image in the data base.
- *InputStream getDBInputStream(OrdImage imgObj):* This method gives me back an InputStream when I have an OrdImage. Treatment of images.
- *BufferedImage getDBInputStream2(OrdImage imgObj):* This method is similar to previously method, this method give back a BufferedImage. I used this when I want to put an image in the class JImageFrame.
- *ResultSet getConcepts(String name):* The application calls this method when want to know the concepts of one picture.

✓ **DBImage:** Represents image in database, is easier management this instances.

- Attributes:
  - *BufferedImage thumb:* I put here, the thumbnail image.

- *BufferedImage image :* The same, but with regular image
- *String desc:* The description of the image.

- Methods:
  - *BufferedImage getImage():* This is the get method from thumb attribute.
  - *BufferedImage getImageNormal():* This is the get method from image attribute.
  - *String getDesc():* This is the get method from desc attribute.

✓ **JImageWidget:** Visual component contains thumbnail of image and reference to database, I used this class to create a visual field for the images.

- Attributes:
  - *DBImage dbImage:* This attribute represent a link to the database, if we want to change something on the data base.
  - *JImagePanel imagePanel:* Is the panel where will be the image.
  - *Experiment2 mainFrame:* This attribute is the reference to Experiment1 class, because this class is used by Experiment1 and Experiment2. To do a difference.
  - *Experiment1 mainFrame2*: This attribute is the reference to Experiment2 class, because this class is used by Experiment 1 and Experiment 2. To do a difference.
  -
- Methods:
  - *ImageIcon Create_Image():* With the OrdImage of DBImage this method give back an ImageIcon.
  - *DBImage get_dbImage():* This is the method get to DBImage.
  - *void set_mainframe(Experiment1 s):* To know how is calling this instance
  - *ByteArrayInputStream BufferedImageToInputStream(BufferedImage bufferedImage) :* This method converts a BufferedImage to ByteArrayInputStream.

✓ **JImagePanel:** extends javax.swing.JPanel, this class is a Panel encapsulating image.

- Attributes:
  - *BufferedImage image*: image this is a inner image of panel
- Method:

- *void paintComponent(Graphics graphics):* Overrided method for painting inner image

✓ **JImageFrame(BufferedImage image):** Extends JFrame, This is almost fullscreen JFrame with some BufferedImage on top.

- Methods
  - *centerMe(Component component)*: Center any component (dialog, frame) in screen.
  - *void Init(BufferedImage image)*: Create Creates new form JImageFrame and later put the image icon in JLabel.
  - *void ByteArrayInputStream BufferedImageToInputStream(BufferedImage bufferedImage):* Method for convert BufferedImage to ByteArrayInputStream

✓ **XMLSelected:** This IU is the window where the user selected the mpeg7 codification option (ColorLayout, StructureColor, and Homogeneous Texture).

- Attributes:
  - *String values_SC:* Values for Structure Color, they are the coefficients after the processing of the XML document.
  - *String values_CL:* Values for Color Layout, they are the coefficients after the processing of the XML document.
  - *String values_HT:* Values for Homogeneous Texture, they are the coefficients after the processing of the XML document.
- Methods:
  - *void change_string(String name):* Manipulate the path of the image to use later in the file list (Yellowstone.lst).
  - *void change_list(String name):* This method put inside of the list the image to process.
  - *void start_code():* This method is the beginning of the execution of image analysis by eXperimentation Model.
  - *void execute_xperiment():* This method call to process of eXperimentation Model.
  - *void init_files():* This method clean the files to future uses. This files are the xml files of Color descriptor and Texture descriptior.

- ✓ **XMLParser:** This class realize two important things, first one get of the file the XML document and the second thing the instance realize a parser of XML documents
    - Attributes:
        - *String XMLCL:* This attribute have the XML document of ColorLayout after read the file.
        - *String XMLSC:* This attribute have the XML document of StructureColor after read the file.
        - *String XMLHT:* This attribute have the XML document of HomogeniousTexture after read the file.
    - Methods:
        - *String parser_fichero (String XML):* This method realize a parser the XML document of ColorStructure.
        - *String parser_fichero_layout(String XML):* This method realize a parser the XML document of ColorLayout.
        - *String parser_fichero_texture(String XML):* This method realize a parser the XML document of HomogeneousTexture.
        - *String give_XMLCL():* This is the get method to get XMLCL attribute
        - *String give_XMLHT():* This is the get method to get XMLHT attribute
        - *String give_XMLSC():* This is the get method to get XMLSC attribute
- ✓ **XQuery:** This class realize
    - Attributes:
        - *DefaultComboBoxModel dcm*
        - *TModel2 model_table :* The model to manipulate the jTable1.
        - *Control_Table2 ctrl:*
    - Methods:
        - *void load_combox()*
        - *void search()*
- ✓ **XperimentModel:** Extends Thread, and have to call the process of eXperimentation Model, is a thread because the application has to wait this process to go on with the program.

- ✓ **TModel,Tmodel2,Tmodel3:** This classes are model for tables, all of them implements TableModel, and I used it to create custom tables
  - Methods:
    - *int getColumnCount():* Give back the number of columns of the model.
    - *int getRowCount():* The same but with rows.
    - *Object getValueAt(int rowIndex, int columnIndex):* This method give me the object value inside of the table, referred by a row and a column
    - *Object getPicture(int rowIndex):* This method gives the object in the selected row.
    - *void deletePicture (int fila):* Delete the object in the row position.
    - *void insertpicture (DataPicture newpicture):* Introduce a new element in the table model.
    - *String getColumnName(int columnIndex):* Give back the name of the column.
    - *setValueAt(Object aValue, int rowIndex, int columnIndex):* Change the value of an object.

- ✓ **ControlTable,ControlTable2,ControlTable3**: These classes control the previous classes
  - Methods
    - *void borraFila ():* Delete one row of the table model.

- ✓ **InitDataset():** This class init the database when is first time to execute, that means when the data base is empty.

- ✓ **Experiment1:** This class management the experiment1, also is IU where are all the visual components.
  - Attributes:
    - *DefaultComboBoxModel dcm:* It is the model of ComoboBox.
    - *TModel3 model_table2:* The model to manipulate the jTable1.
    - *TModel2 model_table:* The model to manipulate the jTable2.
    - *Control_Table2 ctrl2:* This control the table model.
    - *Control_Table3 ctrl3:* This control the table model 2.
  - Methods:

- *void load_combox():* This method initializes the "dcm", and put inside the name of all the images in the data base.
- *void search():* This method management the algorithm to find similarity images. Only the MPEG7 part.
- *void Search_oracle(String name_picture):* This method management the algorithm to find similarity images. Only the Oracle InterMedia part.
- *void put_preview(ImageIcon a,String desc):* this method put on the IU the preview of the selected image.

✓ **ExperimentII:** This class management the experiment2, also is IU where are all the visual components.

- Attributes:
  - *String cadena:* To put the concepts of an image.
  - *DefaultComboBoxModel name_pictures:* Model of ComboBox where all the names of images in the data base are.
  - *DefaultComboBoxModel attributes:* Model of ComboBox where all the attributes of images in the data base are.
- *Methods:*
  - *void load_combox():* This method initializes the ComboBox models, and put inside the name of all the images in the data base and the concepts.
  - *void update_concept(String name):* this method changes the concepts if we change the image.
  - *void update_image():* This method update the image preview if we change the selected image.

✓ **IDataBase:** IU of "search", this class shows you the images of the database, and you can see the XML of each image.

- Attributes:
  - *TModel model_table:* The model to manipulate the jTable1.
  - *Control_Table ctrl:* This control the table model.
- Methods
  - *void update_table():* This method updates our jTable with the data base datas.

- ✓ **DataPicture:** This class has the structure of an image, his name and the XML documents that define this image.
    - Attributes:
        - *String nameP:* Name of the picture
        - *String XMLColor:* The XML document for ColorLayout.
        - *String XMLStructure:* XML document for ColorStructure.
        - *String XMLTexture*: XML document for HomogeneousTexture.
        - *String DataSet*: The name of the data set of the image.

    - Methods
        - *String GetName():* This is the get method of Name attribute
        - *void SetName(String name):* This is the set method of Name attribute
        - *String GetXMLColor():*This is the get method of XMLColor attribute
        - *String GetXMLStructure():*This is the get method of XMLStructure attribute
        - *String GetDataset():*This is the get method of XMLTexture attribute
        - *String GetXMLTexture():*This is the get method of Dataset attribute

- ✓ **The_search:** This class takes care about the algorithm of quadratic form distance, and gives back the distance result.
    - Attributes:
        - *String my_name:* Name of the selected image to compare with the rest of the pictures.
        - *String my_HT:* Coefficients of Homogeneous Texture of the selected image to compare with the rest of the pictures.
        - *String my_CS:* Coefficients of Color Structure of the selected image to compare with the rest of the pictures.
        - *String my_CL:* Coefficients of Color Layout of the selected image to compare with the rest of the pictures.
        - *String HT:* Coefficients of Homogeneous Texture of an image to be compare with the image selected.
        - *String CS:* Coefficients of Color Structure of an image to be compare with the image selected.
        - *String CL:* Coefficients of Color Layout of an image to be compare with the image selected.

- *int resultCS:* Will be the result of the calculations of ColorStructure.
- *int resultCL:* Will be the result of the calculations of ColorLayout.
- *int resultHT*: Will be the result of the calculations of homogeneousTex*ture*
- *int totalresult:* The distance result of one image and the selected image.

- Methods
  - *public void check_CS():* Method to accomplish calculations of ColorStructure.
  - *public void check_CL():* Method to accomplish calculations of ColorLayout.
  - *void check_HT():* Method to accomplish calculations of HomogenenousTexture.
  - *void check_total():* This method will call the rest of the method, and give back the total result.
  - *int get_resultCS():* This is the get method of resultCS attribute.
  - *int get_resultCL():*This is the get method of resultCL attribute.
  - *int get_Totalresult():* This is the get method of Totalresult attribute.
  - *String get_Name():* This is the get method of Name attribute.

Chapter 7. The Experiments

In this chapter we will see how the experiments work. Inside of the application you can choose between two experiments, the first one is about image analysis with eXperimentation Model and Oracle InterMedia. And the second one is about classification using RapidMiner.

7.1 Experiment of Similarity Search

I'm going to divide this experiment in two, one of them is about find how many images are exactly the same on the database and the second one is if the application can find similar images.

7.1.1 Finding the twin images

We have for this experiment 25 images are in the data base, I put two equal images with different names, and you can see the picture in the figure 7.1:

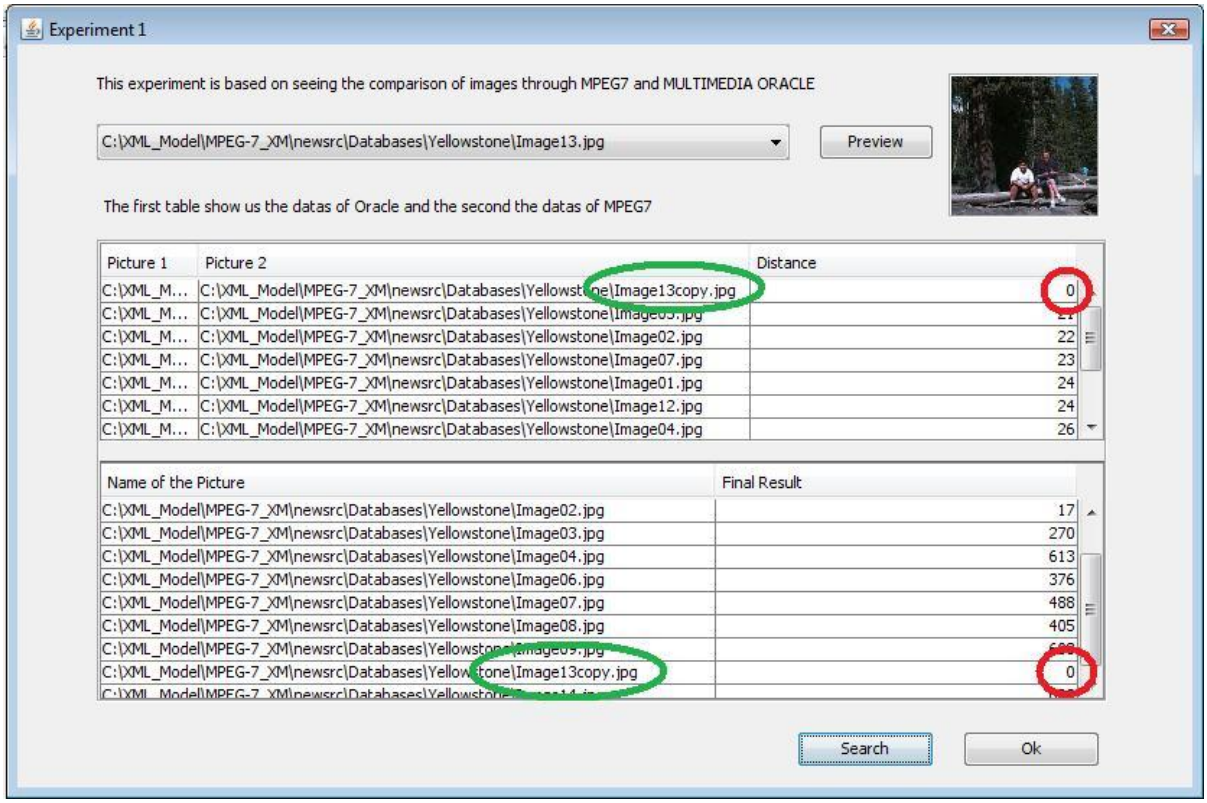


Figure 7.1 – Example of experiment, twin Images

In all moment, the user can do double click in the table and immediately you can see the preview of this picture. Figure 7.2 are the twin images:

Figure 7.2 – Twin Images

You can observe in the table, the program recognize both pictures as the same picture, the result means that between the first image and the second one the distance is 0, so both pictures are equal. In all moment, the user can do double click in the table and immediately you can see the preview of this picture.

7.1.2 Finding the similar image

In this case we have two images they are not equal but yes similar, for example two image from the same thing but different angles of view (see Figure 7.3):

Figure 7.3 – Similar Images

If we select the image47.jgp (or Image 46.jpg, it's the same because the result will be the same) and then push the button "search", will appear in the table the results of the distances. See Figure 7.4:

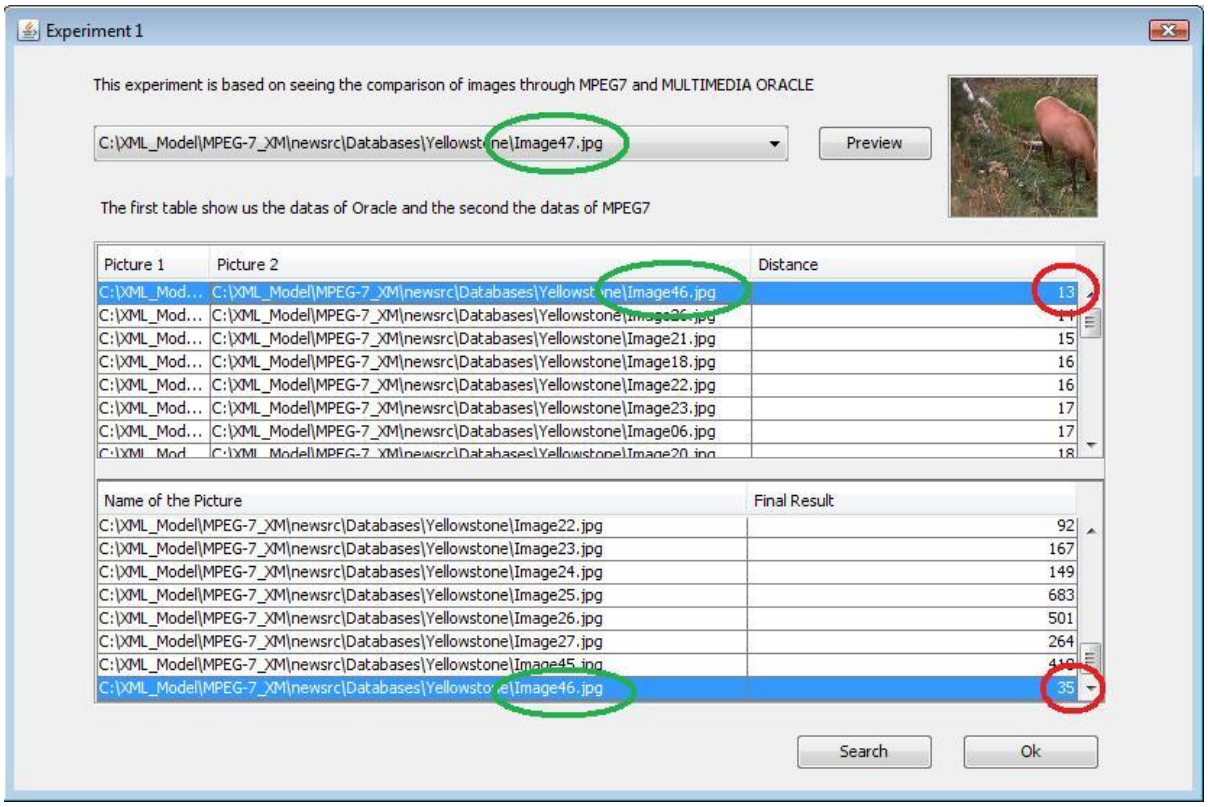


Figure 7.4 – Example of experiment, similar Images

The values are not the same, but this is because both methods using different coefficients to evaluate the images. We can observe the results of the search in the table, the lowest value means than this images is nearest to the selected image.

7.2 Experiment of Classification

I will do this experiment with 27 images on the database, each picture has some keywords to describe it. I choose the label animal and want to know how many pictures have an animal (for example), and made some predictions.

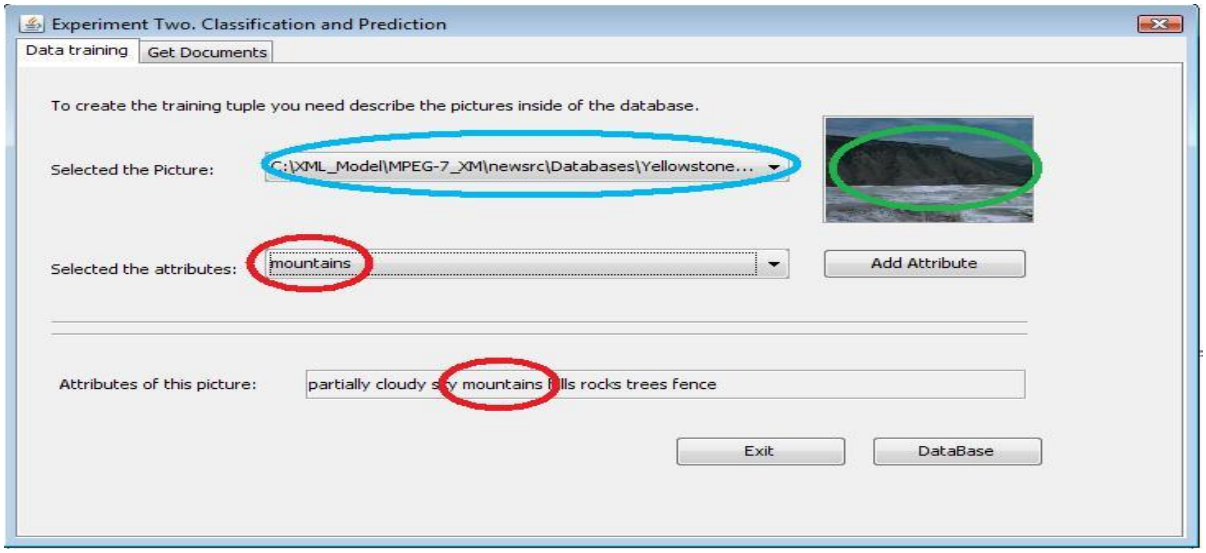


Figure 7.5 – First tab of Example, Experiment two.

In the figure 7.5 we can see the first step, if we want to insert into the descriptions some attribute on an image. In the second step we can create the document but before we have to choose the label attribute. The document will have a "1" if the image has the label attribute and "0" if not.

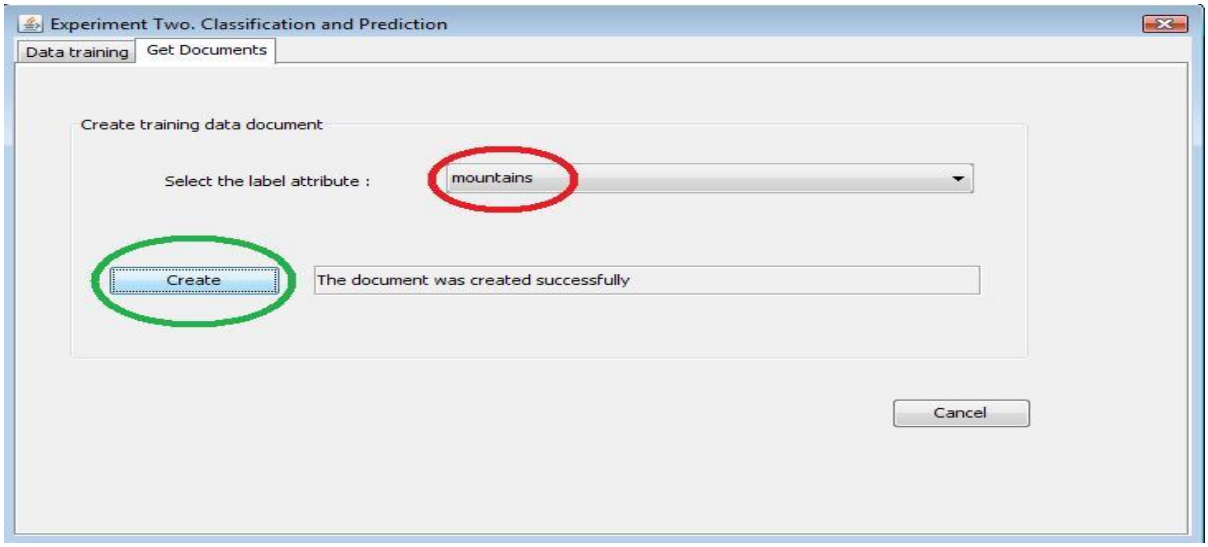


Figure 7.6 – Second tab of the Example, experiment two.

When we have the document we can use the RapidMiner application, in this program we will use SVM with XValidation.

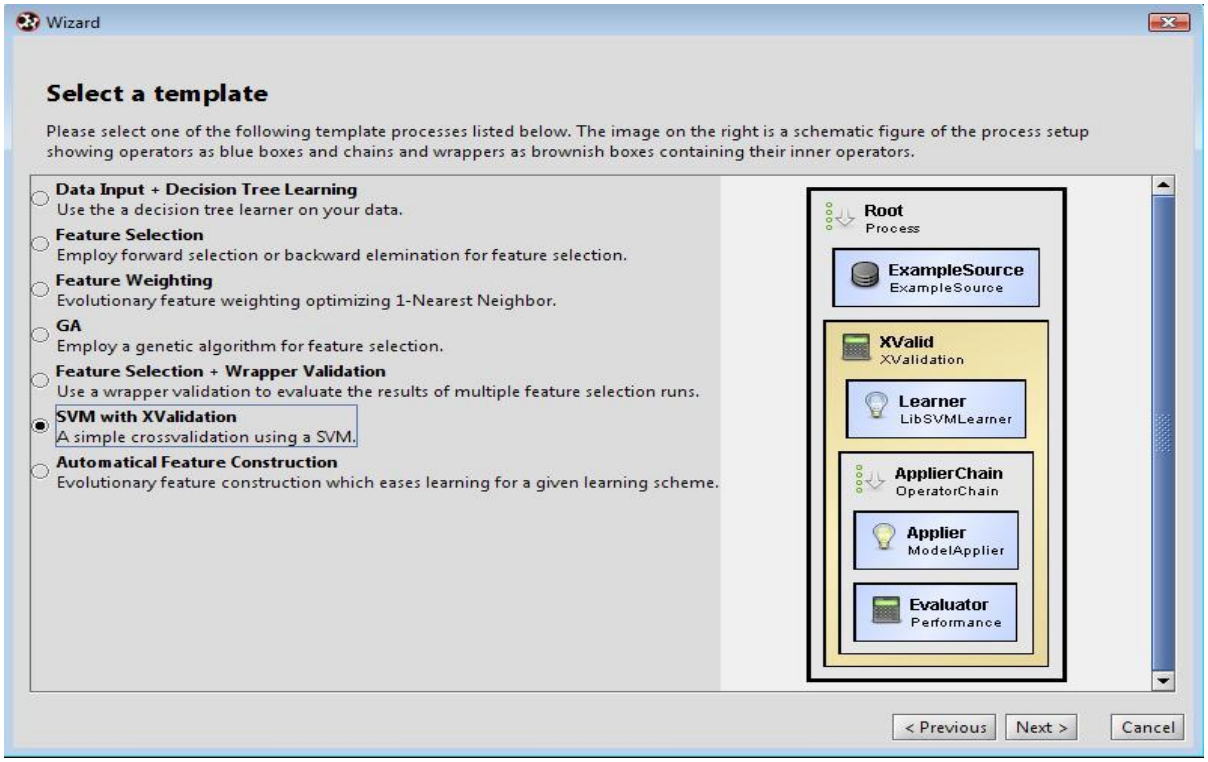


Figure 7.7 – Using Crossvalidation in RapidMiner

The program do an analysis of the document and give us the result in a confusion matrix (you can read the theoretic aspects in previous chapters). For this example the program gives me this result:

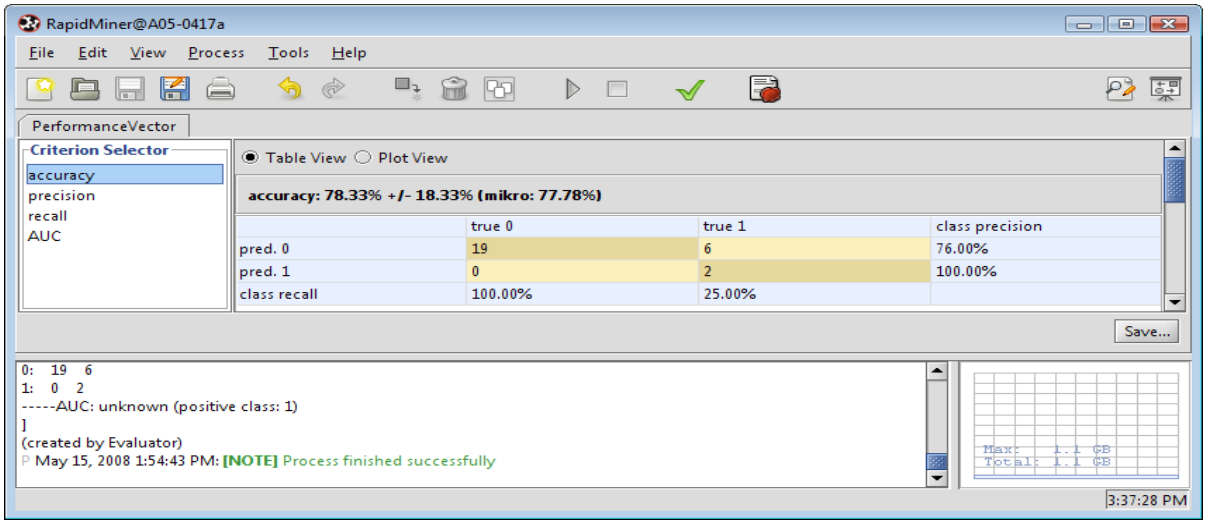


Figure 7.8 – Confusion Matrix

See the RapidMiner manual at the end of the chapter-9 if you want to know how this program works and his options. If you want to know more about RapidMiner look up the last chapter "RapidMiner Manual".

We can observe the results of the matrix confusion, the accuracy of this model is 78.33%, this result will be better if in the database are a lot images. The columns are the predicted class, and the rows are the actual class. We have two possible values in the actual class, "1" means that the label attribute is true and "0" when the label attribute is negative.

In the negative row (0), the first cell is called true negative and the value 19, the next one is the false positive and the value is 6, whit a class precision of 76%. In the positive row (1), the value false negative is 0 and true positive 2.

Chapter 8. Conclusions

8.1 Introduction

In this chapter I explain the principal conclusions extracted from the process of development of the project. Also the future research topics at this field are intended to improve this project.

8.2 Success of achieving the objectives

In the final part, I review the steps followed for covering all the purposed goals in the specification of this Master's Thesis and remark the successes reached in each stage.

As the project has been able to check, if development and reasoning of application have been followed, from the beginning the objectives of the project explained in the chapter one of this memory have been present.

- I have had to increase my knowledge for the realization of this project in management of images (the eXperimentation Model tool) and the relation with the database (Oracle InterMedia).
- Knowledge about MPEG-7, similarity search and cassification. And realized two experiments that using these theoretic topics.
- I have deepened into the use of several fields like SQL and Java, getting better my knowledge of databases and techniques of processing images, developing the ability for the implementation of the program. Besides the tools that I have used for the creation of application.
- Easy and intuitive interface of the application to the user. The user can use this tool without knowledge of the theoretic concepts.
- I take care that the program does not provoke critical failures that put in danger the integrity of the managed data.

Definitively, the realization of the project has served like approximation my future labor life, requiring one point over maturity for your conclusion.

8.3 Surged Inconveniences

At the beginning of the project development, I had some difficulties of knowing the environment that should apply this work. These difficulties disappeared with my tutor's collaboration.

8.4 Conclusion of MPEG-7

MPEG-7 is a standard that it allows us to define contents of multimedia character. You have infinity of benefit in a world where technologies are more advanced and users have to access them, doing that the quantity of information (in this audiovisual case) growth in exponential way.

It should be understood that also this standard has like end to provide a common language when we define multimedia documents. This is not a standard for the search, indexation or recovering of multimedia contents, although they will see that this tool simplifies the use of standards.

MPEG-7 is able to play an important role towards standardized enrichment of multimedia with semantics on higher abstraction levels to improve the quality of query results. However, the complexity of the description schemes makes it sometimes difficult to decide which kind of semantic descriptions have to be used or extended. This may lead to difficulties when interchanging semantic meta-data with other applications. Nevertheless the standardized description language is easy to exchange and filter with available XML technologies. Additionally the Web-based Tools are available on different platforms and could be extended with further components according to the usage of standardized API´s, Client/Server Technologies and XML based Communication. Furthermore the system architecture allows you to break the communication of any web agent to support user specific retrieval specifications.

Weak points:

- Language DDL.
- Integration software Ds and DSs.
- Tools of System and Control.
- Encoding and transportation data MPEG 7.
- Test in real applications.

8.5 Future Work

The future development of the implemented program could be big, because this research field is by now innovative and always appear new tools to manage and analysis of the images.

Some of these improvements could be:

- Facial recognition: This topic is very interesting and has infinite possibilities and applications. It is important in fields like security or medical (for example) the application recognizes a person and takes the visual information about him in the data base.
- Similarity search: The application can use another algorithm described in the literature of similarity search to do better the accuracy of the results.
- New interfaces: Maybe it will be necessary to access this application from other platforms different than a laptop like a PDA or other mobile devices.
- A.I: We can relate this field with the Artificial Intelligence, the machine could recognize images and knowing that it means more than several concepts.

Besides, the proposed application is being distributed under GPL open source license. It is documented with the correspondent diagrams during the engineer process, so it can be reused or modified by the other developers.

References

1. BOOCH, G. JACOBSON, I. RUMBAUGH, J. UML *2 and the Unified Process: practical object-oriented analysis and design*. Second edition. USA: Addison Wesley, 2005. P.569 ISBN: 0-321-32127-8
2. ECKEL, B. *Piensa en Java*. (Spanish Edition of "Thinking Java"). Second Edition 2003, Publisher: Prentice Hall.
3. HAN J; KAMBER M. *Data Mining, Concepts and Techniques*. Publisher: Morgan Kauffman; Second edition (2006). ISBN: 978-1-55860-901-3.
4. ISO/IEC 15938-3/FDIS Information technology – *Multimedia content description interface – Part 3 Visual.* July 2001 (Sydney). ISO/IEC JTC 1/SC 29/WG 11 (Date: 2001-07-20). SO/IEC FDIS 15938-3:2001(E)
5. JAVA - *Introduction.* [Cited February–2008] Available in WWW: http://en.wikipedia.org/wiki/Java
6. JAVA – *Advantages.* [Cited February–2008] Available in WWW: http://arizonacommunity.com/articles/java_32001.shtml
7. KOENEN, R. *MPEG-4 Overview*. ISO/IEC – International Organization for Standardization–ISO/IEC JTC1/SC29/WG11 –N4668 – Coding of Moving Pictures and Audio. Available in WWW: http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm. [Cited December - 2007].
8. KOSCH, H. (2004). *Distributed Multimedia Database Technologies – Supported MPEG-7 by MPEG-21*. Publisher: CRC; 1 edition (November 24, 2003). ISBN: 0849318548
9. MARTINEZ, M. J. *MPEG-7 Overview*. ISO/IEC – International Organization for Standardization – ISO/IEC JTC1/SC29/WG11 – N5525 – Coding of Moving Pictures and Audio. Available in WWW: http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm. [Cited December - 2007].
10. MANJUNATH B.S; SALEMBIER P; SIKORA T. *Introduction to MPEG-7: Multimedia Content Description Language*. Publisher: John Wiley & Sons. (2002). ISBN: 0471486787.
11. ORACLE 9i Database, 2002, Available in WWW: http://www.oracle.com/. [Cited December - 2007].
12. ORACLE Corporation. *Oracle interMedia User's Guide, 10g Release 2* (10.2) [on line]. Sue Pelski. [Redwood City, USA]. June 2005. [Cited February - 2008]. Available in WWW: http://download-uk.oracle.com/docs/cd/B19306_01/appdev.102/b14302.pdf

13. Radial Basis Function. [Cited February - 2008]. Available in WWW: http://en.wikipedia.org/wiki/Radial_basis_function
14. RAY, E.T. *Learning XML*. Edited by O'Reilly.1st edition January 2001. ISBN: 0-59600-046-4.
15. THE MPEG INDUSTRY FORUM. [Cited December - 2007]. Available in WWW: http://www.m4if.org/resources.php
16. WILSON, B. *Induction of Decision Tree*. Available in WWW: http://www.cse.unsw.edu.au/~billw/cs9414/notes/ml/06prop/id3/id3.html
17. ZEZULA P; AMATO G; DOHNAL V; BATKO M. *Similarity Search, The metric Space Approach*, Publisher: Springer. ISBN 13: 978-0387-29146-8. 2006.

Chapter 9. Appendixes & Attachments

9.1 User Manual

When we init the program first time, we can see a intuitive and easy interface, with 4 main parts, the first one is the menu, on the top of the application, there you will find almost the same that in the buttons panel. In the menu you will have access to the different experiments.

The second part is the button panel, where you will find the main functions of the applications, that function we will use frequently. The Image panel is in the middle of the application and is the biggest part of the IU, there when the user load the image, this one will appear there, if you can't see nothing in this part, that's mean you don't have any image load and the user can't do operations like put an image in the data base...

The last part is a text area, where you can see the historical, and the warning of the application. This is really useful to see all the application in one view.

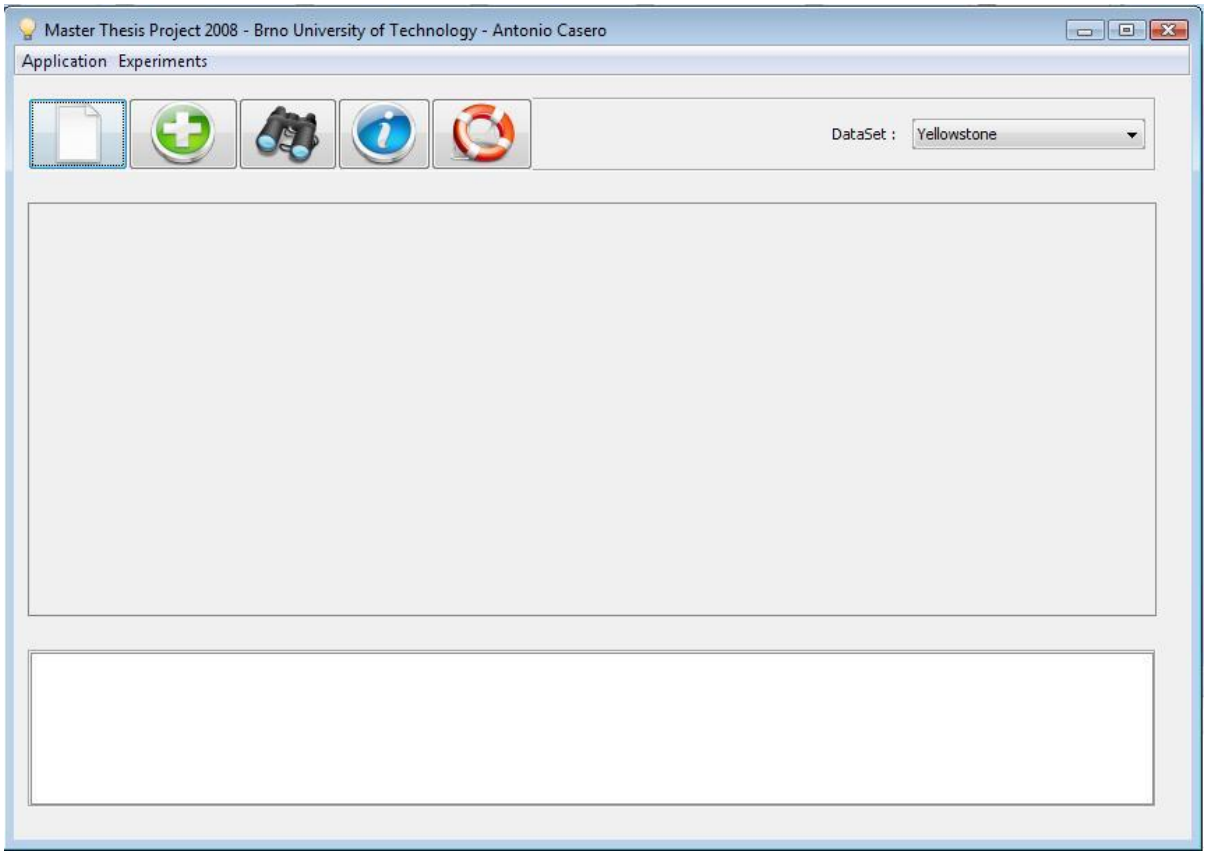


Figure 9.1 – The main frame

9.2 Select Images

At the beginning, you don't have any image in the data base, so you can't work with this application, you need introduce information on the data base, and create the MPEG-7 documents for each image.

To load a image on the application you need select witch one first, to do this press "Picture" button and will appear a window to choose a file, this window have a filter and you only can select files jpg, png, gif.... mean only picture files.

You can see how the IU (Figure 9.2) is, and you can use the preview to know what image you will load.

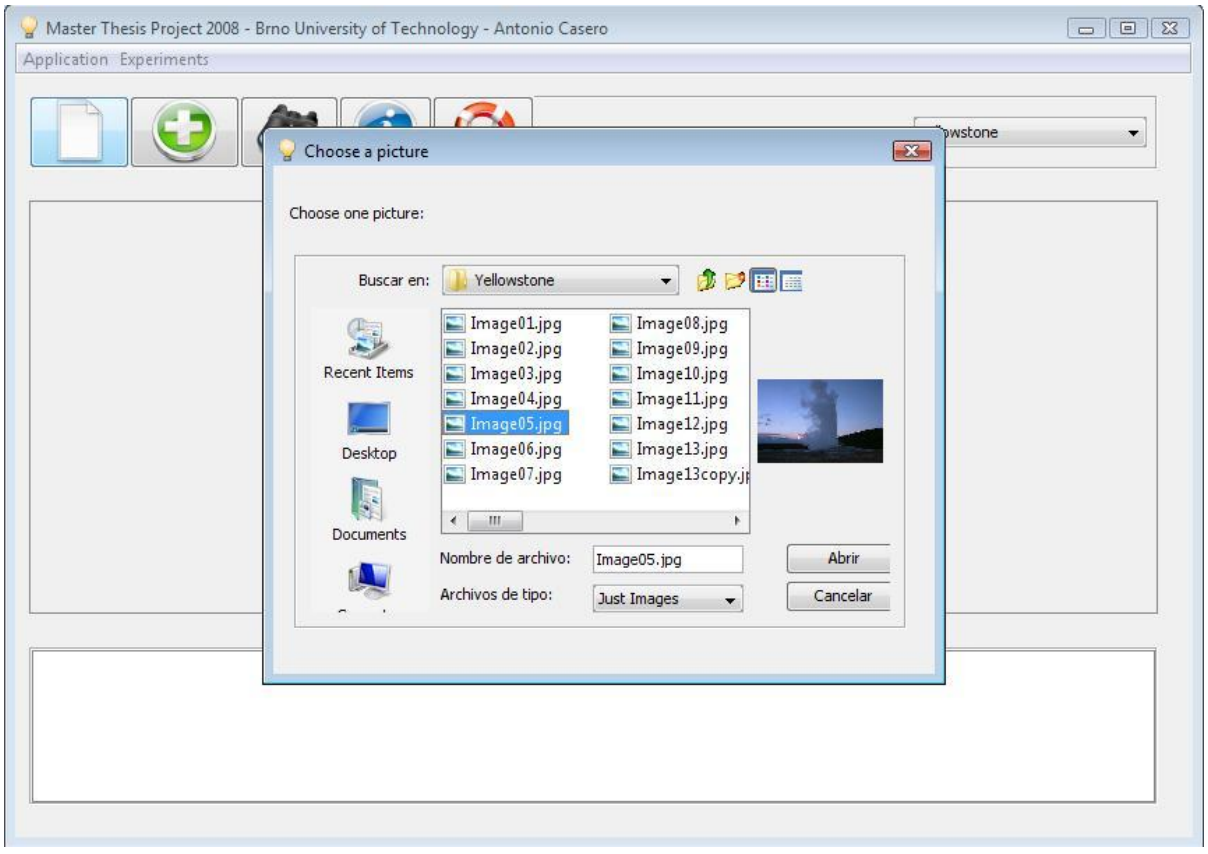


Figure 9.2 Choosing a Image

Remember, you need load the picture first, if the load is successful you will see on the text area of the main frame this sentence. If not you will see on the text area some kind of error.

9.2.1 Analysis of Images

If you want to do an analysis and storage of the image, you need to press “MPEG-7“ button from the button panel, you will see another window with all possibilities to analyzer the image with MPEG7 tools Figure (8.3).

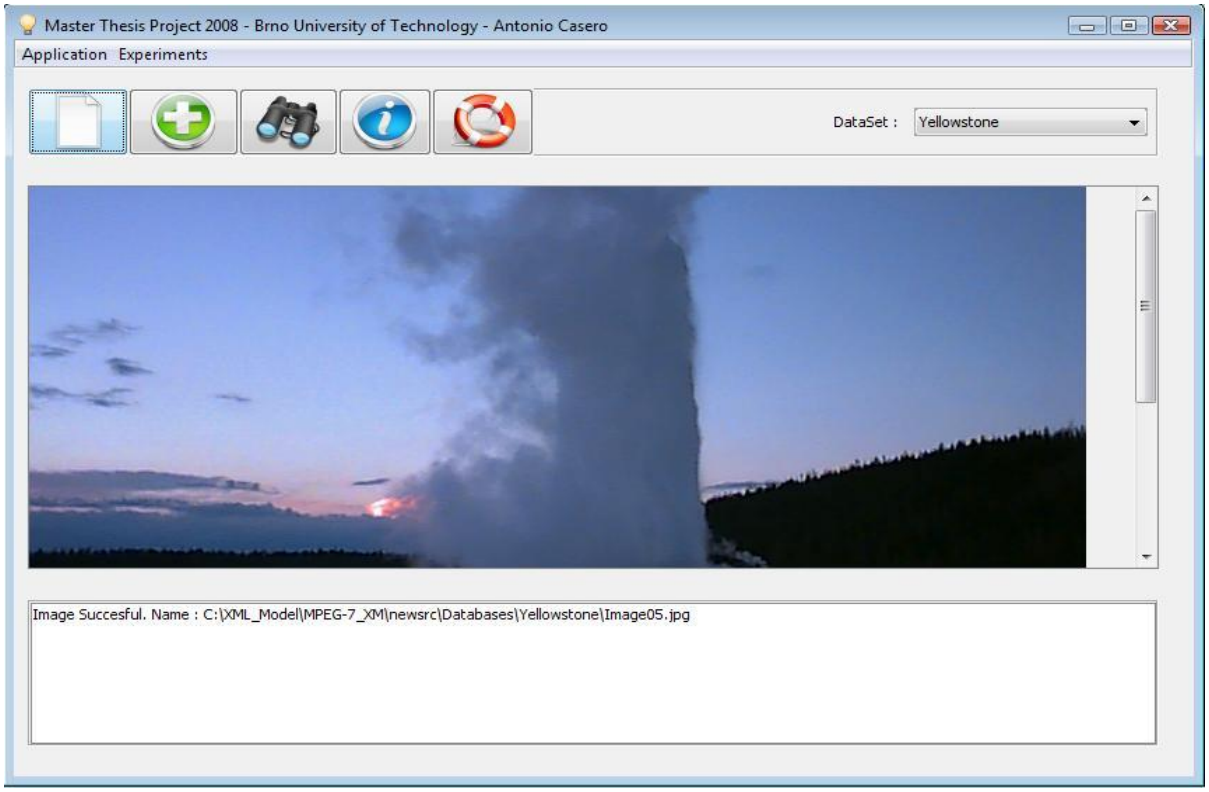


Figure 9.3 - Load Successful

Normally you will choose all the options to have a complete report of the image, but sometimes maybe you only want the Texture description, so it’s up to the user choose one.

Figure 9.4 – MPEG-7 Window

After pressing “OK“ in the MPEG7 window the application will call to the eXperimentation Model, it will write the XML document and the application will put in the data base. After this you will be able to see on the text area if the process was successful or not.

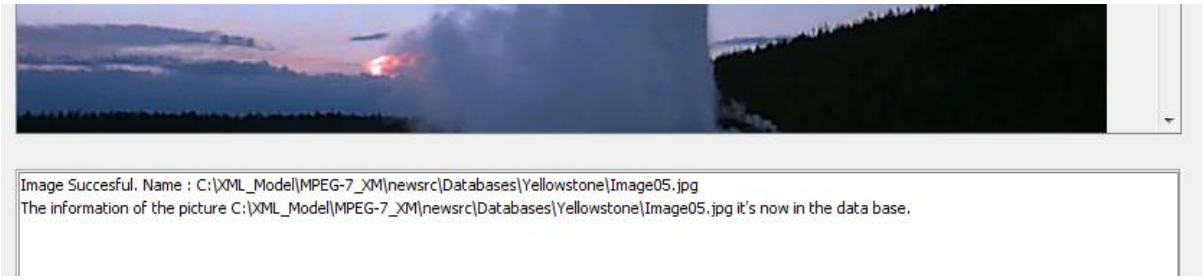


Figure 9.4 – Operation successful

Now we have the picture storage on the data base, as well we have the XML documents and all the coefficients of the different descriptors are also in the data base.

We can repeat this operation as many times as we want. This is better if we put a lot of images because later in the experiments will work better.

9.2.2 Search on the Database

You can look for information about an image in the data base, you only have to press the “Search“button and you will see a window like the figure 8.5. Remember you can delete images from the database, so, you will delete all the information of this Image.

This new window has a table with all the information of the image, name, MPEG7 documents, and dataset. The user can`t edit the table because maybe could damage the database.

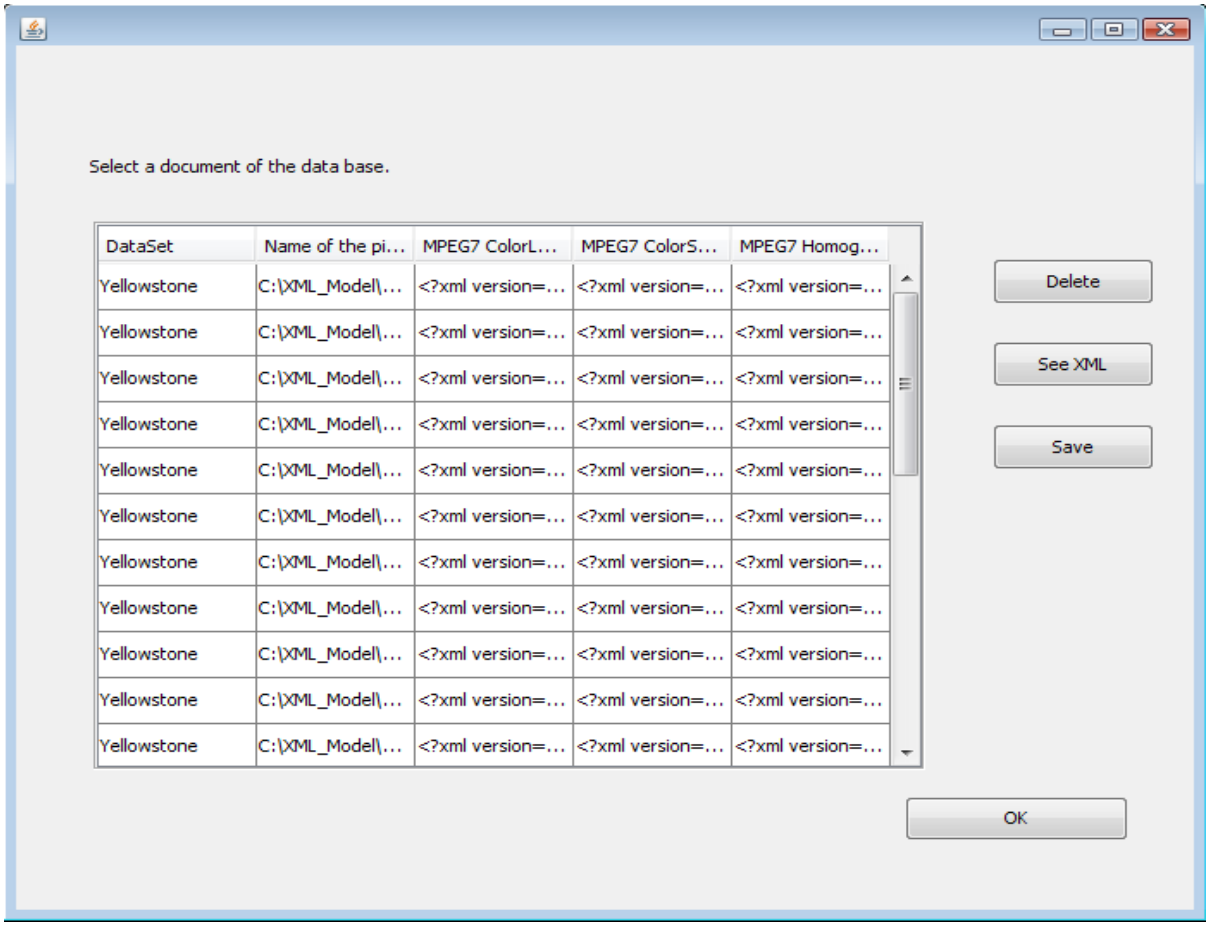


Figure 9.5 – The images in the database

You can select one picture in the table and see the mpeg-7 document of this file, you can delete pictures and if the user want can see the preview of the image with double click on the picture selected (Figure 8.6).

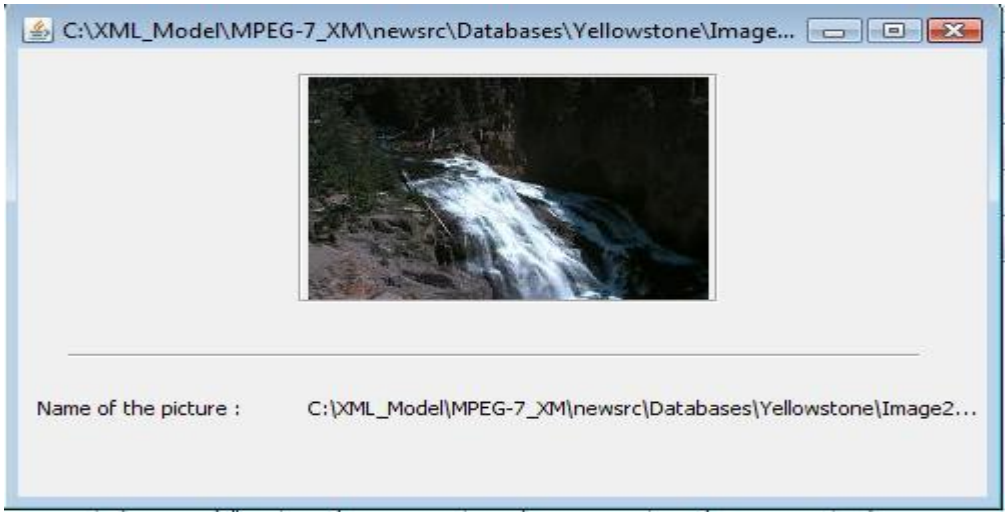


Figure 9.6 – Preview of the Image

9.2.3 The experiments

We are going to explain the main part of the application and it is a sample of how it's working all theoretic concepts explained before. I did two different experiments (Figure 6.5.1) where you can play with the options of the application.

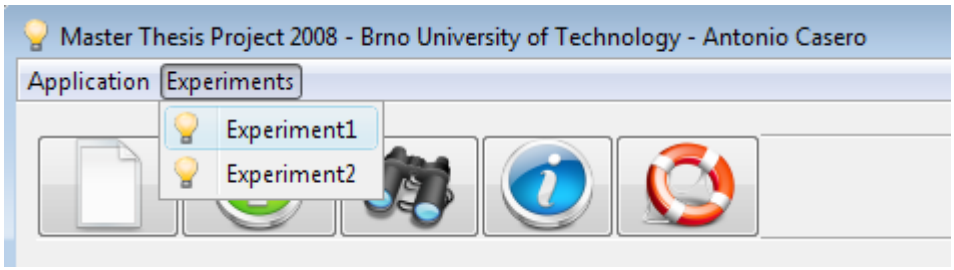


Figure 9.7- Experiments menu.

9.2.3.1 Experiment One

This experiment is about the similarity between photos through two powerful tools of treatment of images. We can see two tables with different results, the first one obtained through Oracle InterMedia are the first table. The second best is through use eXperimentation Model.

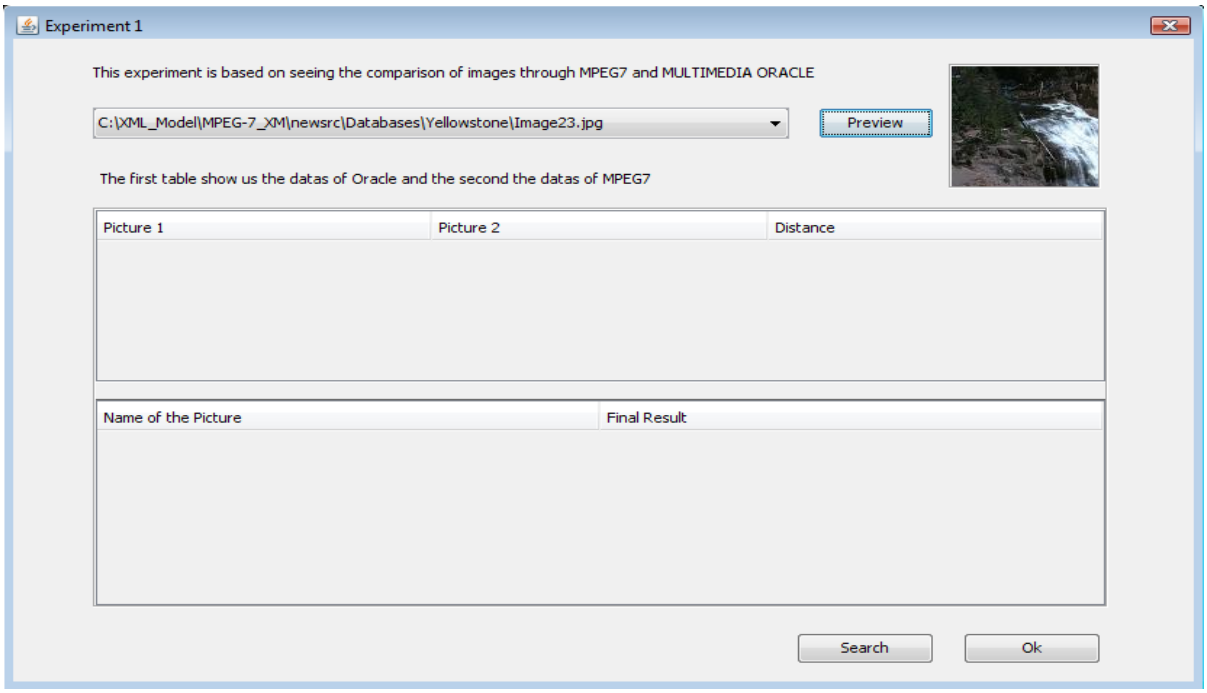


Figure 9.8 – Experiment 1 Interface

The IU of this experiment has a "Preview" button to search pictures in the data base, a simple and visual way the user can elect the image adapted for the experiment (Figure 6.5.1.2).

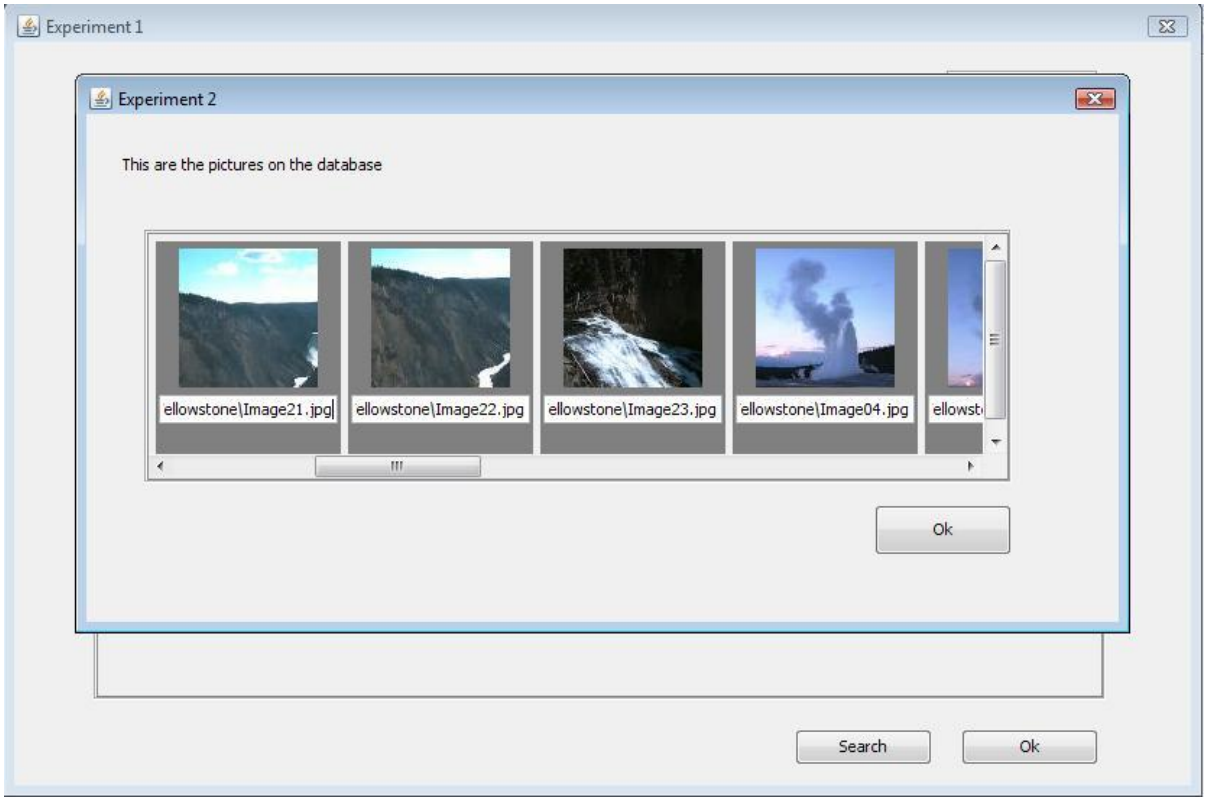


Figure 9.9 - Image Preview

After you have selected the image, you can see the thumbnail in the IU and you can press "search" button, after pressing the button, the algorithm will start and will put the results on the tables (All the theoretic part is describe at the beginning of this project).

In the tables you can see the name of the picture and the next cell is the distance with the other image, after used the algorithm, if the number is low that's mean that the picture selected and this picture are more similar than the rest of the pictures. Both tables using different methods so the values have not necessary the same values. You can see how works in the next example.

9.2.3.2 Experiment two.

This second experiment is about classification of the images in the data base, this will be useful to make annotation text in the images by the computer, that means if you have a big and good model of classification you can predict witch things are in the picture so you could ask to the program "which pictures are raining" or "sunny"…

This is the interface of this experiment Figure 9.10:

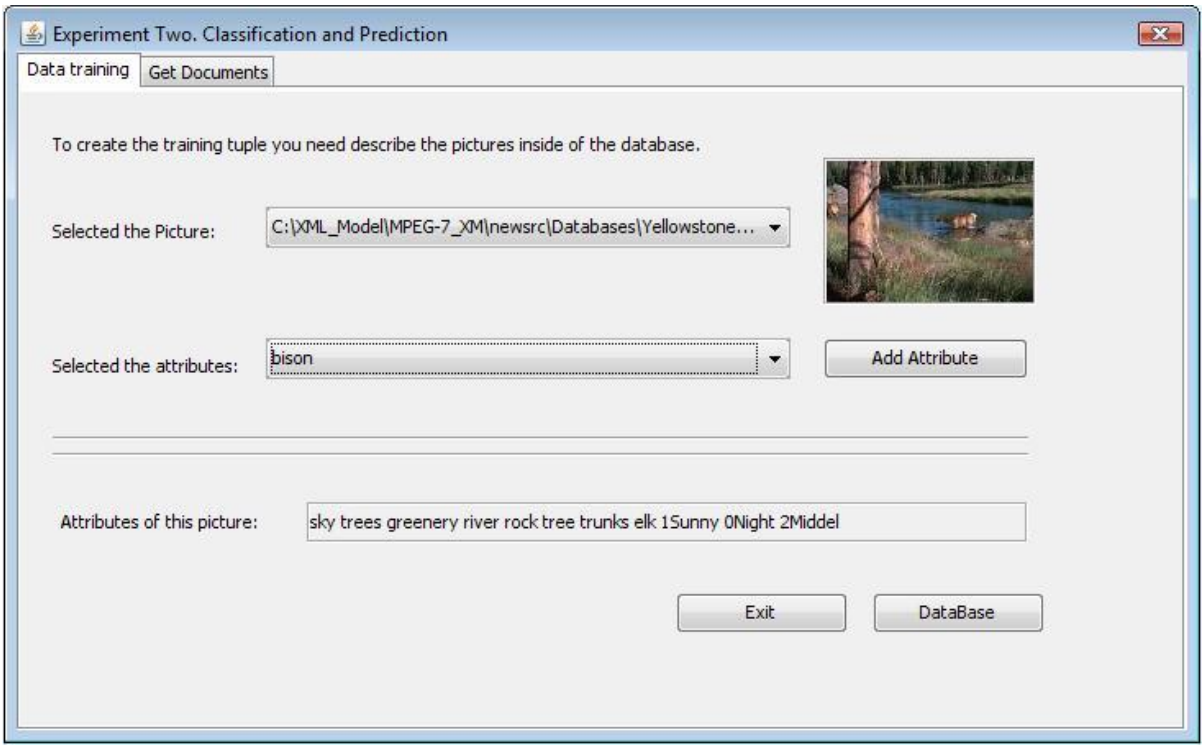


Figure 9.10 – The first tab of the experiment two

This experiment will create a document that we will use like our model of classification, the data training. In the first tab you can choose the picture and see which the images attributes are. And if you want, can you put more attributes to the selected picture in the data base.

On second tab, we could create a model with the attribute selected, and the application will labeling all the images on one file, that we will use later, on the RapidMiner program.

You can see the second interface and witch button you need to press to create the .cvs document.

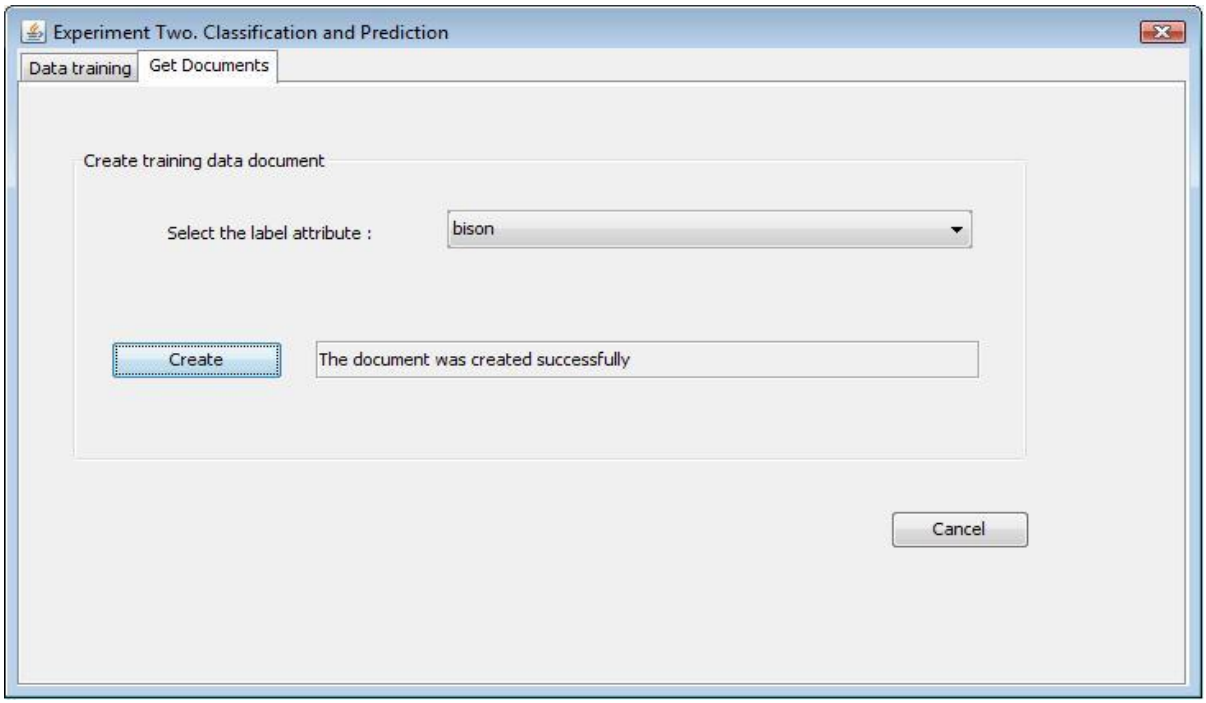


Figure 9.11 – Second tab of experiment two.

9.3 RapidMiner Manual

This project use RapidMiner to create a classifier model, so the application create a document with all data of images in database, this document has to be used in Wizard mode, and select the operator crossvalidation.

To know how its work, follow the next manual. All the screenshots are example from my application. See www.rapid-i.com for more information concerning this program.

9.3.1 General Information

Although the RapidMiner tutorial and this GUI manual contain a huge amount of information about RapidMiner and all of its parts, it is often very convenient to get the desired information during work. Therefore, we added an online help function to almost all parts of RapidMiner. Each parameter, operator and GUI item displays information as tool tip text, which appears after holding the mouse cursor a few moments above the object at hand.

9.3.2 Setting up a Process

9.3.2.1 The Wizard

We assume that you have chosen to start the Wizard (Figure 7.7). The Wizard is also available from the File menu. The Wizard guides you during the process of creating a new process. You start by selecting a template process from a list. This template serves as a kind of skeleton for your process.

Processes in RapidMiner are made up from a set of nested operators. An operator consumes a set of input objects and produces some output objects. These objects can be data files, models, performance criteria, and more. Simple operators like learners consume an example set and produce a model that can be used by an applier for prediction.

Moreover, some operators can have inner operators. For example, k-fold cross-validation splits up an example set into training set and test set and applies its inner operators, which are a learner and an applier. Each time a disjoint test set is used.

If you click on the radio button next to the ”SVM with XValidation” template, the structure of the sample process is depicted on the right. You see an operator chain consisting of an ExampleSource that reads data from a file. This data is then passed to the cross-validation, which itself has inner operators, in this case learner and applier for a support vector machine (SVM). See the RapidMiner Tutorial for more information on SVMs and the individual operators.

Now that you have chosen the template, click on next. In this step you can enter some of the most important parameters. In case of a cross-validation this is e. g. the number of validations.

9.3.2.2 Operator Configuration Wizards

An even more convenient way of loading almost arbitrary data files into RapidMiner and to define attribute description files for your data, is to use the Example Source configuration wizard. Just press the Start Configuration Wizard button at the top of the parameter table of the operator. Configuration wizards are also available for other operators which are hard to define, e.g. for the DatabaseExampleSource operator.

9.3.2.3 The tree view and other process views

As operators can have inner operators and each operator except the root operator is enclosed within another operator, the natural representation of a process is a tree. If you have used the Wizard, you see your process definition on the left side. If you did not use the Wizard you see an empty process consisting only of an empty operator chain. Figure 3 shows this main process view which is called “tree view”.

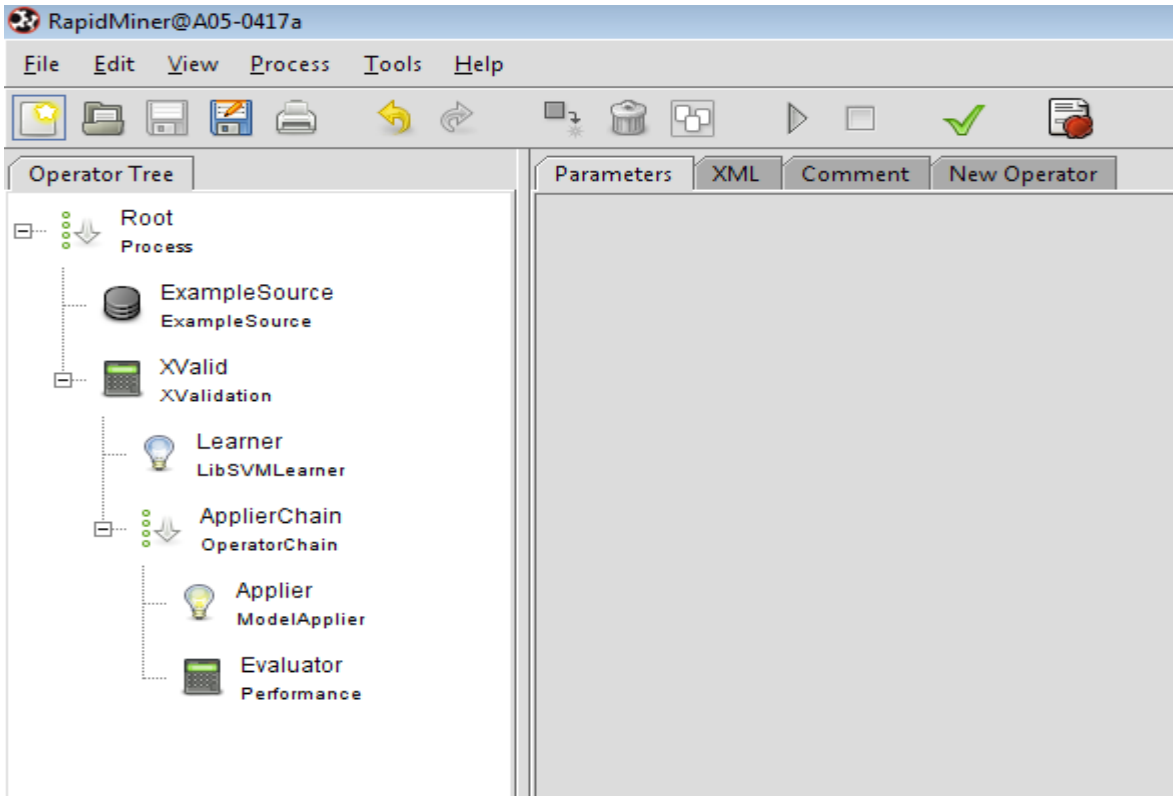


Figure 9.12 – Tree View

By clicking on the XML tab you see the XML configuration file that describes your process. If you like you can always edit it by hand using your favorite text editor. For more information about the XML configuration files see the RapidMiner Tutorial. You can specify HTML comments to each operator in the Comment tab which are saved in the XML files. If you specify a comment for the root operator of the process, this comment is displayed in a dialog each time the process setup was loaded. Selecting the Box View from the View menu shows a nicer box representation of your process you already know from the Wizard. You can use this view for printing.

9.3.2.4 Editing parameters

To the right of the tree you see a table with two columns labeled “Key” and “Value”. Depending on the selected operator you can enter the parameters of this operator.

Mandatory parameters are shown in bold face. Some of the parameters may have a default value, which will be used if no other value was specified. If the entered value is out of range, it will be corrected automatically. Some parameters accept only numbers, others let you select from a list of values. For file parameters, the file name can be entered into the text field or the file is selected by means of a file chooser dialog, that pops up when pressing the [...] button.

All file names can be defined relatively against the location of the process definition file. Of course this only works after the process was saved.

9.3.2.5 The attribute editor

Example sets or instance sets in RapidMiner are described by using a separate XML document. This attribute description file contains information about the type of data and its source. Data sets can be distributed over several files. This may be particularly useful if the label is stored within a file of its own. The RapidMiner Tutorial will give help in case you want to edit this file yourself.

The GUI displays a small Edit button next to an attribute description file property (e. g. the parameter attributes of an ExampleSource) in the property editor. A dialog called Attribute Editor will pop up containing a table with one column for each attribute (Figure 9.13). If the property does not yet reference a proper attribute description file, the dialog will be empty. If you want to follow the instructions below, which describe how to create the XML description file, you can clear the table by clicking on the corresponding button above the table, or selecting ”Clear” from the ”Table” menu, to start from scratch.

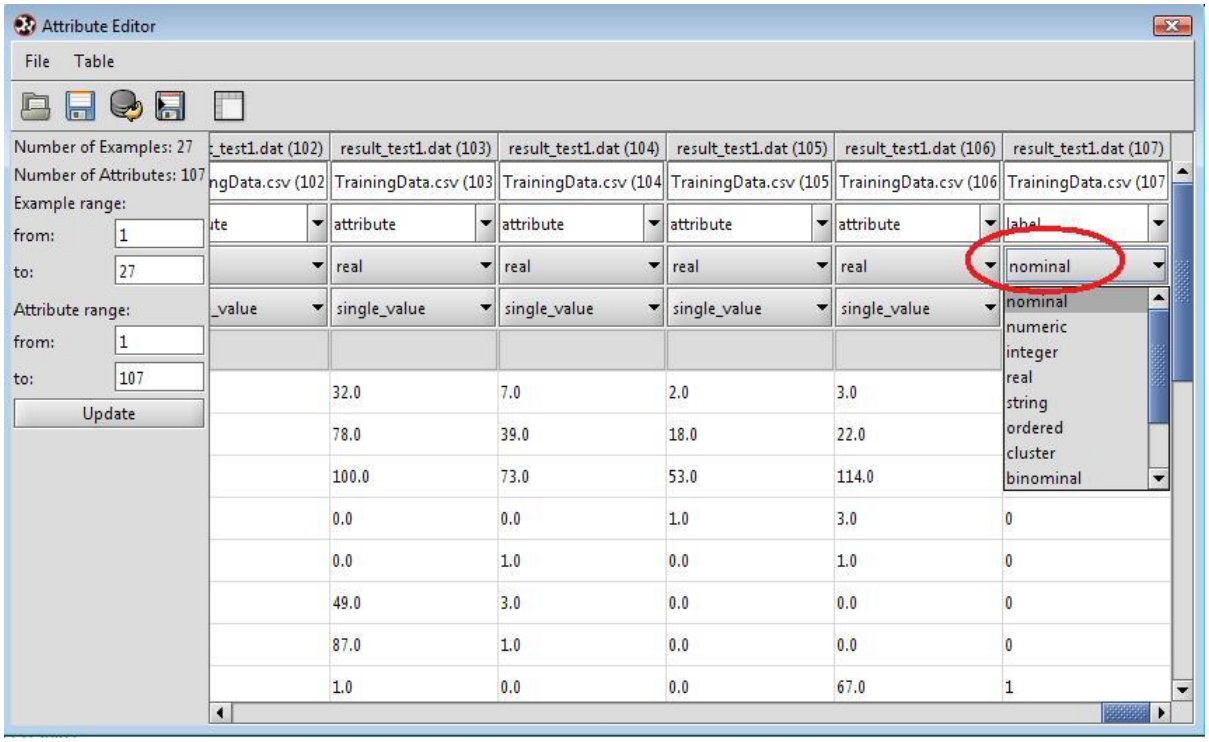


Figure 9.13 – Attribute Editor

The data file of my application is called "*TrainingData.cvs*", this is the file that we use in the wizard mode. When appear the attribute editor we have to change the last row to *nominal*, because the last coefficient is the **Label Attribute** (See chapter-3), remember that this attribute have to be assigned like *"Label"*.

If appear Question marks ("?") indicate missing values. The following enumeration explains the meanings of the table headers:

1. The first header row contains the source file and column index. This is not editable but just for your information.

2. The second rowshows the name of each column. You can edit the name by clicking into the text field.

3. The third row indicates, what the data is used for. For example, it can be an ordinary attribute, a label for classification or regression tasks, or a weight that can be used with certain algorithms. There can be at most one label and one weight attribute.

4. The fourth row is the value type. Most interesting are the choices real / integer and nominal. RapidMiner should have automatically detected these correctly.

5. The last header row is the block type. Most interesting are single value (default) and value series. For some processes, value series are treated in a special way. Do not forget to assign value series start and value series end to the first and last column respectively.

You can change the values according to your needs and load an arbitrary number of data files. Finally click on Save attribute description file, which you can find in the file menu, to write the XML file to disk, or just click on the ”Save” button.

9.3.3 Validating your process definition

Before you run your process you should validate it. You can click on Validate Process in order to check if all operators are nested correctly, provided with their necessary input and mandatory properties are set. Although this might be useful, you do not need to do it manually, since these checks are performed automatically before the process is started.

9.3.4 Running your process

Running your process is quite easy. Select Run from the Process menu or click the corresponding play button.

You may follow the progress of your process by observing the output which is displayed in the Message Viewer. Note, that in GUI mode, the output does not need to be written to a log file. If you did not specify a log file, you can always save the message viewer contents to a file by selecting the corresponding menu item in the Message Viewer’s context menu.

You can also perform a search in the Message Viewer. This option is also accessible from the context menu of the viewer.

9.3.5 Evaluating the results

When your process is finished, the results will be automatically presented, i. e. all output returned by the outermost operator. This can be performance statistics, a decision tree or anything else. RapidMiner automatically selects the Results Mode. You can switch between the Edit Mode and the Results Mode by clicking on the two buttons in the top right corner, or by selecting the mode in the View menu. Pressing the hotkey [F9] is another possibility to toggle between both modes.

When your process was conducted successfully, the view automatically switches to the Results Mode. As far as your process chain produces an output, this mode shows you visualization or a text describing the output. Figure 9.14 shows a decision tree learned from the “TrainingData.cvs”.

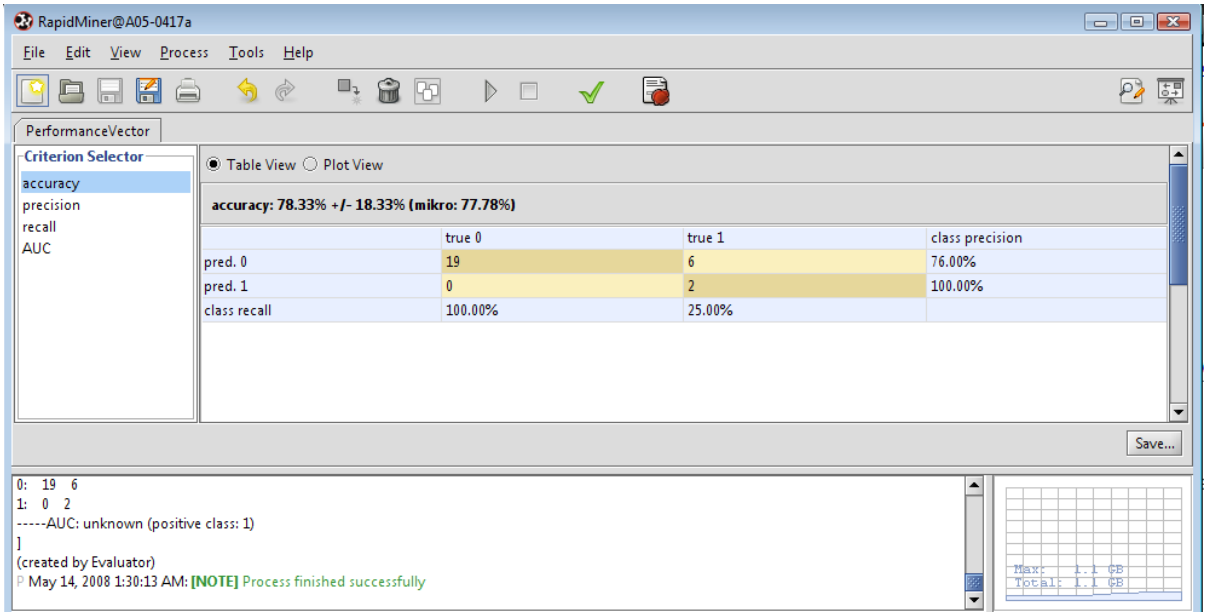


Figure 9.14 – Results

At any time you can stop or pause (and resume) the process using the appropriate buttons or menu items in the Process menu. In any case the operator currently being executed will finish its execution in the background. Since this might take some time (e.g. if the current operator was a learner) this might lead to a delay for the actual process termination. You can, however, directly start and perform changes to the current process setup and even restart the process.

9.4 CD Content

The CD include on this project has all necessary to run this application:

- Documentation – PDF and MS Word formats.
- Sources – Application codes and JDeveloper complete project.
- eXperimentation Model Sources.
- Rapid Miner V4 Beta.
- Includes – Necessary prerequisites.