Technical University of Crete (TUC)
Department of Electronics and
Computer Engineering

# DOGi: An Automatic Image Annotation Tool for Images of Dog Breeds

Adonis Dimas

Dissertation thesis

Committee

Euripides Petrakis,Associate Professor (Supervisor)

Michail Lagoudakis,Associate Professor

Michail Zervakis,Professor

Chania 2011

Image annotation is the process by which a computer system or a human assigns metadata (e.g., a caption) to a digital image.It is used in image retrieval systems to organize and locate images of interest from a database.The use of ontologies in image annotation is meant to make the assigned metadata conseptually motivated i.e that can be used to express the intended meaning of things, and not just words as textual strings.In this work , we present a complete framework that is capable of manual and automatic image annotation for a domain specified by an ontology.In order to make succesfull recognition of the image's class,we use the LIRE library and machine learning with decision trees.In the next paragraph ,we present the key points of our work.

First,an ontology is created that contains all the information that describe the domain of interest(for example in our domain specific example, we create a dog's ontology) .Then ,using LIRE library, we extract a set of low level characteristics and assign them to each respective class of the ontology.Image annotation is implemented as a retrieval process by comparing vectors of low-level descriptors extracted from the input (unkown) image and representative images of each class in the ontology respectively.Two similarity measures are used to compute the similarity between the unknown image and representative images of each class in the ontology :LIRE similarity measure which is a sum of the LIRE descriptors and DOGi similarity measure which is a weighted similarity measure of LIRE features measure with weights determined by decision trees.The result image list is ranked by decreasing similarity to the uknown image.Several ranking methods are used,inlcuding AVR(Average Retrieval Rank) , to estimate the semantic category where the image belong to (i.e.,the unknown image is assigned a class which is computed by voting among the top ranked retrieved images from the ontology).Finally,Annotation in MPEG-7 format ,for the class selected, is generated and stored in the exif metadata tags of the input image.The experimental results demonstrate that approximately 75-80% (depending on similarity and ranking method ) of the input images are correctly annotated . Experiments and evaluations were performed with the developed application DOGi implementing the above ideas..DOGi works with an image dataset consisting of 360 images of dogs belonging to 40 dog breeds

# Contents

# Chapter 1

# Intoduction

## 1.1 Motivation

Nowadays,content based image retieval (CBIR) is gaining ground over text-based or field-based image retrieval."Content-based" means that the search will analyze the actual contents of the image rather than the metadata such as keywords, tags, and/or descriptions associated with the image. The term 'content' in this context might refer to colors, shapes, textures, or any other information that can be derived from the image itself.In Figure 1.1 ,you can see a typical CBIR system.

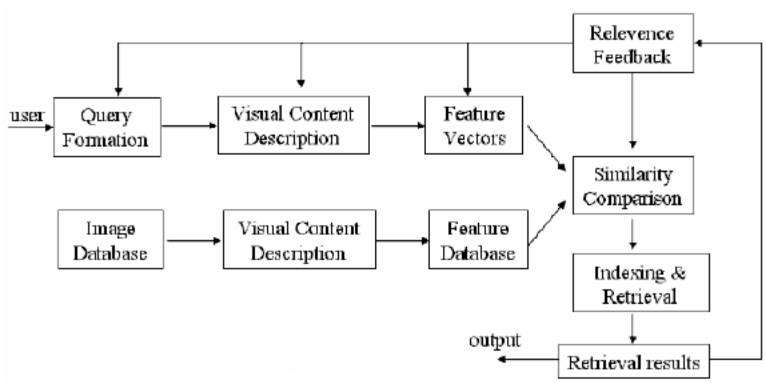



Figure 1.1: Diagram for content-based image retrieval (CBIR) system

CBIR is desirable because most web based image search engines rely purely on metadata and this produces a lot of not related images in the results. Also having humans manually enter keywords for images in a large database can be inefficient, expensive and may not capture every keyword that describes the image. Thus a system that can filter images based on their content would provide better indexing and return more accurate results.[1]

However, similarity between image features (extracted by image analysis) does not always correspond to semantic similarity as perceived by humans through text descriptions.This is referred to as the semantic gap problem [2]. The semantic gap is the lack of coincidence between the information that one can extract from the visual data and the interpreting that the same data have for a user in a given situation.A solution to this problem would be the use of ontologies to enrich low-level features with semantic meanings .This approch is relatively new, so it is still inaccurate how to associate semantic concepts with visual features effectively and efficiently.

## 1.2 Image Annotation

A typical problem in image information systems often occurs when an end-user is given with one or more images and is asked to assign a description to each one.This problem is refered as image annotation or image tagging[3].

Image annotation is a process of assigning metadata descriptions in the image and it is used in image retrieval systems to organize and locate images of interest from a database.There are two types of image annotation, manual and automatic image annotation.In manual annotation , a human assigns text descriptions to each image according to his-her perception.In automatic image annotation a computer system generates annotations for each image based on a content based image retrieval model (CBIR).Manual annotation can provide rich image descriptions, however it is time consuming and thus expensive. On the other hand, automatic annotation based on automatic feature extraction is relatively fast and cheap but is not always accurate compared to manual annotation.[4]

In order to improve image annotation , we must analyze the semantics that describe visual data in an image.In general, the semantics consist of two parts that describe different aspects of visual data: one part contains the feature descriptions for the image itself (content semantics), and the other comprises of content descriptions from the human conceptual aspect (concept semantics).In other words,content semantics describe the low-level characteristics of the image (color,texture,shape...)while concept semantics refer to the high-level charecteristics (who or what is pictured in the image, where it is or when etc.).It is possible to encapsulate these semantics using an ontology of a specific domain.

In this work,we implement automatic image annotation based on automatic feature extraction to recognize the class of a domain that belongs to an input image and we use an ontology of that domain to make the annotation structrured and meaningfull.Image annotation is viewed as an image classification problem: The system is provided with an unknown image and the problem relates to assigning a class name to it, that is mapping the unknown image to one of a number of known classes. The image then inherits the class properties and annotation of its assigned class.

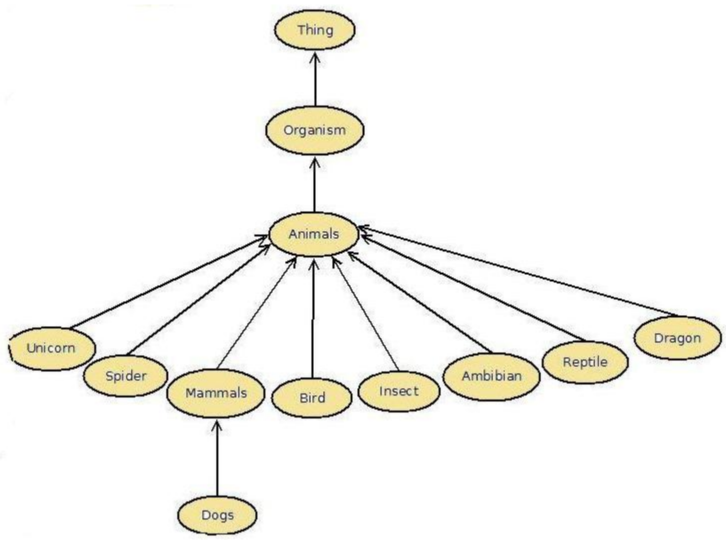



Figure 1.2: A simple ontology of domain : Animals.You can see the upwards isa relationships.For example ,Dogs class is connected with Mammals class with an isa relation...etc.Note that in this example ontology ,only class hierarhies are provided.

## 1.3 Ontologies

In computer and information science, an ontology is a formal representation of knowledge as a set of concepts within a domain, and the relationships between those concepts[5].A popular definition for ontology states:An ontology is a formal ,expicit specification of a shared conceptualization[6].In Figure 1.2 you can see an ontology (the basic hierarhy only) of the domain: Animals.

Ontologies are used in computer science to facilitate sharing and reuse.They are very usefull because they give a shared and common understanding of a domain that can be communicated between people and application systems.The basic and most significant role of ontologies in knowledge and software engineering is to facilitate the construction of a domain model.

In this work, we constructed an ontology with the domain :"dog breeds".This ontology consists of 40 semantic classes of dog breeds (border collie,bulldog,german shepherd...etc.) and each class has it's own property values that are assigned to (for example , a border collie class has country of origin scotland, has main color black and white while a german shepherd class has country of origin germany and has main color black...etc.).Our Ontology of dogs is discussed in detail in chapter 3.

## 1.4 LIRE Library

Our CBIR system works using LIRE library[7].The LIRE (Lucene Image REtrieval) library provides a simple way to retrieve images and photos based on their color and texture characteristics. LIRE creates a Lucene index of image features for content based image retrieval.The visual descriptors that our framework extract and analyze from a raw image are the following:

***Color Descriptors:***

**1.Color Layout Descriptor (CLD)**

**2.Color Structure Descriptor (CSD)**

**3.Scalable Color Descriptor (SCD)**

**4.Auto-Corellogram Descriptor (ACD)**

**5.Fuzzy-Color Histogram Descriptor (FCHD)**

**6.Simple-Color Histogram Descriptor (SCHD)**

***Texture Descriptors:***

**7.Gabor Descriptor (GD)**

**8.Tamura Descriptor (TD)**

**9.Edge Histogram Descriptor (EHD)**

***Hybrid Descriptors:***

**10.Color-Edge Directivity Descriptor (CEDD)**

**11.Fuzzy-Color Texture Histogram Descriptor (FCTHD)**

**12.Joint CEDD-FCTH Descriptor(JD)**

*Note that FCHD and CSD are not part of LIRE library.
All visual descriptors are discussed in detail in Chapter 2

## 1.5 Mpeg -7 Annotation and Exif

An annotation can be expressed in many different formats such as: owl, rdf, xml, mpeg7 to name a few. Standardized Web technologies and XML based description languages are required in order to achieve interoperapility with other applications. MPEG- 7 plays an important role towards the standardized enrichment of multimedia with semantics on higher abstraction levels and a related improvement of query results.It is used as the standard metadata format for exchanging automatic analysis results[8].

Exchangeable image file format (Exif) is a specification for the image file format used by digital cameras (including smartphones) and scanners. The specification uses the existing JPEG, TIFF Rev. 6.0, and RIFF WAV file formats, with the addition of specific metadata tags.

In this work,we generate annotations in MPEG-7 format ,by mapping owl classes and properties(owl is the The Web Ontology Language in which an ontology is described) to MPEG7.Then, the generated MPEG7 annotation is saved in the “annotations” metadata Exif tag of the input(unknown) image.

## 1.6 Related Work

In recent years,a number of notable work dealing with image annotation and the semantic gap has been presented.

Researches are trying to arrange low-level features to semantic meaningful categories (keywords) [9].Besides associating features to keywords, another important source of information is the relationship between semantic labels, often referred to as semantic ontology. Usually these ontologies provide a multi-layer tree structure hierarchy description of contents. This enables machines to identify the low-level feature descriptions for human conceptual items through the keywords given by users [10]

Other studies proposed a cross-media relevance model in which automatic image annotation and retrieval is conducted using blobs of image features that are extracted by clustering techniques.Thus,given a training set of images with annotations,the probability of generating a word given the blobs in an image is predicted [11].Also,recent efforts are trying to arrange visual features to semi-concepts values. Image annotation in this case is based on semantic inference rules[12]

Finally,there are researches more close to our approach that are based on the idea of image annotation using ontologies.Ontologies are used to maintain keywords along with high-level information for semantic text retrieval purposes. It is widely accepted that the retrieval of images annotated with keywords may provide potentially better results[13]. The quality of the retrieved results depends on the amount, quality, and consistency of the metadata associated with each image [14]. High-level concepts are efficiently stored and automatically mapped to visual features or objects which are extracted by various image analysis techniques [15, 16].

Our work is based on SIA [17], a semantic image annotation framework.We used SIA's main principles but we tried to focus on the ontology constuction and the visual descriptors.We have added five more low level descriptors in order to enhance the content description scheme.We have computed and used two similarity measures instead of one used in SIA.Also,we have constructed a more enriched ontology and a better training set( nearly twice training samples and 9 instances for each semantic category instead of 6 used in SIA).Finally,In SIA annotations where generated but not stored in actual metadata form.In our framework,we have managed to store annotations in exif metadata tags of the annotated image.Our experiments,in chapter 6, show an overall improvement of the automatic image annotation using our framework instead of SIA.Our framework managed 95% correct annotation in the first 3 answers while SIA had a nearly 90%.

We have tried to combine the available CBIR techniques described for semantic category estimation and ontologies for the image annotation.Thus,we have created a framework that bridges the semantic gap created by "content" based image retrieval.Our research show that this approach can be very efficient and usefull when a general content description scheme and descriptive enough ontology are constructed.

## 1.7 Methodology

We propose an intelligent framework for image annotation using ontologies, by combining the analysis of visual content and the manually contructed description of image data. High level descriptions and low-level information are efficiently stored in an ontology model providing formal descriptions. Low-level features have their implementation value enabling the ontology being an annotation or content-based image retrieval system.

The problem of image annotation is treated as a retrieval problem which is facilitated by two similarity measure schemes on descriptors making the retrieval possible, and a voting scheme for computing the semantic category of an unknown image from the categories of images in the retrieved set. Our system works in four steps:

**Building the Ontology:** First step is to build an ontology model for keeping all the necessary information about the images in the database. This includes both low-level features and concept descriptions. In this work, such concept descriptions are obtained from WordNet and Wikipedia and are provided in the form of short text descriptions.

**Image Similarity:** Two similarity measures are computed between images.LIRE similarity measure is the distances sum of all the descriptors used while Weigthed similarity measure is the LIRE similarity measure with relative importance of each low-level feature(color,texture) determined using machine learning by decision trees. Weights are ranged between [0-1].

**Image Retrieval:** Given an unknown image, the ontology is searched to retrieve the images most similar to it. Image matching is implemented using image content descriptions (color, texture and hybrid features). The similarity measures computed above are used for image similarity. Finally the result images are ranked in decreasing order of similarity.

**Image Annotation:** The unknown image is classified into one of known semantic cat egories. The semantic image category with instances having best ranks in the retrieved set is chosen. Finally its description is assigned to the unknown image.

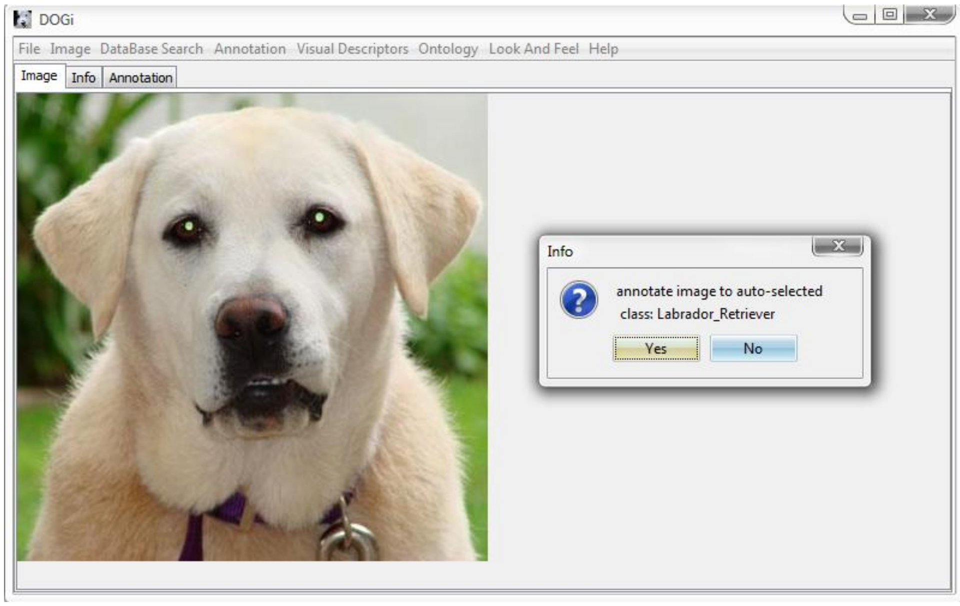



Figure 1.3 : Snapshot of DOGi in action.In this example,DOGi selected labrador retriever as the class belonging to the image in the panel.

## 1.8 DOGi : Dog OntoloGy Image Annotator

We developed the application: DOGi (Dog OntoloGy Image Annotator) capable of automatically (and manually) annotating images that contain dogs into 40 dog breeds following the principles and ideas of the CBIR system proposed in this thesis.DOGi is a succesfull example of how a CBIR system can produce structered and meaningfull descriptions(annotations) using ontologies.

DOGi is a specilization to the general problem of automatic image annotation and uses the ontology with domain:Dogs(discussed in chapter 3). It has a basic graphic user interface for loading and displaying images , loading and viewing ontologies to the panel.The user can select a ROI (Region Of Interest), in which the system performs image retrieval.Then,the user selects a similarity measure(between LIRE similarity measure and DOGi similarity measure both discussed in chapter 4) and an annotation method(including AVR).The visual descriptors of ROI are extracted and compared with the ones of the 360 database images of the ontology.The class of the annotation method is selected by the user and it is used to generate a MPEG7 annotation.Finally,the user has the option to save the annotation in a file seperately or to save it in the original image's exif metadata tag: annotations.

All presented experiments and implementations are conducted through DOGi.Figure 1.3 illustrates a snapshot of the DOGi interface .

## 1.9 Structure of this thesis

The rest of this thesis is organized as followed:

**Chapter 2:** ***Image Content Analysis***

The visual descriptors used in this work are discussed

**Chapter 3:** ***Ontology construction***

The construction of the domain using an Ontology is presented

**Chapter 4: Similarity Measures**

The similarity measures as well as the normalization techniques used in our system, are presented

**Chapter 5:** ***Annotation***

Mapping owl to mpeg7,Mpeg7 annotation and exif are discussed

**Chapter 6:** ***Experiments***

Experimental including the data set and issues related to the evaluation methodology that has been followed are discussed.

**Chapter 7:** ***Conclusion***

Summarizes the main achievements of this thesis, discusses results obtained, and provides suggestions for further work.

# Chapter 2

# Image Content Analysis

In this chapter ,we analyze the image content descriptors used by our framework to perform a succesfull image retrieval.

A descriptor is defined as a representation of a feature. A descriptor defines the syntax and semantics of the feature representation. Examples of low-level visual features include color, shape, motion, and texture. As noted previously,We use 3 types of descriptors.Color descriptors ,texture descriptors and hybrid descriptors (a combination of color and texture)

## 2.1 Color Descriptors

Color is the most basic quality of visual content. One of the most recognizable elements of image content and is the most commonly used feature in image retrieval because of its invariance with respect to image caling, translation and rotation. Color features are independent of image size and orientation and can be used for describing content in still images and video. A good reference on how color descriptors can be used in image retrieval is [18]

### 2.1.1 MPEG7 Color Descriptors:

MPEG-7, formally named "Multimedia Content Description Interface", is a standard for describing the multimedia content data that supports some degree of interpretation of the information meaning, which can be passed onto, or accessed by, a device or a computer code.Mpeg7 visual descriptors are often used for image retrieval[8, 19, 20].

Mpeg-7 supports 4 color descriptors (as shown in the Figure 2.1) : Dominant Color,Scalable color,colour strucure and color layout.In this work,we do not use dominant color because of implementation issues.

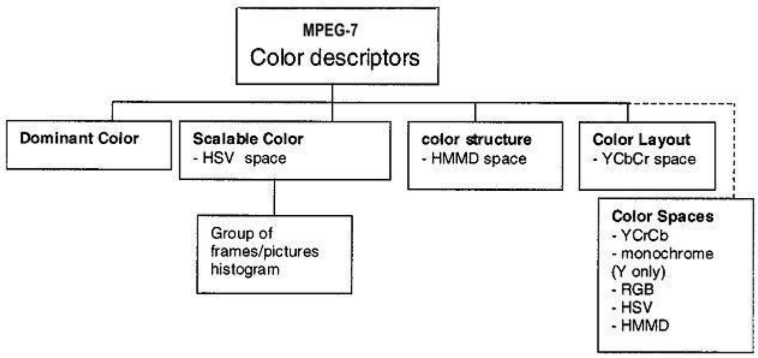



Figure 2.1 : MPEG-7 color descriptors

#### 2.1.1.1 Scalable Color Descriptor (SCD)

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.

The histogram values are extracted, normalized and non-linearly mapped into a 4-bit integer representation, giving higher significance to small values. The Haar transform is applied to the 4-bit integer values across the histogram bins. The basic unit of the transform consists of a sum operation and a difference operation (see Figure 2.2 (a)), which relate to primitive low pass and high pass filters.Summing pairs of adjacent bins is equivalent to the calculation of a histogram with half number of bins. From the sums of every two adjacent Hue bin values out of the 256-bin histogram,we get a representation of a 128-bin histogram with 8 levels in H,4 levels in S and 4 levels in V. If this process is repeated, the resulting 64, 32 or 16 sum coefficients from the Haar representation are equivalent to histograms with 64, 32 or 16 bins.

Table 2.1 shows the equivalent partitioning of the HSV color space for different number of coefficients of the Haar transform. If an application does not require the full resolution, limited number of Haar coefficients may simply be extracted from a 128, 64 or 32 bin histogram. This would still guarantee interoperability with another representation where all coefficients were extracted, but only to the precision of the coefficients that are available in both of the representations.

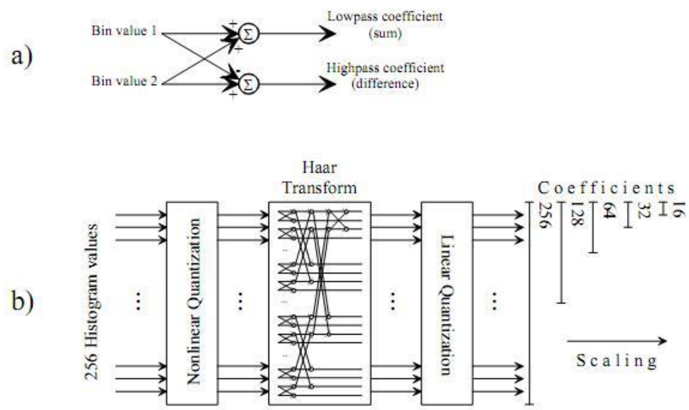



Figure 2.2 : (a) Basic unit of Haar transform (b)A schematic diagram of scalable color descriptor generation.

| # of coeff's | # of bins H | # of bins S | # of bins V |
|---|---|---|---|
| 16 | 4 | 2 | 2 |
| 32 | 8 | 2 | 2 |
| 64 | 8 | 2 | 4 |
| 128 | 8 | 4 | 4 |
| 256 | 16 | 4 | 4 |

Table 2.1 : Equivalent partitioning of HSV color space for different numbers of coefficients in the scalable color descriptor

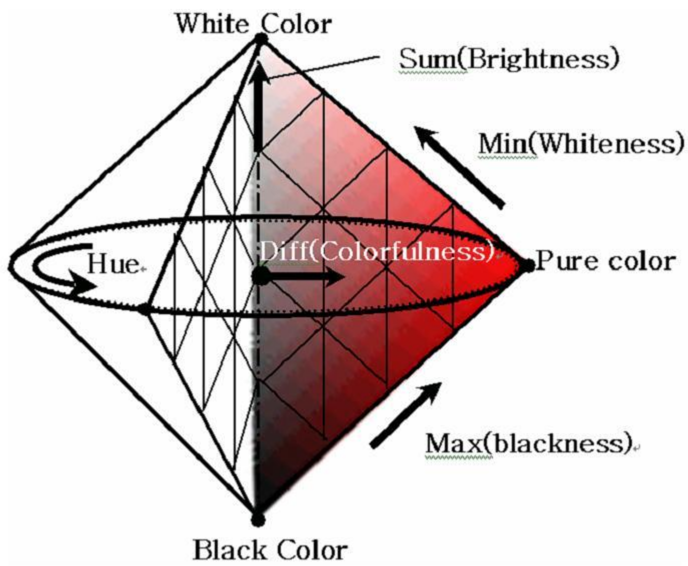



Figure 2.3 : Double cone representation of the HMMD color space

In order to use this descriptor to perform similarity retrieval,we use the matching function that is based on L1 metric.The matching function for Scalable Color is

$$D_{SC} = \sum_{i=1}^{N} |H_A[i] - H_B[i]$$

In principle, any other matching method suitable for histograms can be used, although it was found that L1 metric give very good retrieval performance in the MPEG-7 core experiments.

#### 2.1.1.2 Color Structure Dexcriptor (CSD)

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(position of color).

The CSD works on a special version of the HMMD color space (see Figure 2.3) dened by a non-uniform color space quantication. First, the HMMD color space is divided into five subspaces. This division is performed according to the Diff value where subspaces 0, 1, 2, 3, and 4 correspond respectively to Diff intervals [0, 6], [6, 20], [20, 60], [60,110], and [110, 255]. Then a pixel is quantized along the Hue and Sum axes according to its subspace (see Table 2.2).

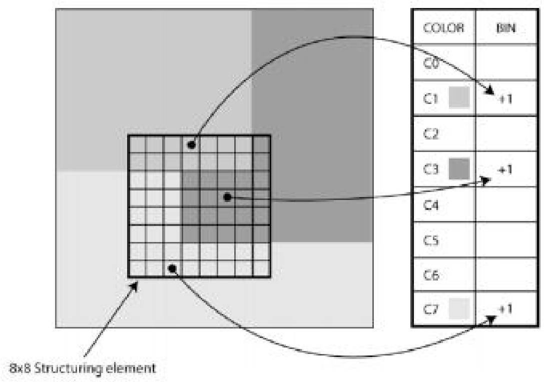



Figure 2.4 : 8x8 structing element

TABLE II
HMMD COLOR SPACE QUANTIZATION FOR CSD

| Component | Subspace | Number of quantisation levels for different numbers of histogram bins | | | |
|---|---|---|---|---|---|
| | | 184 | 120 | 64 | 32 |
| Hue | 0 | 1 | 1 | 1 | 1 |
| | 1 | 8 | 4 | 4 | 4 |
| | 2 | 12 | 12 | 6 | 3 |
| | 3 | 12 | 12 | 4 | 2 |
| | 4 | 24 | | | |
| Sum | 0 | 8 | 8 | 8 | 8 |
| | 1 | 4 | 4 | 4 | 2 |
| | 2 | 4 | 4 | 4 | 4 |
| | 3 | 4 | 4 | 4 | 2 |
| | 4 | 2 | | | |

Table 2.2 : HMMD color space quantization for CSD

Once the quantization step is done, the CSD is computed by visiting all locations in the image.At each location, the color $Cm$ (with $m \in [0, M-1]$) of all the pixels contained in the 8x8 structuring element overlaid are retrieved.The CSD bins are incremented according to these colors.

In other words, even if 14 of the 64 pixels are dened by color C1 in the structuring element of Figure 2.4,the bin C1 is only incremented of one.

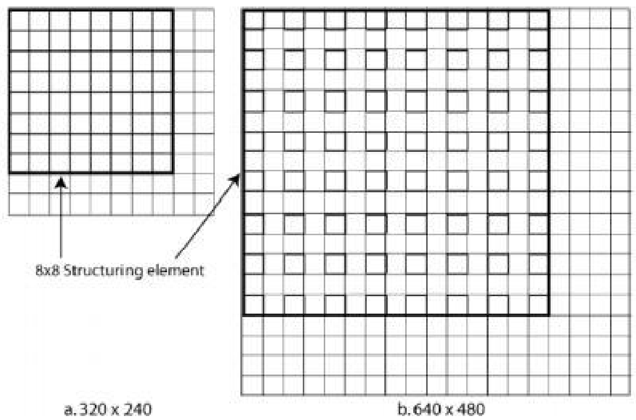



Figure 2.5 : 64 structing points for 2 structing element sizes

This structuring element is exploited to avoid the lost of structure with typical histograms. The number of structuring points is always 64 and the distance between them increases with the image size (Figure 2.5).

Once the CSD histogram is computed, a non-linear quantization step is performed to obtain a 8-bits coding for each bins. Since the structure of the descriptor is the same,the same matching functions can be used. The default matching function is the L1 metric, as in the case of Scalable Color.

#### 2.1.1.3 Color Layout Descriptor (CLD)

The CLD descriptor captures the spatial layout of the representative colors on a region or image.Representation is based on coefficients of the Discrete Cosine Transform. This is a very compact descriptor being highly efficient in fast browsing and search applications. It provides image-to-image matching as well as ultra high-speed sequence- to-sequence matching.

The Color Layout uses an array of representative colors for the image,expressed in the YCbCr color space,as the starting point for the descriptor denition.The size of the array is fixed to 8x8 elements to ensure scale invariance of the descriptor. The array obtained in this way is then transformed using the Discrete Cosine Transform (DCT), which is followed by zig-zag re-ordering(see Figure 2.6).

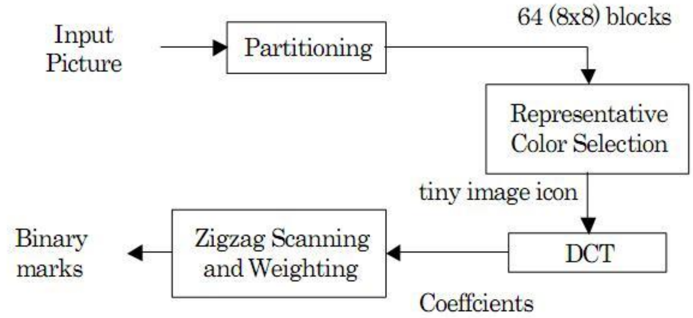



Figure 2.6 : The extraction process of the color layout descriptor

A representative color was chosen for each block by averaging the values of all the pixels in each block. This results in three 8x8 arrays, one for each color component.This step is directly visualized in the first window of Figure 2.7 .Each 8x8 matrix was transformed to the YCbCr color space (second window of Figure 2.7).Next each 8x8 matrix was transformed by 8x8 DCT to obtain 3 8x8 DCT matrices of coefficients, one for each YCbCr component (third window of Figure 2.7).The CLD descriptor was formed by reading in zigzag order 6 coefficients from the Y-DCT-matrix and 3 coefficients from each DCT matrix of the two chrominance components.The descriptor is saved as an array of 12 values.

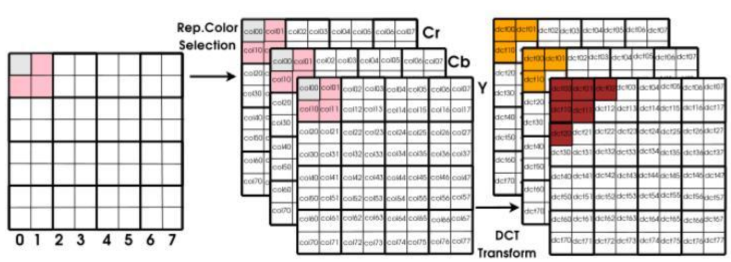



Figure 2.7 : Stages of CLD computation

The default matching function is essentially a weighted sum of squared dierences between the corresponding descriptor components.

$$D_{CL} = \sqrt{\sum_i w_i^y (Y_i - Y_i')^2} + \sqrt{\sum_i w_i^b (C_{b_i} - C_{b_i}')^2} + \sqrt{\sum_i w_i^r (C_{r_i} - C_{r_i}')^2}$$

where $Y$ , $Cb$ and $Cr$ are the DCT coefficients of the respective color components, $w_i^y$ , $w_i^r$ , $w_i^b$ are weights chosen to reflect the perceptual importance of the coefficients and the summation is over the number of coefficients.

CLD ,SCD,CSD are described in detail in the following papers [21, 22, 20]

### 2.1.2 Color (Histogram) Descriptors:

The color histogram Descriptors are statistics that can be viewed as an approximation of an underlying continuous distribution of colors values.They include simple color histogram,fuzzy color histogram and auto corellogram.

#### 2.1.2.1 Simple Color Histogram Descriptor (SCHD)

The simple color histogram descriptor (SCHD) is a histogram descriptor that indicates the frequency of occurrence of every color in an image.The appealing aspect of the SCHD is its simplicity and ease of computation[23].

A Color histogram refers to the probability mass function of the image intensities.This is extended for color images to capture the joint probabilities of the intensities of the three color channels. More formally, the color histogram is defined by,

$$h_{A,B,C}(a, b, c) = NProb(A = a, B = b, C = c)$$

where A , B and C represent the three color channels (R,G,B or H,S,V) and N is the number of pixels in the image.

Each pixel is associated to a specific histogram bin only on the basis of its own color, and color similarity across different bins color dissimilarity in the same bin are not taken in account. Since any pixel in the image can be described three components in a certain colour space(for instance, red, green and blue components in RGB space or hue, saturation and value in HSV space) histogram, i.e., the distribution of the number pixels for each quantized bin, can be defined for each component.

Computationally, the color histogram is formed by discretizing the colors within an image and counting the number of pixels of each color. Since the typical computer represents color images with up to 224 colors, this process generally requires substantial quantization of the color space. The main issues regarding the use of color histograms for indexing involve the choice of color space and quantization of the color space. When a perceptually uniform color space is chosen uniform quantization may be appropriate. If a non-uniform

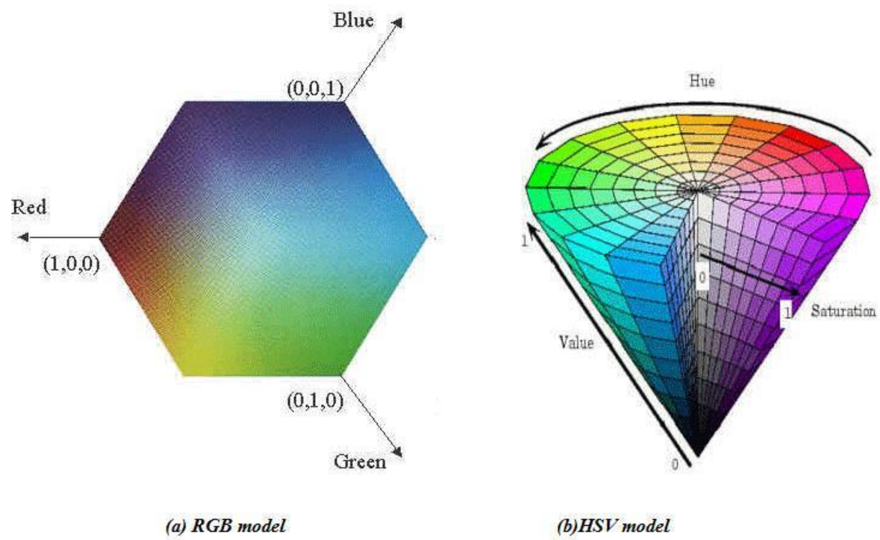



Figure 2.8 : The most popular color models

color space is chosen, then non-uniform quantization may be needed. The color histogram can be thought of as a set of vectors. For gray-scale images these are two dimensional vectors. One dimension gives the value of the gray-level and the other the count of pixels at the gray-level. For color images the color histograms are composed of 4-D vectors. This makes color histograms very difficult to visualize.

There are several distance formulas for measuring the similarity of color histograms using SCHD. In this work ,we use the euclidean distance. Let $h$ and $g$ represent two color histograms. The euclidean distance between the color histograms h and g can be computed as:

$$d^2(h,g) = \sum_A \sum_B \sum_C (h(a,b,c) - g(a,b,c))^2$$

In this distance formula, there is only comparison between the identical bins in the respective histograms. Two different bins may represent perceptually similar colors but are not compared cross-wise. All bins contribute equally to the distance.

#### 2.1.2.2 Fuzzy Color Histogram Descriptor (FCHD)

The fuzzy color histogram (FCHD), is a histogram descriptor that considers the color similarity of each pixel's color associated to all the histogram bins through fuzzy-set membership function.

Because each histogram bin represents a local color range in the given color space, color histogram represents the coarse distribution of the colors in an image. Two similar colors will be treated as identical provided that they are allocated into the same histogram bin. On the other hand, two colors will be considered totally different if they fall into two different bins even though they might be very similar to each other. This makes color histograms sensitive to noisy interference such as illumination changes and quantization errors.Thus,fuzzy color histogram descriptor (FCHD) created to efficiently solve this aforementioned issue.

In contrast with Simple color histogram Descriptor (SCHD) which assigns each pixel into one of the bins only, FCHD considers the color similarity information by spreading each pixel's total membership value to all the histogram bins. FCHD is less sensitive to noisy interference such as lighting in- tensity changes and quantization errors than SCHD. Moreover, in contrast with quadratic histogram distance exploited for measuring the degree of similarity between SCHD's, simple Euclidean distance measurement over their FCHDs can yield similar retrieval results.

The color histogram is viewed as a color distribution from the probability viewpoint. Given a color space containing n color bins, the color histogram of image containing N pixels is represented as $H(I) = [h_1, h_2, h_3....., h_n]$,where $h_i = N_i/N$ is the probability of a pixel in the image belonging to the ith color bin, and $N_i$ is the total number of pixels in the ith color bin.

According to the total probability theory,$h_i$ can be defined as follows:

$$h_i = \sum_{j=1}^{N} P_{i|j} P_j = \frac{1}{N} \sum_{j=1}^{N} P_{i|j}$$

where $P_j$ is the probability of a pixel selected from image I being the jth pixel, which is $\frac{1}{N}$, and $P_{i|j}$ is the conditional probability of the selected jth pixel belonging to the ith color bin.In the context of SCHD, $P_{i|j}$ is defined as:

$$P_{i|j} = \begin{cases} 1, & if\, the\, jth\, pixel\, is\, quantized\, into\, the\, ith\, color\, bin \\ 0, & otherwise \end{cases}$$

This definition leads to the boundary issue of SCHD such that the histogram may undergo abrupt changes even though color variations are actually small. This reveals the reason why the SCHD is sensitive to noisy interference such as illumination changes and quantization errors.

The FCHD essentially modifies probability $P_{i|j}$ as follows. Instead of using the probability $P_{i|j}$ , we consider each of the N pixels in image I being related to all the n color bins via fuzzy-set membership function such that the degree of "belongingness" or "association" of the jth pixel to the ith color bin is determined by distributing the membership value of the jth pixel, $\mu_{ij}$, to the ith color bin.

The fuzzy color histogram descriptor (FCHD) of image I can be expressed as

$$F(I) = [f_1, f_2, f_3, ...., f_n]$$

,where

$$f_i = \sum_{j=1}^{N} \mu_{ij} P_j = \frac{1}{N} \sum_{j=1}^{N} \mu_{ij},$$

$P_j$ has been defined in (1), and $\mu_{ij}$ is the membership value of the jth pixel in the ith color bin.

In contrast with SCHD, FCHD considers not only the similarity of different colors from different bins but also the dissimilarity of those colors assigned to the same bin. Therefore, FCHD effectively alleviates the sensitivity to the noisy interference.A usefull reference for this descriptor is [24]

#### 2.1.2.3 Auto - Corellogram Descriptor (ACD)

The auto (color) correlogram Descriptor (ACD) expresses how the spatial correlation of pairs of colors changes with stance.ACD includes the spatial correlation of colors, can be used to describe the global distribution of local spatial correlation of colors, it is easy to compute, and the size of the feature is fairly small.

Let $I$ be an $n \times n$ image . (For simplicity, we assume that the image is square.) The colors in $I$ are quantized into $m$ colors $c_1, c_2, c_3....., c_m$ .(In practice $m$ , is a constant)For a pixel $p = (x, y) \in I$,let $I(p)$ denote its color.Let $I_c = \{p|I(p) = c\}$.Thus, the notation $p \in I_c$ is synonymous with $p \in I, I(p) = c$. For convenience, we use the $L_\infty$-norm to measure the distance between pixels, i.e., for pixels $p_1 = (x_1, y_1), p_2 = (x_2, y_2)$,we define $|p_1 - p_2| = max\{|x_1 - x_2|, |y_1 - y_2|\}$.We denote the set $\{1, 2 \ldots, n\}$by $[n]$.The histogram $h$ of $I$ is defined for $i \in [m]$ by:

$$h_{ci}(I) = n^2 Pr[p \in I_{ci}]$$

For any pixel in the image ,$h_{ci}(I)/n^2$, gives the probability that the color of the pixel is $c_i$.

Let a distance $d \in [n]$ be fixed a priori. Then, the correlogram of $I$ is defined for $i, j \in [m], k \in [d]$ as:

$$\gamma^{(k)}_{ci,cj}(I) =_{p_1 \in I_{ci} p_2 \in I} Pr[p_2 \in I_{c_j} | \, |p_1 - p_2| = k]$$

Given any pixel of color $c_i$ in the image, $\gamma^{(k)}_{c_i,c_j}$ gives the probability that a pixel at distance $k$ away from the given pixel is of color $c_j$ .Note that the size of the correlogram is $O(m^2d)$ . The autocorrelogram of $I$ captures spatial correlation between identical colors only and is defined by:

$$\alpha^{(k)}_c(I) = \gamma^{(k)}_{c,c}(I)$$

For example consider the simple case when $m = 2$ and $n = 8$. Two sample images are shown below in Figure 2.9(a). The autocorrelograms corresponding to these two images are shown in Figure 2.9(b). The change of autocorrelation of the foreground color with distance is perceptibly different for these images. Note that it is difficult to distinguish between these two images using histograms.More details about corellograms can be found here [25].

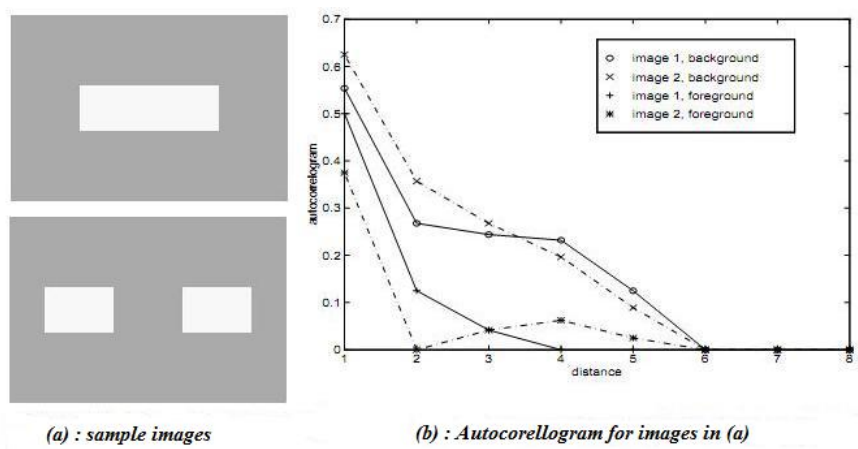



Figure 2.9 : Example of autocorellograms (b) for the images in (a)

## 2.2 Texture Descriptors:

Image texture has emerged as an important visual primitive to search and browse through large collections of similar looking patterns. An image can be considered as a mosaic of textures and texture features associated with the regions can be used to index the image data. Color descriptors has shown good performance for discriminating images based on color. However, in many cases, color is usually insucient for discriminating between images with the same color but dierent texture. Texture features are capable of recognizing repeated patterns in an image, analyzing the energy distribution in the frequency domain.

### 2.2.1 Edge Histogram Descriptor (EHD)

The edge histogram descriptor (EHD) captures the spatial distribution of edges, somewhat in the same spirit as the CLD.The edge histogram descriptor represents the spatial distribution of five types of edges, namely four directional edges and one non-directional edge.

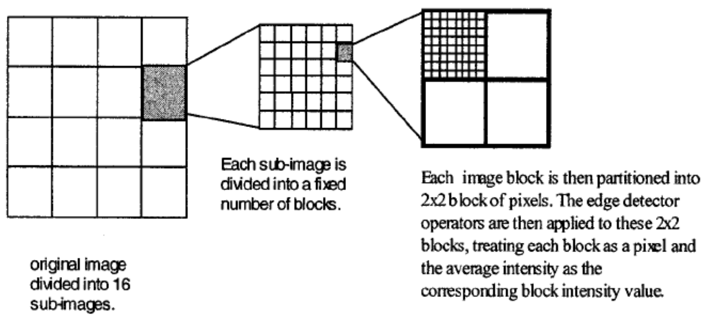



Figure 2.10 :Subimages and macroblocks decompositions

Since edges play an important role for image perception, it can retrieve images with similar semantic meaning. Thus, it primarily targets image-to-image matching (by example or by sketch), especially for natural images with non-uniform edge distribution. In this context, the image retrieval performance can be significantly improved if the edge histogram descriptor is combined with other Descriptors such as the color histogram descriptor. Besides, the best retrieval performances considering this descriptor alone are obtained by using the semi-global and the global histograms generated directly from the edge histogram descriptor as well as the local ones for the matching process.

This descriptor is implemented as follows: Firstly, a gray-intensity image is divided n 4 4 sub-images. Each sub-image has its own local histogram with 5 bins. These 5 ins correspond to the 5 edge types: vertical, horizontal, 45 diagonal, 135 diagonal, and otropic.

In order to ll the local histograms each sub-image is divided in macroblock. A macroblock is composed by 22 macropixels and is associated to an edge type. Figure 2.10 gives details on how the macroblock are built.

To associate a macroblock with an edge type, a convolution with 5 simple edge detectors is performed and the one with the strongest reply is linked to the macroblock:

$$\begin{vmatrix} 1 & -1 \\ 1 & -1 \end{vmatrix} \begin{vmatrix} 1 & 1 \\ -1 & -1 \end{vmatrix} \begin{vmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{2} \end{vmatrix} \begin{vmatrix} 0 & \sqrt{2} \\ \sqrt{2} & 0 \end{vmatrix} \begin{vmatrix} 2 & -2 \\ -2 & 2 \end{vmatrix}$$

The local histogram is therefore built counting the result of each macroblock. Finally the global histogram is quantied in 3 bits per bin (EHD(i) 2 [0 7]).

Note that there are a total of 80 bins, 3 bits/bin, in the edge histogram. One can use the 3-bit number as an integer value directly and compute the L1 distance between two edge histograms.More Information about EHD can be found in [26].

### 2.2.2 Tamura Descriptor (TD)

Tamura et. al.[?] proposed texture features that correspond to human visual perception. They dened six textural features (coarseness, contrast, directionality, line-likeness, regularity and roughness) and compared them with psychological measurements for human subjects. The first three features described below attained very successful results and are used in our evaluation, both separately and as joint values.

Coarseness has a direct relationship to scale and repetition rates and was seen by Tamura as the most fundamental texture feature. An image will contain textures at several scales coarseness aims to identify the largest size at which a texture exists, even where a smaller micro texture exists. Computationally one first takes averages at every point over neighborhoods the linear size of which are powers of 2. The average over the neighborhood of size 2k 2k at the point (x, y) is:

$$A_k(x,y) = \sum_{i=x-2^{k-1}}^{x+2^{k-1}-1} \sum_{j=y-2^{k-1}}^{y+2^{k-1}-1} f(i,j)/2^{2k}$$

Then at each point one takes differences between pairs of averages corresponding to on-overlapping neighborhoods on opposite sides of the point in both horizontal and vertical orientations. In the horizontal case this is:

$$E_{k,h}(x,y) = |A_k(x+2^{k-1},y) - A_k(x-2^{k-1},y)|$$

At each point, one then picks the best size which gives the highest output value, where $k$ maximizes $E$ in either direction. The coarseness measure is then the average of $S_{opt}(x,y) = 2^{k_{opt}}$

Contrast aims to capture the dynamic range of grey levels in an image, together with the polarization of the distribution of black and white. The first is measured using the standard deviation of grey levels and the second the kurtosis a4. The contrast measure is therefore dened as:

$$F_{con} = \sigma/(\alpha_4)^n$$

where,

$$\alpha_4 = \mu_4/\sigma^4$$

Experimentally, Tamura found n = 1/4 to give the closest agreement to human measurements. This is the value we used in our experiments.

Directionality is a global property over a region. The feature described does not aim to differentiate between different orientations or patterns, but measures the total degree of directionality. Two simple masks are used to detect edges in the image. At each pixel the angle and magnitude are calculated. A histogram, $Hd$ , of edge probabilities is then built up by counting all points with magnitude greater than a threshold and quantizing by the edge angle. The histogram will reflect the degree of directionality. To extract a measure from $Hd$ the sharpness of the peaks are computed from their second moments. Finally distances between images vectors were calculated upon feature vectors using the Manhattan metric.An intresting reference to tamure descriptor and its image retrieval use is [27]

### 2.2.3 Gabor Descriptor (GD)

Gabor descriptor is a texture descriptor based on a multiresolution decomposition using gabor wavelets and is based on gabor filtering [28].

The descriptor has two parts: The first part relates to a perceptual characterization of texture in terms of structuredness, directionality and coarseness (scale).This part is called the perceptual browsing component (PBC). The second part provides a quantitative description that can be used for accurate search and retrieval and is referred to as the similarity retrieval component (SRC).Both of the components are derived from a multiresolution Gabor filtering.

From the multiresolution decomposition, a image is decomposed into a set of filtered images. Each of these images represents the image information at a certain scale and at a certain orientation.The PBC captures the regularity (or the lack it) in the texture pattern.

$$PCB = [u_1, u_2, u_3, u_4, u_5].$$

Regularity ($u_1$): $u_1$ represents the degree of regularity or structuredness of the texture. A larger value of $u_1$ indicates a more regular pattern. Consider the two patterns in the Figure 2.11 . Pattern Figure 2.11(a) is intuitively more regular than Figure 2.11(b), and hence should have a larger $u_1$ compared to Fig.2.11 (b).

Directionality ($u_2$, $u_3$ ): These represent the two dominant orientations of the texture. The accuracy of computing these two components often depends on the level of regularity of the texture pattern.

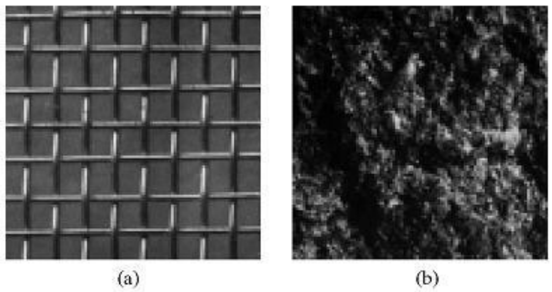



Figure 2.11 : Regularity of patterns (a)regular pattern, (b)irregular pattern

Scale $(u_4, u_5)$: These represent two dominant scales of the texture.Similar to directionality, the more structured the texture, the more robust the computation of these two components.

FInally to Compute the similarity retrieval component (SRC)

The mean $\mu_{mn}$ and the standard deviation $\sigma_{mn}$ of the magnitude of the transform coefficients are used to form the SRC:

$$\mu_{mn} = \int\int |W_{mn}(x,y)|dxdy$$

and

$$\sigma_{mn} = \sqrt{\int\int (|W_{mn}(x,y)| - \mu_{mn})^2 dxdy}$$

The similarity retrieval component (SRC) vector is now constructed using $\mu_{mn}$ and $\sigma_{mn}$ . For S scales and K orientations, this results in a vector

$$SRC = [\mu_{11}\sigma_{11} \ldots \mu_{SK}\sigma_{SK}]$$

In order to use this descriptor to perform similarity retrieval,we use a distance measure on the feature vector of GD.Consider two image patterns $i$ and $j$. Then the distance between the two patterns is:

$$d(i,j) = \sum_m \sum_n d_{mn}(i,j)$$

where

$$d_{mn}(i,j) = |\frac{\mu_{mn}^{(i)} - \mu_{mn}^{(j)}}{\alpha(\mu_{mn})}| + |\frac{\sigma_{mn}^{(i)} - \sigma_{mn}^{(j)}}{\alpha(\sigma_{mn})}|$$

$\alpha(\mu_{mn})$ and $\alpha(\sigma_{mn})$ are the standard deviations of the respective features over the entire database, and are used to normalize the individual feature components.

## 2.3 Hybrid Descriptors:

Hybrid descriptors can be formulated by incorporating color and texture to a new descriptor. We will describe three low-level descriptors ,used in this work,which contain both color and texture information.

### 2.3.1 Color Edge Directivity Descriptor (CEDD)

The Color edge directivity descriptor (CEDD) is a descriptor that combines, in one histogram, color and texture information[29]. CEDD size is limited to 54 bytes per image, rendering this descriptor suitable for use in large image databases.

First, the image is devided in a fixed number of blocks (eg 3x3). In order to extract the color information, a set of fuzzy rules undertake the extraction of a Fuzzy-Linking histogram. This histogram stems from the HSV color space. Twenty rules are applied to a three-input fuzzy system(one for each HSV) in order to generate eventually a 10- bin quantized histogram. Each bin corresponds to a preset color. The number of blocks assigned to each bin is stored in a feature vector. Then, 4 extra rules are applied to a two nput fuzzy system, in order to change the 10- bins histogram into 24-bins histogram, mporting thus information related to the hue of each color that is presented.

Next, the 5 digital filters that were proposed in the MPEG-7 Edge Histogram Decriptor (see 2.2.2) are also used for exporting the information which is related to the exture of the image, classifying each image block in one or more of the 6 texture regions hat has been fixed, shaping thus the 144 bins histogram.With the use of the Gustafson Kessel fuzzy classier 8 regions are shaped, which are hen used in order to quantize the values of the 144 CEDD factors in the interval 0-7,limiting thus the length of the descriptor in 432 bits You can see the schematic diagram of CEDD in Figure 2.12

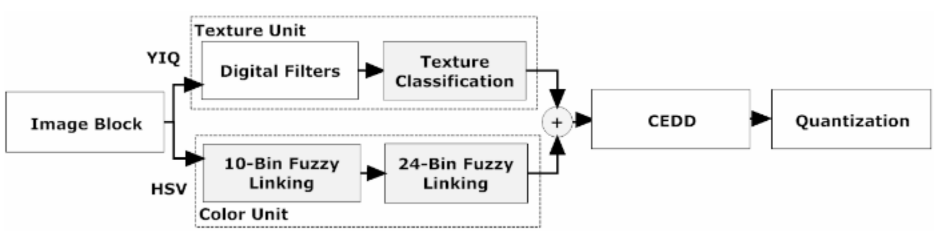



Figure 2.12 :A schematic diagram of CEDD

For the measurement of the distance of CEDD between images, Tanimoto coecient is used:

$$T_{ij} = t(x_i, x_j) = \frac{x_i^T x_j}{x_i^T x_i + x_j^T x_j - x_i^T x_j}$$

Where xT is the transpose vector of x.

### 2.3.2 Fuzzy Color Texture Histogram Descriptor (FCTHD)

The Fuzzy Color Texture Histogram Descriptor (FCTHD) results from the combination of 3 fuzzy systems. FCTHD size is limited to 72 bytes per image. FCTHD works exactly the same with CEDD with a little difference in texture information extraction.

Initially the image is segmented in a present number of blocks. Next extracts the same color information as CEDD. For the extraction of texture information each image block is transformed with Haar Wavelet transform and a set of texture elements are exported. These elements are used as inputs in a third fuzzy system which converts the 24-bins histogram in a 192- bins histogram, importing texture information in the proposed feature. Eight rules are applied in a three-input fuzzy system. For the quantization process Gustafson Kessel fuzzy classiers are used. You can see the schematic diagram of FCTH in Figure 2.13

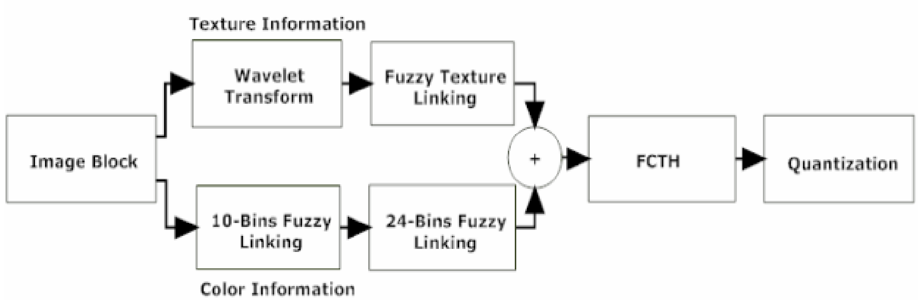



Figure 2.13 : A schematic diagram of FCTH

For the measurement of the distance of FCTH between the images, Tanimoto coefficient is used. Additional information about the descriptor can be found at [30].

### 2.3.3 Joint Composite Descriptor (JCD)

Joint Composite Descriptor (JCD)is a combined compact vector that contains color and texture information at the same time.JCD succesfully combines CEDD and FCTH[31].

As mentioned above,The structure of CEDD and FCTH descriptors consists of n texture areas.Each texture area is separated into 24 sub regions, with each sub region describing a color. CEDD and FCTH use the same color information, as it results from 2 fuzzy systems that map the colors of the image in a 24-color custom palette.To extract texture information, CEDD uses a fuzzy version of the five digital filters proposed by the EHD forming 6 texture areas.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| **CEDD** | Linear | Non Directional | Horizontal Activation | Vertical Activation | 45 Degree Diagonal | 135 Degree Diagonal | - | - |
| **FCTH** | Linear Low Energy | Horizontal Low Energy | Vertical Low Energy | Both Directions Low Energy | Linear High Energy | Horizontal High Energy | Vertical High Energy | Both Directions High Energy |

Figure 2.14 : Compact composite descriptors texture areas

In contrast, FCTH uses the high frequency bands of the Haar wavelet Transform in a fuzzy system, to form 8 texture areas. The types of texture areas adopted by each descriptor are illustrated below in Figure 2.14.

This new descriptor is made up of 7 texture areas, with each area made up of 24 sub regions that correspond to color areas. The colors that represent these 24 sub regions are: (0) White, (1) Grey, (2) Black, (3) Light Red, (4) Red, (5) Dark Red, (6) Light Orange, (7) Orange, (8) Dark Or- ange, (9) Light Yellow, (10) Yellow, (11) Dark Yellow, (12) Light Green, (13) Green, (14) Dark Green, (15) Light Cyan, (16) Cyan, (17) Dark Cyan, (18) Light Blue, (19) Blue, (20) Dark Blue, (21) Light Magenta, (22) Magenta, (23) Dark Magenta. The texture areas are as follows: JCD(0) Linear Area, JCD(1) Horizontal Activation, JCD(2) 45 Degrees Activation, JCD(3) Vertical Activation, JCD(4) 135 Degrees Activation, JCD(5) Horizontal and Vertical Activation and JCD(6) Non directional Activation

# Chapter 3

# Ontology Construction

In this chapter,we present the DOGi ontology.We use the domain of dogs (as noted previously) which is a speciliazation of the domain animals.We analyze the main components and the structure of this ontology and we present our conclusions for optimized ontology design.

## 3.1 Ontologies

Ontologies have been proven to be useful for image annotation are very usefull as they provide a way to enhance descriptions with conseptual meanings.In other words,using ontologies a computer is capable of generating meaningfull descriptions rather than letting a human do it(which is time consuming and does not scale-up for large data sets)[14].Furthermore,the use of ontologies helps to recognise and use the possible semantic relations between the described concepts. .For example,we may want to describe small dogs.By using an ontology of dogs,the computer will know that chihuahua is a small dog while an alsatian isn't(alsatian is a big dog which is the opposite of a small dog).In contrary,if we didn't use an ontology of dogs,the computer would have named small dogs (like chihuahua) but it would not be possible to understand that an alsatian is not a small dog.In conclusion,using ontologies as a basis for defining visual vocabulary or as a framework for automatic image annotation increases the number of concepts an image annotation system can recognize and may as a means for improving the performance of image retrievals.

## 3.2 Animals Ontology

The backbone of the animals ontology consists of a simple IS-A hierarchy (a taxonomy) of animals.Our ontology of dogs is a specilization (or in other words ,an expansion) of the animals ontology, as you can see in fugure 3.1.With this approach,a dog belonging to a class in the dogs ontology will inherit properties belonging to upper classes (that does not necessary belong in dogs ontology).For example, a dog in the terrier dog group is a mammal so it inherits the propeties of a mammal etc.

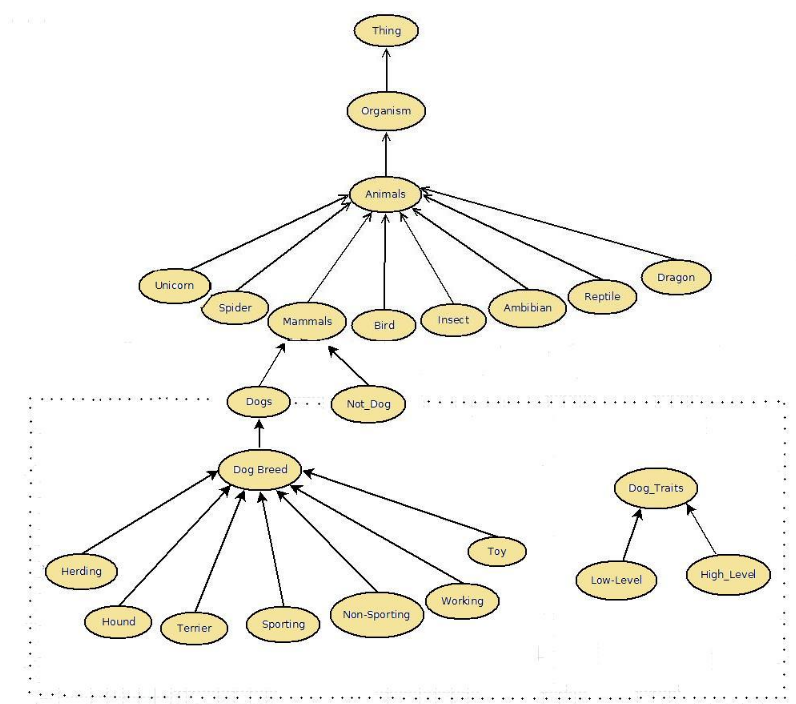



Figure 3.1 : A part of dogs Ontology(inside the box) as a sub-ontology of the animals ontology.

## 3.3 DOGi Ontology

The dogs ontology consists of two main class hierarchies,The dog hierarhy,which is the basic class categorization of dogs and the dog traits hierarhy ,which is the class categorization of the high and low level dog traits(descriptions).These two hierarhies are connected through a number of relations between instances of correspontding classes(see Figure 3.2).

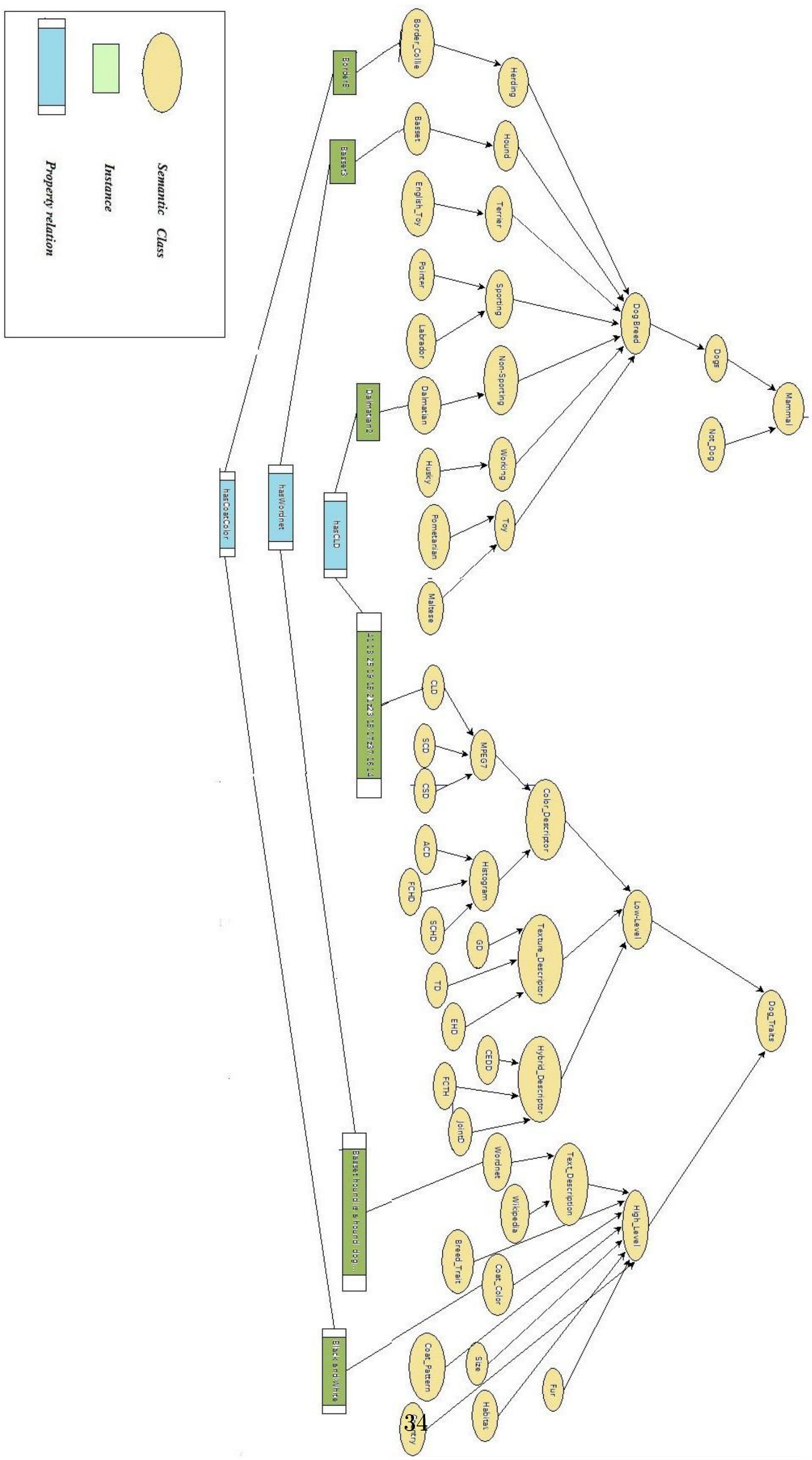



Figure 3.2 : The Dogs Ontology.Note,that not all classes and instances are shown but only some of them.

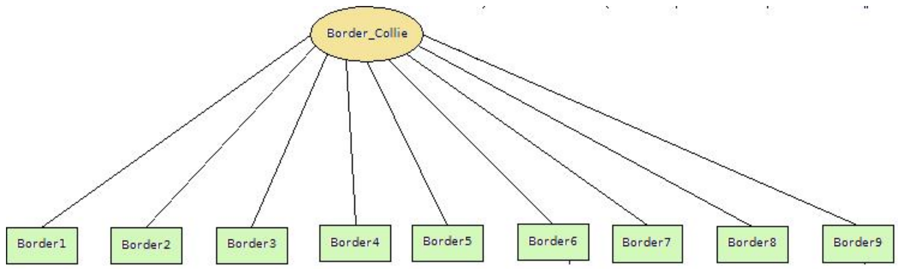



Figure 3.3 : The 9 instances representing the 9 training images used for every dog breed

### 3.3.1 Dog Class Hierarchy

The dog class hierarchy is based on a formal class hierarchy of Wordnet.WordNet is one of the largest conceptual hierarchies available and our approach is easily extendable to other domains.The dog breed taxonomy is generated following the nouns hierarchy of Wordnet (i.e., dog,herding group,border collie...etc.). The 40 leaf classes in the dog hierarchy represent the different semantic categories(dog breeds) our framework can recognize.For each semantic class,we have assigned 9 instances , each representing a training image of the corresponding breed.For example the leaf class Border_Collie has 9 instances ,the same number of images used for training for the dog breed border collie (see Figure 3.3).These instances are connected with corresponding traits belonging to the dog traits hierarhy.For example,the instance border1 is connected with the low level trait instance cedd1(which is the cedd vector extracted from the 1st training image of border collie) through a property relation hasCedd,ie border1 hasCedd cedd1...etc.All relations will be discussed later in detail.

### 3.3.2 Dog Traits Class Hierarchy

Dog traits class hierarchy, is an hierarchy of dog traits which are used in our framework for constructing meaningfull descriptions.Dog traits hierarhy is divided into two main components.Low level and high level.

Low level traits are all the corresponding low level features used in image content analysis(see chapter 2 for more details) to recognize an image by its content.In this ontology,the low level traits are used only for the training of our system and not for the annotation.This means that only the 9 instances of every of 40 leaf semantic classes are connected to the low level traits.

High level traits are the traits that can fully describe a dog breed i.e., a text description, color ,country of origin etc.In our ontology,we have included the following high level traits:

***1.Wordnet Text Description*:**

This is the text description of wordnet for the corresponding dog breed.For example ,border collie has the following text description :"Border collie – (developed in the area between Scotland and England usually having a black coat with white on the head and tip of tail used for herding both sheep and cattle)"

***2.Wikipedia Text Description:***

This is the text descriptioin of wikipedia for the corresponding dog breed.For example,border collie has the following text description:"The Border Collie is a herding dog breed developed in the Anglo-Scottish border region for herding livestock, especially sheep. It is the most widespread of the collie breeds."

***3.Breed Trait* :**

This is the breed trait of the corresponding dog breed.For example ,border collie has breed trait herding because was originally bred for hearding.

***4.Coat Color*:**

The color of the dog breed.For example,border collie has black and white coat color.

***5.Coat Pattern*:**

The pattern of the coat,i.e., if the dog breed has spots ,lines,multicolors etc.

***6.County*:**

The country of origin of the dog breed

***7.size*:**

The size of the dog breed.There are three options.small,medium and big.

***8.fur*:**

The fur that the dog breed has.There are three options:smouth ,rough,no-fur.

The aforementioned high level traits are used to generate the annotation that could describe a dog breed recognised in an input image by our CBIR system.You should read chapter 5 and see annotation examples to fully understand the conspepts described here.

Summarizing, the dogs ontology (shown in Figure 3.2) consists for the following parts: a) A nouns class hierarchy of dog breeds with instances of leaf classes representing images and b) Dog traits arranged in class hierarchies with instances of leaf classes representing low-level description and high level information. Semantic categories (images) are also associated to low-level description and high level information. Object properties are used to connect instances of semantic classes (images) with instances from classes containing description (low-level or high level).The dog taxonomy, visual text description and text description were generated manually using Protege.Low-level description and association to images were generated using Protege-Owl api . Image feature extraction is implemented using Lire .Finally an image ontology in OWL language under the domain of dog breeds was successfully constructed.

# Chapter 4

# Image Similarity

We use two similarity measures to estimate the similarity between a pair of images: LIRE similarity measure and DOGi similarity measure.LIRE similarity measure is the summary of the normalized (in range 0-1) low-level descriptors while DOGi similarity measure is LIRE similarity measure with weigths for each descriptor computed by a decision tree.

## 4.1 Region Of Interest(ROI)

Figure 4.1: Original Image (left) and Region Of Interest (right)

Our CBIR system starts with an unknown image as input (i.e., the image of a dog in this work).The input image may contain several regions, and it is rather natural to assume that some of them may be more relevant for the user's information need than others (e.g., the foreground or the center of the image might be more relevant than the background or elements towards the boundary of the image).Dog's head is the most representative part of a dog image and thus the most usefull part of the image.In this work, the ROI is a sub-image of

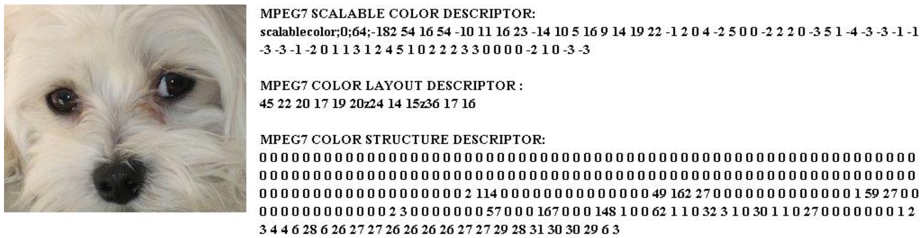



Figure 4.2 : Region Of Interest and the corresponding MPEG7 color descriptors vector representations

the original image containing dog's head with as less as possible background.We let the user select the ROI by excluding a sub-image by the original image with a rectangle selection.There are other approaches like background subtraction or image segmentation but these techniques have unreliable results.In Figure 4.1 you can see an original image and the correponding ROI for this image.

## 4.2 Feature Extraction

After the ROI selection,we extract the image features (image content) from the region of interest.Image features are represented as visual descriptors which are vectors of numbers.For example,color layout descriptor(CLD) has a vector representation of this form : 50 13 22 25 12 27z22 ... while scalable color descriptor(SCD) has this form : scalablecolor;0;64;-173 48 -3 57 4 9 22 30 -24 ...With this vector representation,visual descriptors of a pair of images cannot be compared i.e.,we cannot compare CLD1 descriptor of image1 with CLD2 descriptor of image2.Instead,we can use the distance of these descriptors.The distance between two descriptors of a pair of images is a number from 0 to + infinity(Notice that comparison can be performed only between visual descriptors of the same type,ie. CLD1 with CLD2 ,SCD1 with SCD2 etc).For example, the distance between a pair of CLD descriptors can be a number between 0-470 while SCD can be a number between 0-550.The lower the distance ,the higher the similarity of the pair of descriptors compared.When the distance is zero ,the visual descriptors compared are the same.In Figure 4.2 ,you can see a ROI and the corresponding MPEG7 color descriptors vector representations.

## 4.3 Gaussian Normalization

We compute the similarity of a pair of images as a function of feature distances extracted from the compared images.However,the different range of distances each pair of descriptors have (i.e., range of cld distance is 0-470 but range of scd distance is 0-550 etc.) yield the problem of not giving equal emfasis to each descriptor.We can use normalization techniques to solve this problem.

In order to achieve normalization,we must have a considerable number of distances for each descriptor.In our work,we use nearly 3500 distances for each descriptor based on the comparison between 360 database images( 9 images per dog breed with 40 dog breeds available).

The most simple normalization technique is the L1 metric: for each element i in a ranked list of k elements having distance d, the D normalized distance is:

$$D = \frac{d_i}{d_{max}}$$

Where $d_i$ is the ith distance and $d_{max}$ is the maximum distance

A second popular normalization metric , is linear scaling to unit range ,L2 metric:

$$D = \frac{d_i - d_{min}}{d_{max} - d_{min} + 1}$$

where $d_{max}$and $d_{min}$ are the maximum and minimun values of $d$ .

The above normalization techniques are not usefull when a limited database of images is available.This is because the distances of each descriptor usually fall into a small subrange of the entire possible range. As a result, the linear normalization will possibly compact the distances into a very narrow and indiscriminate range within [0, 1] and distances of descriptors from unknown images (not in our database)will be mapped to different subranges within [0, 1] which makes these distances incomparable with the ones of our database.

Gaussian normalization can solve this problem as it is capable of normalizing distances in a range within [0,1] that follow a gaussian distribution with standard deviation equal to 1.The Gaussian normalization is defined:

$$D = \frac{1}{2}(1 + \frac{d_i - \mu}{3 * \sigma})$$

where $\mu$ is the mean value and $\sigma$ is the standard deviation of the distances computed from our database images.

Thus,the presence of a few abnormally large or small values does not bias the importance of a feature measurement in computing the similarity between two images.In our work,we use the gaussian normalization to normalize distances in the range within [0,1]

## 4.4 Image Similarity Measures

Given an unknown image, the database images (in other words ,the instances) of the ontology are searched using a similarity measure and the most similar to the unknown image are retrieved.In this work,two similarity measures are used:LIRE similarity measure and DOGi similarity measure.Both are described in the next paragraphs

### 4.4.1 LIRE similarity Measure

LIRE similarity measure is a function of LIRE feature distances computed from the compared images.We define LIRE similarity measure as follows:

$$S_{LIRE}(A, B) = \sum_i (1 - d_i)$$

where $A, B$ is the pair of images and $d_i$ is the gaussian normalized distance of the ith descriptor.Note that we use $1 - d_i$ and not $d_i$ distance.This is done because we want the distances range within [0,1] and not in [1,0] as we compute similarity and not distance.In other words,the similarity is increasing when the distance $d_i$ decreases.

LIRE similarity measure can compare pairs of images good enough especially when these images differ a lot.For example,if we had to compare an image containing a building with an image containing a dog, LIRE similarity measure would give a really low similarity denoting that these images do not match.However,in most cases ,images compared are identical therefore we cannot compare them by using only LIRE similarity measure.We need an overall measure that can give relative importance in each descriptor .

### 4.4.2 DOGi Similarity Measure

DOGi similarity measure,is LIRE similarity measure having weights over the summation of normalized descriptors.These weights are computed using a decision tree(see 4.5).we define weigthed similarity measure as follows:

$$S_{DOGi}(A,B) = \sum_i W_i(1 - d_i)$$

where $A, B$ is the pair of images, $d_i$ is the gaussian normalized distance of the ith descriptor and $W_i$ the weight of the ith descriptor.

DOGi similarity measure is better in most of cases than LIRE similarity measure.The weights $W_i$ in the above formula represent the relative importance of the features involved and are computed by a decision tree.

## 4.5 Decision tree

The weights used in DOGi similarity measure are computed by a decision tree.The decision tree relies on the training data set provided.Our training dataset consists of 3474 training instances.These instances represent pair of images and are classified into two categories,similar and not-similar.1584 instances are classified similar and 1890 are classified not-similar.

The decision tree is constructed using the training data set has the descriptors as nodes and classified instances similar or not-similar as leaves.For each pair of images, we compute a vector of distances on all features and the decision tree decides if the images are similar or not (or how similar the two images are). Each branch of the decision tree places a criterion (i.e., threshold) on a feature of this vector based on which the similarity (or the dissimilarity) between the images as a whole is computed. .For example,in our decision tree if the CLD distance of the pair of image is less than 0.31 , we contunue on the left branch of the tree or else we continue on the right branch.Then,comparisons of descriptors continue until we reach a leaf. If the leaf is noted as similar then the pair of images are considered to be similar.Each Descriptor can be in more than one node ,thus the more times a descriptor appears the more important the descriptor is.Moreover,The higher nodes of the tree are more important that the lower ones.Therefore, the most frequently a feature appears in higher nodes of the decision tree, the more important it is.These conclusions are taken by the weights computation formula below.

Baratis in [32] proposed that weights are computed based on properties of a trained decision tree as follows:

$$w_{fi} = \sum_{nodej=f_i} \frac{Maxdepth + 1 - depth(f_i)}{\sum_j Maxdepth + 1 - depth(node_j)}$$

Where $fi$ is every feature, $nodej$ is each node of the decision tree and Maxdepth is the maximum depth of the tree. The summation is taken over all nodes. This formula suggests that the higher a descriptor is and the more frequently it appears, the higher its weight will be.

Weka [33] was used as an interface of testing and visualizing the decision tree. C4.5 (J48) was the learning method applied and stratified cross-validation was the method for testing the decision tree.We choose to prun the decision tree because prunning helps making the tree robust,by excluding less impartment features from computing similarity and preventing the tree of becoming over-fitted to the given data set. .Thus,85% of the original decision tree is kept(setting confidencefactor being equal to 0.25).

# Chapter 5

# Annotation

After the computation of image similarity measures,semantic category is estimated using methods like AVR or best match .Then,using the class estimated , annotation in MPEG7 format is generated and stored in the exif metadata tags of the unknown input image.

## 5.1 image annotation

As mentioned before, image annotation is a process that takes as input an unknown image and assigns to it a label denoting its category along with a text description. In this work the semantic category the query image belongs to, is computed based on the analysis of the retrieval results. After semantic category estimation, query image then inherits all high level information of the category estimated.

## 5.2 Semantic category estimation

Our CBIR system uses five methods to estimate the semantic category of an unknown image.These methods use the retrieval results list which is a ranked list containing the database images of ontology retrieved in a decreasing similarity order.In the next section,we present briefly each method.

### 5.2.1 Best Match

Best match selects the semantic category of the most similar image to the unknown image i.e the class belonging to the retrieved database image with the highest similarity score.

**Algorithm** Semantic Category Estimation using $AVR$

$rankbreedimage(qi1)$ = best first rank in ranked list for picture from breed i for the query q

$\vdots$

$rankbreedimage(qi\mathrm{n})$ = n best rank in ranked list for picture from breed i for the query q

query q
List list= similar retrieved images in decreasing order of similarity
$list[0]$ = best image
**for** eachbreed==i **do**
  AVR[i]=(rankbreedimage(qi1)+ $\cdots$ +rankbreedimage(qin))/n
**end for**
bestavr=bubblesort(AVR,breeds)
bestavr[0]=best breed

Figure 5.1 : AVR algorithm using n instances.

### 5.2.2 Max Occurence

Max Occurence selects the semantic category that has the maximum number of instances in the first 20 answers. If multiple classes have the same number of instances, best match between competing classes is employed

### 5.2.3 AVR

For a query q with a ground-truth size of NG(q), we define rank(k) as the rank of the kth ground-truth image on the top-N result list. The average retrieval rank is then computed as follows:

$$AVR(q) = \sum_{k=1}^{NG(q)} \frac{rank(k)}{NG(q)}$$

AVR is used as the standard metric of semantic category estimation of unknown image (correctly classified image) based on the retrieval results. A ranked list of instances is obtained answering the image query. The result list includes all semantic categories of our ontology (breeds), having a fixed number of instances(nine images per breed). For each of the semantic categories AVR is computed. From all AVR computed(same number as semantic categories), the class having the best AVR (eg the least) would be the semantic class the query image belongs to. If multiple classes have the same AVR, best match is employed between competing classes.In Figure 5.1 you can see the algorithm of AVR for n instances.In our case,n is equal to 9.

### 5.2.4 Summation Of 9 Instances

Summation of 9 instances selects the semantic category (dog breed) belonging to the 9 instances that achieve the highest summation of similarities in the retrieved list.

### 5.2.5 Decision Tree Method

Decision tree method selects the semantic category (dog breed) belonging to the 9 instances that achieve the highest score summation with score depending on the depth in which the instance was decided similar.Lets say that the 9 instances of the dog breed border collie are border1,border2...border9.Assume that :border1 is decided as not similar with the unknown image so its score is zero.border2 is decided similar at depth 1 so its score is 11 ,border3 is decided similar at depth5 so its score is 7 etc.The score summation of these instances would be 0+11+7...etc.The semantic category of the 9 instances that achieve the highest score summation is selected as the dominant category

## 5.3 Annotation Formats-Interoperability

The high level information from which the annotation is generated, are taken from the ontology.Ontologies are described by owl ,the standard web ontology language, therefore it is straight forward to generate annotations in owl format.However owl and rdf are more appropriate in the Semantic Web world while MPEG7 is used as the standard metadata format for exchanging automatic analysis results.

MPEG7 format ensure interoperability with other semantic web applications. The semantic class the query image belongs to is estimated. Information about the class is encapsulated in the ontology with properties of OWL language. An additional important issue is to achieve semantic interoperability between OWL and MPEG-7. The final step includes mapping owl classes and properties of them to MPEG-7 format so that the multimedia content services offered by different vendors may interoperate. Our main objective, in the final annotation, is to describe the owl class recognized, its superclass and some descriptive owl object properties containing high level information. Transformation rules, between OWL and Mpeg-7, were adopted in order to achieve semantic interoperability.

Tsinaraki et.al [34] proved that OWL Ontologies can be transformed into MPEG-7 Abstract Semantic Entity Hierarchies. OWL domain ontology classes and individu- als are represented as MPEG-7 semantic elements of type 'SemanticBaseType'. The AbstractionLevel' element of the 'SemanticBaseType' and the MPEG-7 semantic relationships are used to capture ontology semantics. An abstract semantic entity that represents a domain ontology class is related with each of its subclasses through a pair of 'Relation' elements of type 'generalizes'/' specializes'. The properties dened in the domain ontology classes are transformed into 'Property' elements (datatype properties).

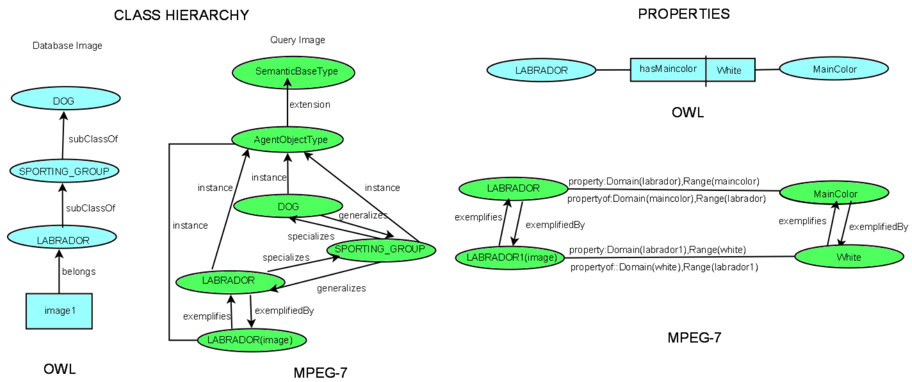



Figure 5.2 : Mapping OWL to MPEG7

Since the image is classified in one of known classes, it is considered as an individual of its corresponding class. Some contents of the annotation file are the same for every query image. This includes description of the ontology, some abstract semantic entities and abstract semantic properties (i.e., dog class).

Next ,the semantic class recognized (dog breed) and its superclass (group) are represented with semantic elements of type 'SemanticBaseType'.Their abstraction connection are represented with 'relation' elements of type 'generalizes'/'specializes'. The query image is represented as individual through pairs of exemplies/exempliedBy relationships with the semantic entity is related to. In the 'SemanticBaseType' representing the query image an element of 'MediaOccurenceType' is added to provide the 'URI' of the query image. Figure 5.2. Trasformation rules where applied thanks to connection between Protege-Owl api , and MPEG-7 MDS (Multimedia Description Schemes) api [35].

## 5.4 Exif Metadata Tags

After the annotation is generated ,it must be stored as metadata along with the image file.In this work , we choose to store annotations in the exif metadata tags because exif supports the most popular file formats like jpeg,tiff and wav.

Exchangeable image file format (Exif) is a specification for the image file format used by digital cameras and scanners[36]. The metadata tags defined in the Exif standard are used to cover Date and time information,camera model , orientation (rotation), aperture, shutter speed, focal length, metering mode, and ISO speed information.

In Figure 5.3 ,you can see the exif tags(not all of them) of the corresponding image.We use the annotations metadata tag of exif to store our annotations .

*(a) an image*

**EXIF TAGS :**
**Make: 'Canon'**
**Model: 'Canon PowerShot S60'**
**Orientation: 1**
**XResolution: 392**
**YResolution: 392**
**Resolution Unit: 2**
**Software: 'Adobe Photoshop 7.0'**
**Modify Date: '2005:12:23 09:35:33'**
**YCbCr Positioning: 1**
**Unknown Tag (0x1001): 2592**
**Unknown Tag (0x1002): 1944**
**Rating: 0**
**Unknown Tag (0x4a04): 1, 0, 116, 108, 10**
**Unknown Tag (0x5012): 1**
**Exif Offset: 7850**
**Unknown Tag (0xa401): 0**
**Unknown Tag (0xa402): 0**
**Unknown Tag (0xa403): 0**
**Unknown Tag (0xa404): 1**
**Unknown Tag (0xa406): 0**
**Exposure Time: 1/125 (0,008)**
**FNumber: 53/10 (5,3)**

*(b) exif tags of the image (a)*

Figure 5.3 : Exif tags (b) of an image (a)

# Chapter 6

# Experiments

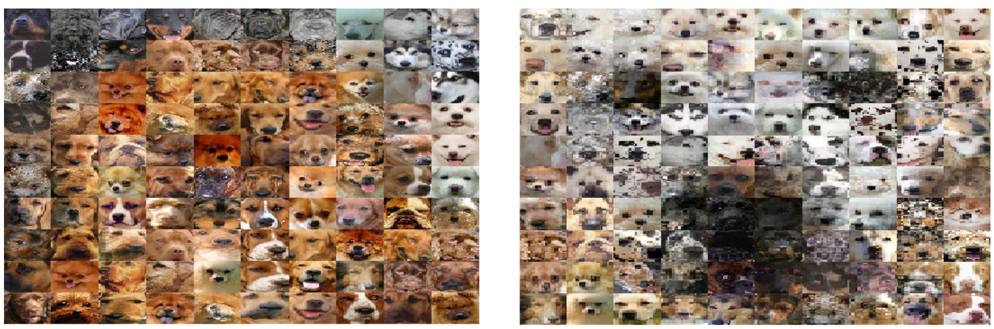


Figure 6.1 : Part of 360 database images

In this Chapter , we present our experiments.We evaluate our system using 40 image queries containing dogs(Figure 6.2).For each query image ,a ROI is manualy selected and searched by its extracted visual content.Our database consists of 360 images (containing dogs), 9 images(instances) for each dog breed(Figure 6.1).

In 6.1 section,the retrieved results for each query , using Lire similarity measure and DOGi similarity measure, are presented and discussed.In the next sections 6.2,6.3 DOGi similarity measure performance when changing the prunning factor and the decision tree used are presented and discussed.In section 6.4,Performance of our system when changing the number of categories recognized is presented and discussed.In section 6.5 we present and compare the results of the annotation methods used.Finally,In section 6.6 ,we present many examples of complete MPEG7 annotations generated and stored automatically for an input image.

Our basic IR evalutation tools are precision and recall:

$$precision = \frac{|\{Relevant\ Images\} \cap \{Retrieved\ Images\}|}{|\{Retrieved\ Images\}|}$$

$$recall = \frac{|\{Relevant\ Images\} \cap \{Retrieved\ Images\}|}{|\{Relevant\ Images\}|}$$

Precision and recall help us to measure how well our system performs.In our work we used average retrieval-recall as a quality measure:

$$P(r) = \sum_{i=1}^{N_q} \frac{P_i(r)}{N_q}$$

where $N_q$ number of queries and $P_i(r)$ is precision at recall level r for i th query.

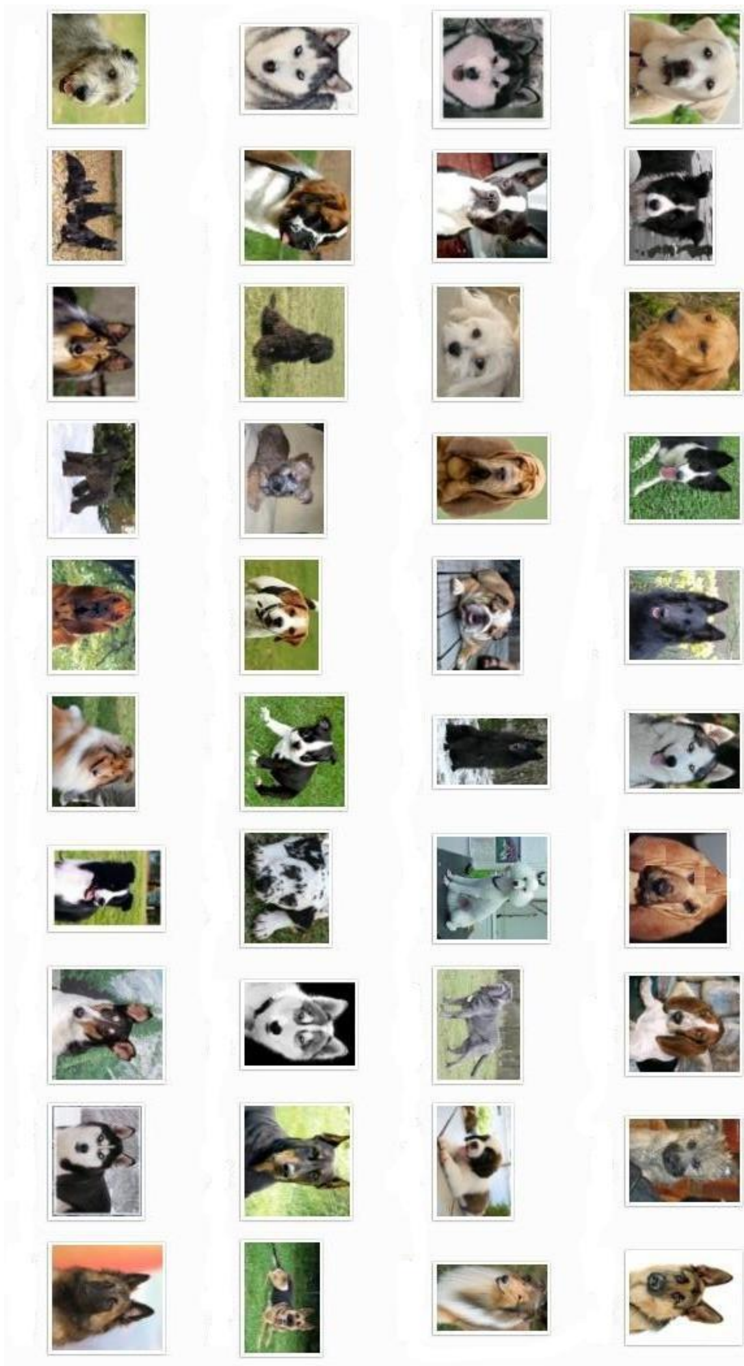


Figure 6.2 : The 40 query images used for system evalution

## 6.1 LIRE Similarity Measure vs DOGi Similarity Measure

We used 40 queries containing dogs (Figure 6.2) and we compute the precision recall for the first 40 answers.For each query tested, the 9 instances of the semantic class the query actually belongs to ,are noted as the relevant images.In Figure 6.3 ,you can see the precision recall diagram of the similarity measures used.We can see that DOGi similarity outperforms LIRE similarity.In fact,DOGi similarity in most of the cases corrected the similarity result list by increasing similarities of relevant images or decreasing similarities of irrelevant ones.

However,DOGi similarity measure doesn't always improve the LIRE similarity and unfortunately it may also worsen similarities .There are few reasons for this.One reason is that the training set used in the decision tree has only 9 instances for every semantic class.If we increase the number of instances used ,we would evantually have better results but the complexity would increase too.Another reason is the quality of the 9 images used for every semantic class.If these images are not representive enough ,then DOGi similarity measure could produce worst results from query images belonging to these instance's semantic classes.Also,as stated in 1.1,similarity between image features (extracted by image analysis) does not always correspond to semantic similarity as perceived by humans.Practically,This means that there is a limit on how much we can improve similarity measures based on visual descriptors.

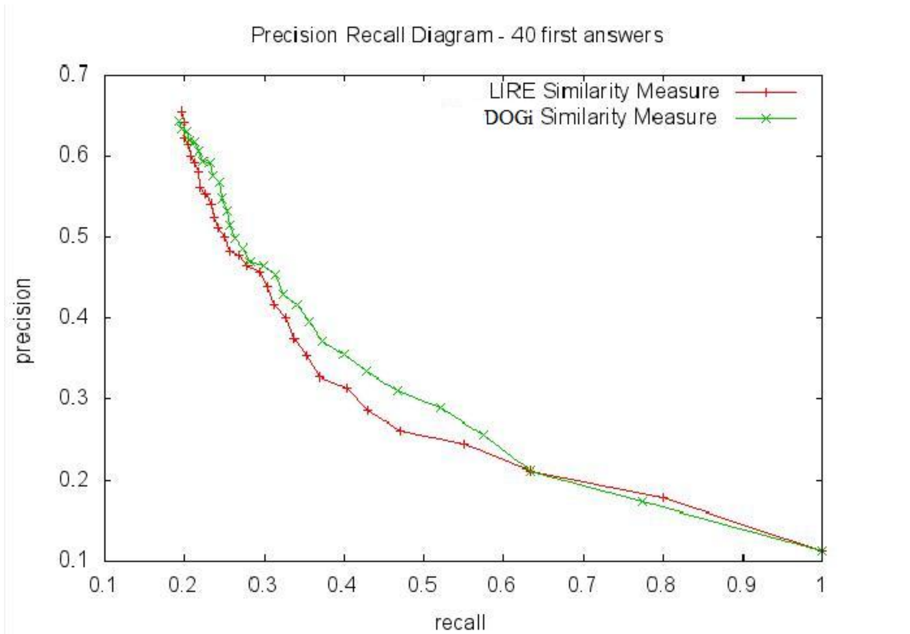
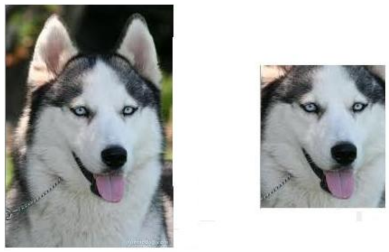


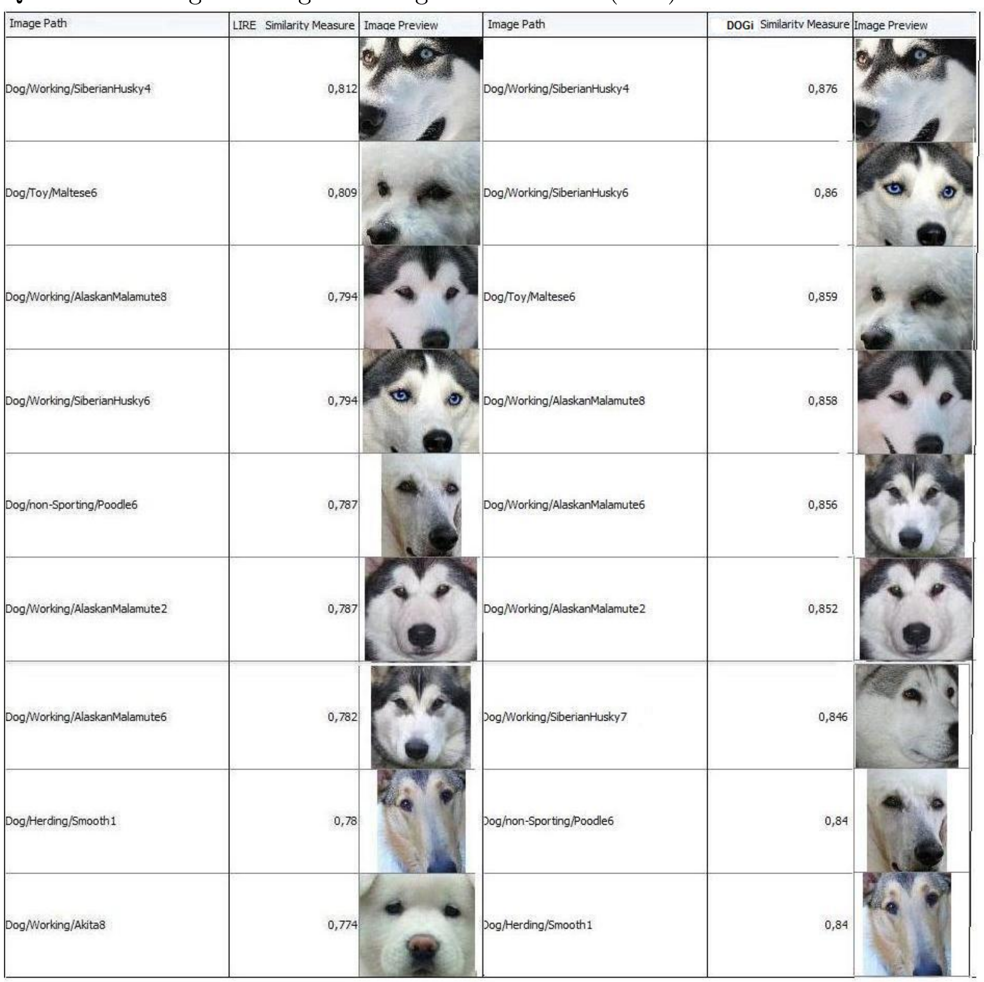

Figure 6.3 :Precision-Recall Diagram for 40 first answers using LIRE similarity measure and DOGi similarity measure

Below,we present five example queries and their corresponding result lists using LIRE similarity measure and DOGi similarity measure

*Note that LIRE and DOGi similarity measures are normalized in range between [0,1] using linear normalization.



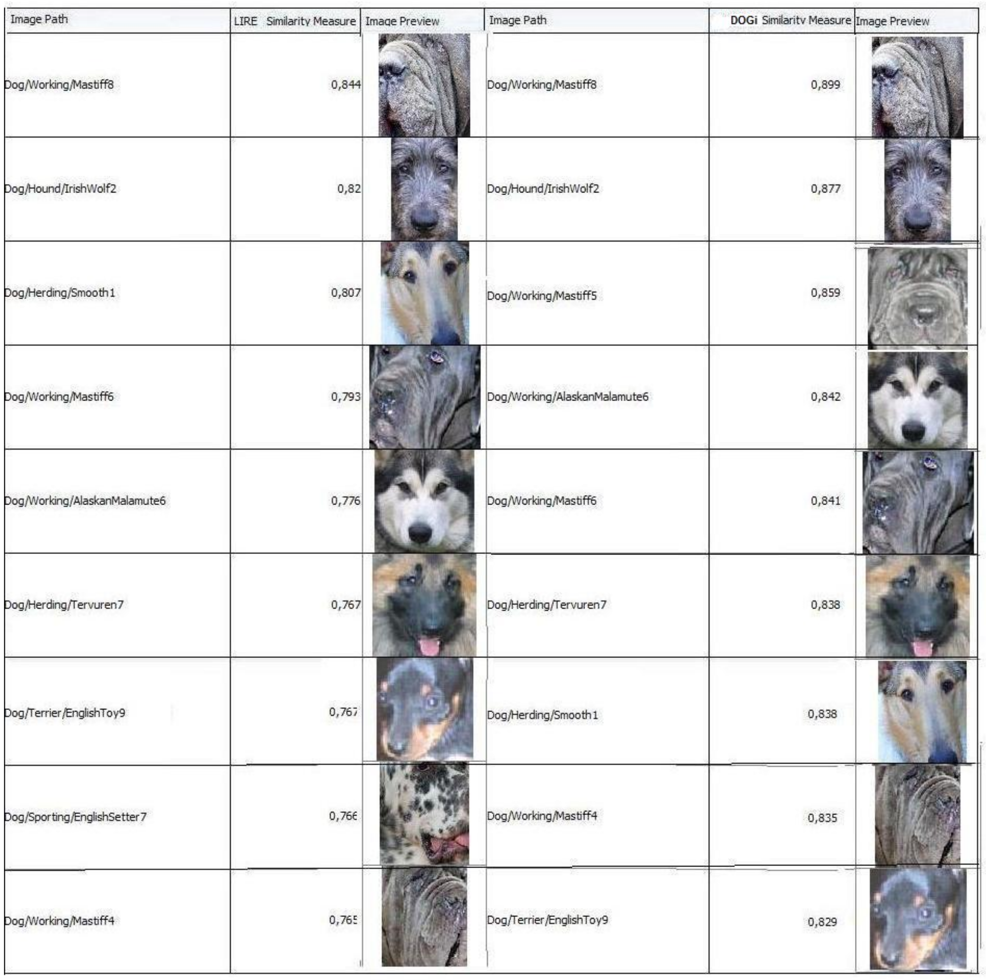
QUERY 1 : Original Image and Region Of Interest (ROI)



| Image Path | LIRE Similarity Measure | Image Preview | Image Path | DOGi Similarity Measure | Image Preview |
|---|---|---|---|---|---|
| Dog/Working/SiberianHusky4 | 0,812 | | Dog/Working/SiberianHusky4 | 0,876 | |
| Dog/Toy/Maltese6 | 0,809 | | Dog/Working/SiberianHusky6 | 0,86 | |
| Dog/Working/AlaskanMalamute8 | 0,794 | | Dog/Toy/Maltese6 | 0,859 | |
| Dog/Working/SiberianHusky6 | 0,794 | | Dog/Working/AlaskanMalamute8 | 0,858 | |
| Dog/non-Sporting/Poodle6 | 0,787 | | Dog/Working/AlaskanMalamute6 | 0,856 | |
| Dog/Working/AlaskanMalamute2 | 0,787 | | Dog/Working/AlaskanMalamute2 | 0,852 | |
| Dog/Working/AlaskanMalamute6 | 0,782 | | Dog/Working/SiberianHusky7 | 0,846 | |
| Dog/Herding/Smooth1 | 0,78 | | Dog/non-Sporting/Poodle6 | 0,84 | |
| Dog/Working/Akita8 | 0,774 | | Dog/Herding/Smooth1 | 0,84 | |

Retrieved Results using LIRE similarity(left) and DOGi Similarity(right) for QUERY1.

QUERY2: Original Image and Region Of Interest (ROI)



| Image Path | LIRE Similarity Measure | Image Preview | Image Path | DOGi Similarity Measure | Image Preview |
|---|---|---|---|---|---|
| Dog/Working/Mastiff8 | 0,844 | | Dog/Working/Mastiff8 | 0,899 | |
| Dog/Hound/IrishWolf2 | 0,82 | | Dog/Hound/IrishWolf2 | 0,877 | |
| Dog/Herding/Smooth1 | 0,807 | | Dog/Working/Mastiff5 | 0,859 | |
| Dog/Working/Mastiff6 | 0,793 | | Dog/Working/AlaskanMalamute6 | 0,842 | |
| Dog/Working/AlaskanMalamute6 | 0,776 | | Dog/Working/Mastiff6 | 0,841 | |
| Dog/Herding/Tervuren7 | 0,767 | | Dog/Herding/Tervuren7 | 0,838 | |
| Dog/Terrier/EnglishToy9 | 0,767 | | Dog/Herding/Smooth1 | 0,838 | |
| Dog/Sporting/EnglishSetter7 | 0,766 | | Dog/Working/Mastiff4 | 0,835 | |
| Dog/Working/Mastiff4 | 0,765 | | Dog/Terrier/EnglishToy9 | 0,829 | |

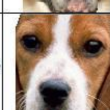
Retrieved Results using LIRE similarity(left) and DOGi Similarity(right) for QUERY2.

QUERY 3 : Original Image and Region Of Interest (ROI)

| Image Path | LIRE Similarity Measure | Image Preview | Image Path | DOGi Similarity Measure | Image Preview |
|---|---|---|---|---|---|
| Dog/Herding/Border6 | 0,782 | | Dog/Herding/Border6 | 0,842 | |
| Dog/Herding/Border1 | 0,777 | | Dog/Herding/Border1 | 0,838 | |
| Dog/Terrier/Kerry8 | 0,764 | | Dog/Terrier/Kerry8 | 0,822 | |
| Dog/Herding/Border9 | 0,753 | | Dog/Herding/Border9 | 0,82 | |
| Dog/Terrier/Kerry1 | 0,753 | | Dog/Herding/Border3 | 0,818 | |
| Dog/Working/SaintBernard1 | 0,787 | | Dog/Herding/Border7 | 0,817 | |
| Dog/Herding/Border7 | 0,748 | | Dog/Terrier/BostonTerrier3 | 0,812 | |
| Dog/Terrier/BostonTerrier3 | 0,747 | | Dog/Herding/Border4 | 0,811 | |
| Dog/Herding/Border4 | 0,745 | | Dog/Working/SaintBernard1 | 0,81 | |

Retrieved Results using LIRE similarity(left) and DOGi Similarity(right) for QUERY3.


QUERY 4 : Original Image and Region Of Interest (ROI)

| Image Path | LIRE Similarity Measure | Image Preview | Image Path | DOGi Similarity Measure | Image Preview |
|---|---|---|---|---|---|
| Dog/Working/SaintBernard6 | 0,825 |  | Dog/Hound/Basset9 | 0,874 |  |
| Dog/Hound/Basset3 | 0,823 |  | Dog/Hound/Basset3 | 0,872 |  |
| Dog/Hound/Basset9 | 0,82 |  | Dog/Working/SaintBernard6 | 0,872 |  |
| Dog/Hound/Beagle2 | 0,811 |  | Dog/Hound/Beagle2 | 0,856 |  |
| Dog/Terrier/Airedale8 | 0,805 |  | Dog/Hound/Basset2 | 0,855 |  |
| Dog/Herding/Rough9 | 0,802 |  | Dog/Hound/Basset4 | 0,851 |  |
| Dog/Hound/Basset2 | 0,799 |  | Dog/Hound/Beagle5 | 0,85 |  |
| Dog/Hound/Beagle5 | 0,796 |  | Dog/Herding/Rough9 | 0,847 |  |
| Dog/Terrier/Stantffordshire1 | 0,786 |  | Dog/Terrier/Airedale8 | 0,845 |  |

Retrieved Results using LIRE similarity(left) and DOGi Similarity(right) for QUERY4.


QUERY 5 : Original Image and Region Of Interest (ROI)

| Image Path | LIRE Similarity Measure | Image Preview | Image Path | DOGi Similarity Measure | Image Preview |
|---|---|---|---|---|---|
| Dog/Sporting/Labrador7 | 0,882 | | Dog/Sporting/Labrador7 | 0,935 | |
| Dog/Toy/Maltese2 | 0,85 | | Dog/Sporting/Labrador3 | 0,905 | |
| Dog/Sporting/Labrador3 | 0,844 | | Dog/Toy/Maltese2 | 0,905 | |
| Dog/Sporting/Cocker9 | 0,838 | | Dog/Sporting/Labrador6 | 0,883 | |
| Dog/Sporting/Labrador9 | 0,832 | | Dog/Sporting/Labrador9 | 0,88 | |
| Dog/Toy/Maltese1 | 0,83 | | Dog/Sporting/Cocker9 | 0,88 | |
| Dog/Working/Akita6 | 0,828 | | Dog/Toy/Maltese1 | 0,879 | |
| Dog/Working/Akita5 | 0,821 | | Dog/Working/Akita6 | 0,877 | |
| Dog/non-Sporting/Dalmatian3 | 0,817 | | Dog/non-Sporting/Dalmatian3 | 0,876 | |

Retrieved Results using LIRE similarity(left) and DOGi Similarity(right) for QUERY5.

## 6.2 Decision Tree Pruning

Pruning is a crucial step in making the decision tree robust and optimized.In Figure 6.4,you can see the precision-recall diagram of DOGi similarity measure using percentages of the original tree and LIRE similarity measure. Our results show that pruning improves the DOGi similarity measure performance because it removes information less important .However,it is important not to overprun the tree as information will be lost.So,pruning the 15%-25% of the tree (in other words ,keeping the 85%-75% of the tree ) produces the best results.In Figure 6.5, you can see the precision recall diagram of DOGi similarity without pruning,DOGi similarity with pruning(keeping the 79% of the tree) and LIRE similarity measure.In this work,we used DOGi similarity measure keeping the 79% of the tree (the green line in Figure 6.5).

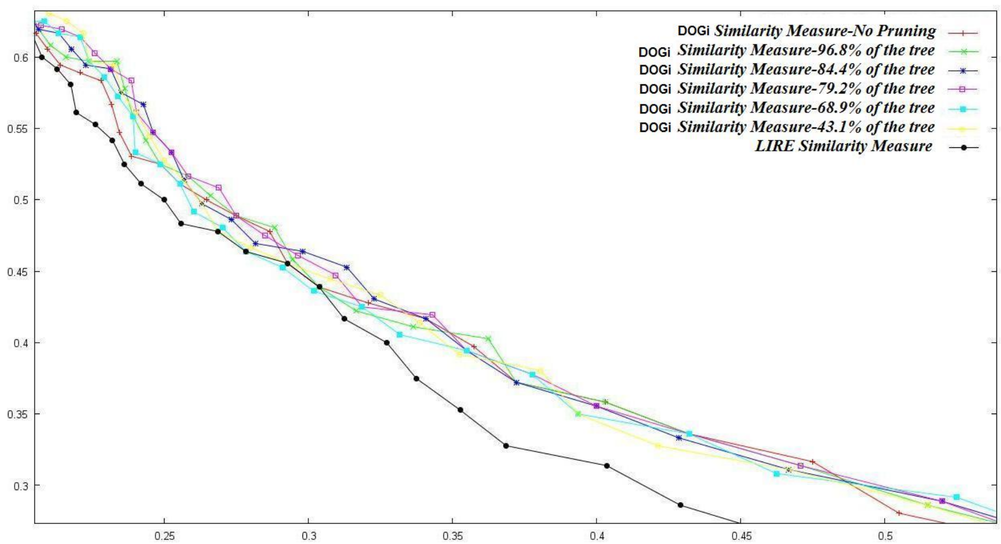



Figure 6.4 : Precision-Recall diagram of weigthed similarity measures with different pruning factors

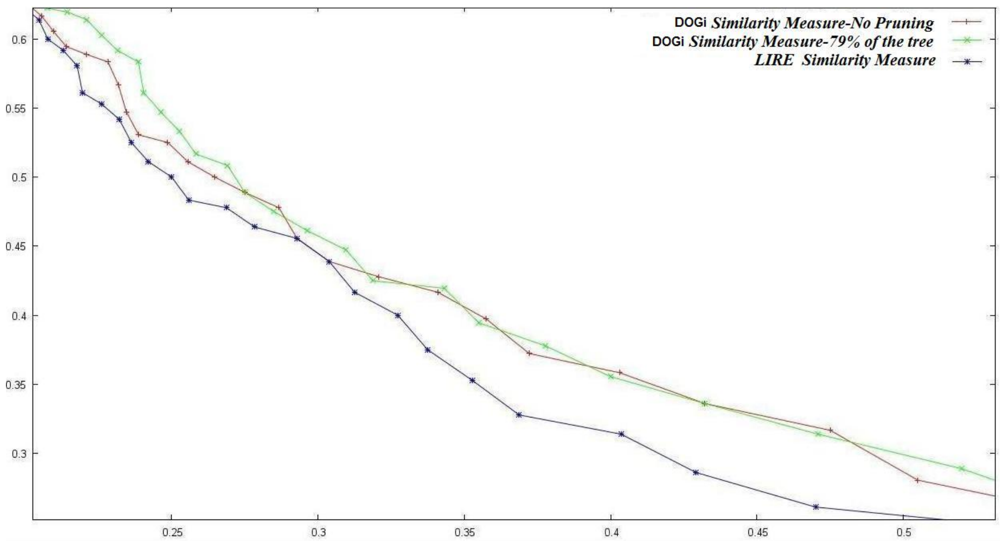



Figure 6.5: Precision Recall diagram of the Lire similarity measure,DOGi similarity measure without pruning and with pruning using 79% of tree

## 6.3 Decision Tree Summary

The next Figures are a summary of our decision tree.Note that we used weka [33]to constuct the decision tree and pruning was done by decreasing the confidencefactor (in our case is equal to 0.25 which means that we have kept the 79% of the tree).In Figure 6.6 you can see a part of the decision tree,in Figure 6.7 the training set used and in Figure 6.8 the statistics of the decision tree .

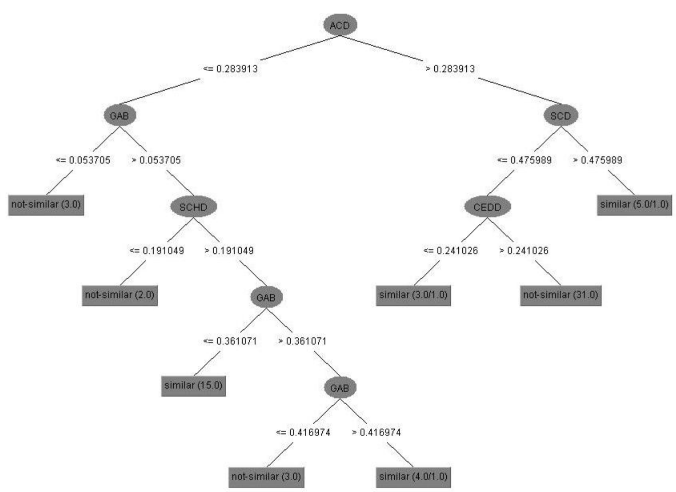



Figure 6.6 : Part of our decision tree

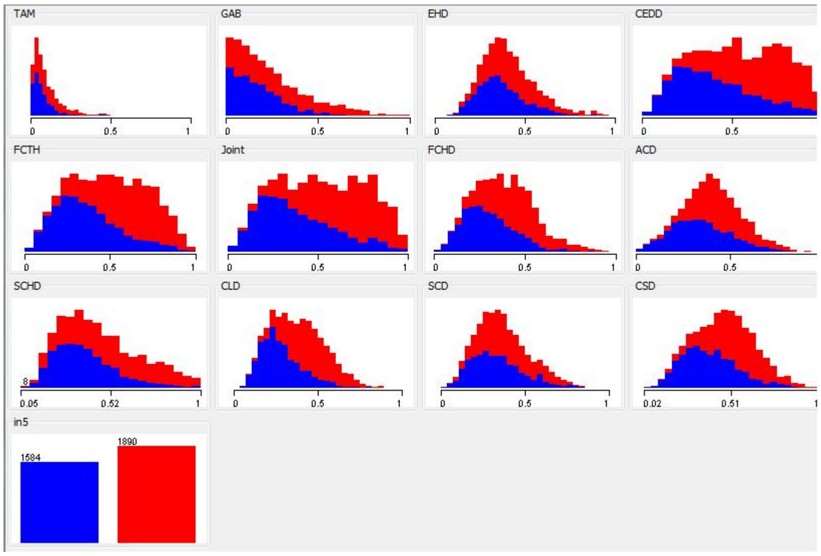



Figure 6.7 : The training set used for classifing pair of images into two categories:similar and not-similar.1584 instances are similar and 1890 are not similar.

```
Number of Leaves  :     82

Size of the tree :      163


Time taken to build model: 0.35 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2692               77.4899 %
Incorrectly Classified Instances       782               22.5101 %
Kappa statistic                          0.5466
Mean absolute error                      0.2824
Root mean squared error                  0.419
Relative absolute error                 56.9175 %
Root relative squared error             84.1272 %
Total Number of Instances             3474

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.757     0.21       0.751     0.757     0.754      0.81     similar
                 0.79      0.243      0.795     0.79      0.792      0.81     not-similar
Weighted Avg.    0.775     0.228      0.775     0.775     0.775      0.81

=== Confusion Matrix ===

    a    b   <-- classified as
 1199  385 |    a = similar
  397 1493 |    b = not-similar
```

Figure 6.8 : statistics of our decision tree

## 6.4 Number Of Categories

The number of categories recognized ,affects the overall performance of our system.In Figure 6.9 we can see the precision-recall diagram of LIRE and DOGi similarity measures for different number of categories recognized.We conclude that when the number of categories is decreased ,the quality of our system is increased.However,decreasing the number of categories recognized ,obviously decreases the ability of our system to recognize categories.Thus, it is a matter of system design on how many categories will be chosen.If we want a high-performing system ,we will choose to have less categories recognized in favor of greater performance while if we want more categories to be recognized ,we will sacrifice performance.

The average case is to choose 20-30 categories as we have a satisfying performance with respectible number of categories.In our work,we have chosen 40 categories to be recognized because we want to have a more robust system.Moreover,in order to test the annotation methods and the similarity measure ,we ought to have a less performing ,more general system.

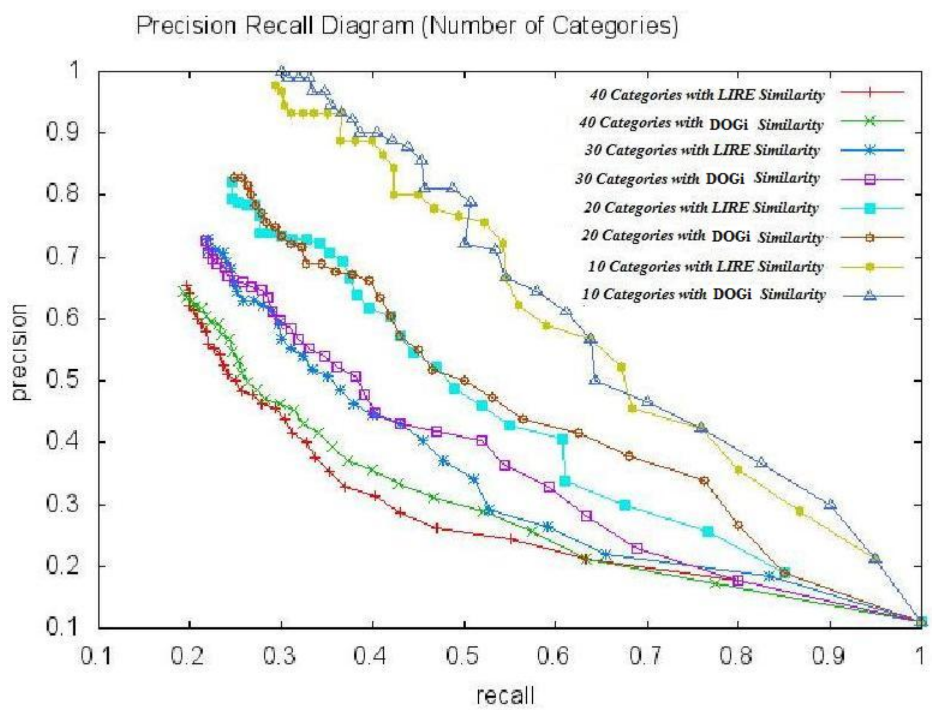



Figure 6.9 : Precion-Recall diagrams of Lire and DOGi similarity measures using different number of categories


Figure 6.10: The example queries (denoting each corresponding semantic category) original image and ROI

## 6.5 Comparison Of Semantic Category Estimation Methods

To properly annotate an unknown image, its semantic category should be estimated. In this work the semantic category is estimated based on further analysis of the retrieval results. For each unknown image, a ranked list of images is produced in decreasing order of similarity. Each of the known classes contain 9 instances in the ranked list adopted. As we previously mentioned ,four methods using the two similarity measures available have been tested for the semantic category estimation based on retrieval results, summation of 9 instances,max occurrence, AVR and Best match.Also, a fifth method(decision tree method) that is independent to the similarity measure ,has been tested . In the table below you can see the results for each method.These results were taken using over 40 queries (Figure 6.2).

With LIRE similarity measure the semantic category estimation methods performed as follows:

Best Match, 48.5% of the queries belong to the same class as the first ranked image, 10% of the queries belong to the same class as the second best image in the ranked list, and 10% of queries belong to the same class with the third best image in the ranked list.

Max occurrence , 55% of the queries belong to the class that has the maximum number of instances in first 20 answers, 15% of the queries belong to the class that has the second maximum number of instances in first 20 answers, and 15% of the queries belong to the class that has the third maximum number of instances in first 20 answers.

AVR, 57.5% of the queries belong to the class with the best AVR , 17.5% of the queries belong to the class with the second best AVR, and 12% of the queries belong to the class with the third best AVR.

summation of 9 instances, 60% of the queries belong to the class with the highest summation , 19% of the queries belong to the class with the second highest summation, and 10% of the queries belong to the class with the third highest summation.

With DOGi similarity measure the semantic category estimation methods performed as follows:

Best Match, 50% of the queries belong to the same class as the first ranked image, 10% of the queries belong to the same class as the second best image in the ranked list, and 10% of queries belong to the same class with the third best image in the ranked list.

Max occurrence , 65% of the queries belong to the class that has the maximum number of instances in first 20 answers, 15% of the queries belong to the class that has the second maximum number of instances in first 20 answers, and 10% of the queries belong to the class that has the third maximum number of instances in first 20 answers.

AVR, 62.5% of the queries belong to the class with the best AVR , 22.5% of the queries belong to the class with the second best AVR, and 10% of the queries belong to the class with the third best AVR.

summation of 9 instances, 72.5% of the queries belong to the class with the highest summation , 17.5% of the queries belong to the class with the second highest summation, and 5% of the queries belong to the class with the third highest summation.

Finally, with decision tree method 52% of the queries belong to the same class as the first ranked image, 18% of the queries belong to the same class as the second best image in the ranked list, and 12.5% of queries belong to the same class with the third best image in the ranked list.

Obviously, semantic category estimation methods using DOGi similarity measure are superior.summation of 9 instances is proved to be the most reliable method while best match method is the most unreliable.AVR and max occurence both have an average performance and decision tree method can be usefull when other methods(AVR ,summation of 9...) fail to classify correctly the unknown image. For the five example queries(Figure 6.10),we present the results for each semantic category estimation method(annotation method)

| Semantic Category Estimation Methods | 1st Answer Correct | 2nd Answer Correct | 3rd Answer Correct | Overall 1-3 Answers Correct |
|---|---|---|---|---|
| Summary Of 9 Instances( **DOGi** Similarity Measure) | 72.5% | 17.5% | 5% | 95% |
| AVR( **DOGi** Similarity Measure) | 62.5% | 22.5% | 10% | 92.5% |
| Max Occurence( **DOGi** Similarity Measure) | 65% | 15% | 10% | 90% |
| Summary Of 9 Instances (LIRE Similarity Measure) | 60% | 19% | 10% | 89% |
| AVR(LIRE Similarity Measure) | 57.5% | 17.5% | 12% | 87% |
| Max Occurence(LIRE Similarity Measure) | 55% | 15% | 15% | 85% |
| Decision Tree Method | 52% | 18% | 12.5% | 82.5% |
| Best Match( **DOGi** Similarity Measure) | 50% | 10% | 10% | 70% |
| Best Match (LIRE Similarity Measure) | 48.5% | 10% | 10% | 68.5% |

Figure 6.11 : Percentages of first ,second,third correct answers for every semantic category estimation method

***AVR Method***

| Image | Category | Similarity (Min=4 , Max=401) | Percentage % |
|---|---|---|---|
| | Siberian_Husky | 42.666666666666664 | 90.33333333333333 % |
| | Maltese | 51.888888888888886 | 88.02777777777777 % |
| | Dalmatian | 54.666666666666664 | 87.33333333333333 % |
| | Alaskan_Malamute | 55.77777777777778 | 87.05555555555556 % |
| | Akita | 68.22222222222223 | 83.94444444444444 % |
| | English_Setter | 80.22222222222223 | 80.94444444444444 % |

***Summary Of 9 Instances Method***

| Image | Category | Score Summary Of 9 Instances | Percentage % |
|---|---|---|---|
| | Siberian_Husky | 6.5681077985486045 | 72.97897553942893 % |
| | Maltese | 6.443120492181131 | 71.59022769090146 % |
| | Alaskan_Malamute | 6.426886178777313 | 71.40984643085902 % |
| | Dalmatian | 6.328974547383681 | 70.32193941537423 % |
| | Akita | 6.211502604892058 | 69.01669560991175 % |
| | Poodle | 6.087055264058439 | 67.63394737842711 % |

***Decision Tree Method***

| Image | Category | Decision Tree Method | Percentage % |
|---|---|---|---|
| | Siberian_Husky | 80.8 | 80.8 % |
| | Dalmatian | 67.67 | 67.67 % |
| | Mastiff | 67.67 | 67.67 % |
| | Alaskan_Malamute | 62.62 | 62.62 % |
| | Poodle | 60.6 | 60.6 % |
| | Akita | 57.57 | 57.57 % |

***Max Occurence Method***

| Image | Category | Max Occurence (min 1,max 9) | Percentage % |
|---|---|---|---|
| | Siberian_Husky | 4 | 44.44444444444444 % |
| | Maltese | 4 | 44.44444444444444 % |
| | Alaskan_Malamute | 4 | 44.44444444444444 % |
| | Pomeranian | 3 | 33.33333333333333 % |
| | Collie_Smouth | 2 | 22.22222222222222 % |
| | Poodle | 2 | 22.22222222222222 % |

Annotation methods results for QUERY1 using LIRE similarity measure

***AVR Method***

| Image | Category | Similarity (Min=4 , Max=401) | Percentage % |
|---|---|---|---|
|  | Siberian_Husky | 42.666666666666664 | 90.33333333333333 % |
|  | Maltese | 51.888888888888886 | 88.02777777777777 % |
|  | Dalmatian | 54.666666666666664 | 87.33333333333333 % |
|  | Alaskan_Malamute | 55.77777777777778 | 87.05555555555556 % |
|  | Akita | 68.22222222222223 | 83.94444444444444 % |
|  | English_Setter | 80.22222222222223 | 80.94444444444444 % |

***Summary Of 9 Instances Method***

| Image | Category | Score Summary Of 9 Instances | Percentage % |
|---|---|---|---|
|  | Siberian_Husky | 7.2295157648471635 | 93.39554648899865 % |
|  | Alaskan_Malamute | 7.005705264021495 | 90.50421811867707 % |
|  | Dalmatian | 6.978852880114895 | 90.15732170803726 % |
|  | Maltese | 6.968855473957266 | 90.02816876862404 % |
|  | Akita | 6.765388433478687 | 87.39965033722427 % |
|  | Poodle | 6.695859605890282 | 86.50143210787434 % |

***Decision Tree Method***

| Image | Category | Decision Tree Method | Percentage % |
|---|---|---|---|
|  | Siberian_Husky | 80.8 | 80.8 % |
|  | Dalmatian | 67.67 | 67.67 % |
|  | Mastiff | 67.67 | 67.67 % |
|  | Alaskan_Malamute | 62.62 | 62.62 % |
|  | Poodle | 60.6 | 60.6 % |
|  | Akita | 57.57 | 57.57 % |

***Max Occurence Method***

| Image | Category | Max Occurence (min 1,max 9) | Percentage % |
|---|---|---|---|
|  | Siberian_Husky | 4 | 44.44444444444444 % |
|  | Maltese | 4 | 44.44444444444444 % |
|  | Alaskan_Malamute | 4 | 44.44444444444444 % |
|  | Pomeranian | 3 | 33.33333333333333 % |
|  | Collie_Smouth | 2 | 22.22222222222222 % |
|  | Poodle | 2 | 22.22222222222222 % |

Annotation methods results for QUERY1 using DOGi similarity measure

***AVR Method***

| Image | Category | Similarity (Min=4 , Max=401) | Percentage % |
|---|---|---|---|
| | English_Setter | 60.44444444444444 | 85.88888888888889 % |
| | Mastiff | 72.0 | 83.0 % |
| | Alaskan_Malamute | 72.66666666666667 | 82.83333333333333 % |
| | Dalmatian | 78.0 | 81.5 % |
| | Akita | 88.88888888888889 | 78.77777777777777 % |
| | Siberian_Husky | 89.77777777777777 | 78.55555555555556 % |

***Summary Of 9 Instances Method***

| Image | Category | Score Summary Of 9 Instances | Percentage % |
|---|---|---|---|
| | Mastiff | 6.302960695106927 | 70.03289661229918 % |
| | English_Setter | 6.211339442021339 | 69.01488268912598 % |
| | Alaskan_Malamute | 6.1274348627707065 | 68.08260958634118 % |
| | Irish_Wolf_Hound | 6.0547147233359375 | 67.27460803706597 % |
| | Dalmatian | 6.011382530222534 | 66.79313922469483 % |
| | Siberian_Husky | 5.942506791800286 | 66.0278532422254 % |

***Decision Tree Method***

| Image | Category | Decision Tree Method | Percentage % |
|---|---|---|---|
| | English_Setter | 85.85 | 85.85 % |
| | Dalmatian | 77.77 | 77.77 % |
| | Mastiff | 74.74 | 74.74 % |
| | Irish_Wolf_Hound | 72.72 | 72.72 % |
| | Alaskan_Malamute | 67.67 | 67.67 % |
| | Siberian_Husky | 55.55 | 55.55 % |

***Max Occurence Method***

| Image | Category | Max Occurence (min 1,max 9) | Percentage % |
|---|---|---|---|
| | Mastiff | 5 | 55.55555555555556 % |
| | Irish_Wolf_Hound | 4 | 44.44444444444444 % |
| | Alaskan_Malamute | 4 | 44.44444444444444 % |
| | Maltese | 3 | 33.33333333333333 % |
| | English_Setter | 3 | 33.33333333333333 % |
| | Tervueren | 2 | 22.22222222222222 % |

Annotation methods results for QUERY2 using LIRE similarity measure

***AVR Method***

| Image | Category | Similarity (Min=4 , Max=401) | Percentage % |
|---|---|---|---|
|  | English_Setter | 62.666666666666664 | 85.33333333333333 % |
|  | Alaskan_Malamute | 72.33333333333333 | 82.91666666666667 % |
|  | Dalmatian | 72.55555555555556 | 82.86111111111111 % |
|  | Mastiff | 74.44444444444444 | 82.38888888888889 % |
|  | Siberian_Husky | 88.66666666666667 | 78.83333333333333 % |
|  | Irish_Wolf_Hound | 91.22222222222223 | 78.19444444444444 % |

***Summary Of 9 Instances Method***

| Image | Category | Score Summary Of 9 Instances | Percentage % |
|---|---|---|---|
|  | Mastiff | 6.8797958233723735 | 88.8776387736637 % |
|  | English_Setter | 6.825996380131257 | 88.18262287415634 % |
|  | Alaskan_Malamute | 6.771568479494911 | 87.47948815676659 % |
|  | Dalmatian | 6.72522084418653 | 86.8807395173146 % |
|  | Irish_Wolf_Hound | 6.6748256279904234 | 86.22970161793656 % |
|  | Siberian_Husky | 6.6091152605164725 | 85.38081271861864 % |

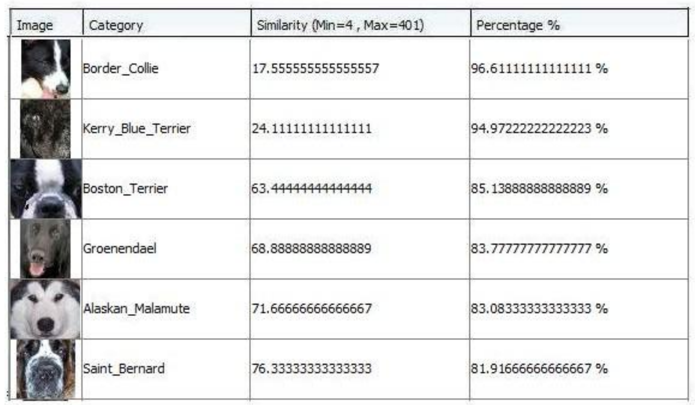
***Decision Tree Method***

| Image | Category | Decision Tree Method | Percentage % |
|---|---|---|---|
|  | English_Setter | 85.85 | 85.85 % |
|  | Dalmatian | 77.77 | 77.77 % |
|  | Mastiff | 74.74 | 74.74 % |
|  | Irish_Wolf_Hound | 72.72 | 72.72 % |
|  | Alaskan_Malamute | 67.67 | 67.67 % |
|  | Siberian_Husky | 55.55 | 55.55 % |

***Max Occurence Method***

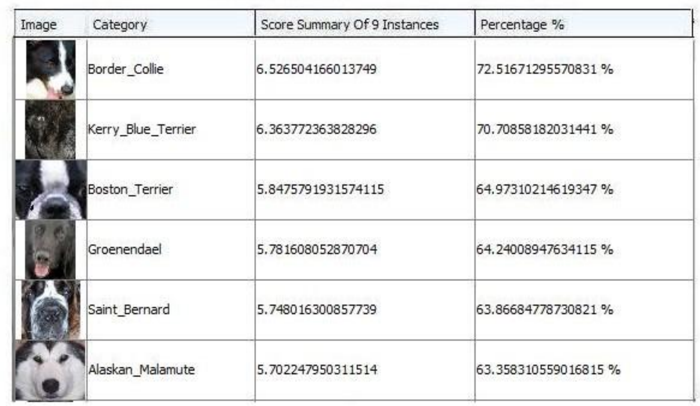
| Image | Category | Max Occurence (min 1,max 9) | Percentage % |
|---|---|---|---|
|  | Mastiff | 5 | 55.5555555555556 % |
|  | Irish_Wolf_Hound | 4 | 44.44444444444444 % |
|  | Alaskan_Malamute | 4 | 44.44444444444444 % |
|  | Maltese | 3 | 33.33333333333333 % |
|  | English_Setter | 3 | 33.33333333333333 % |
|  | Tervueren | 2 | 22.22222222222222 % |

Annotation methods results for QUERY1 using DOGi similarity measure



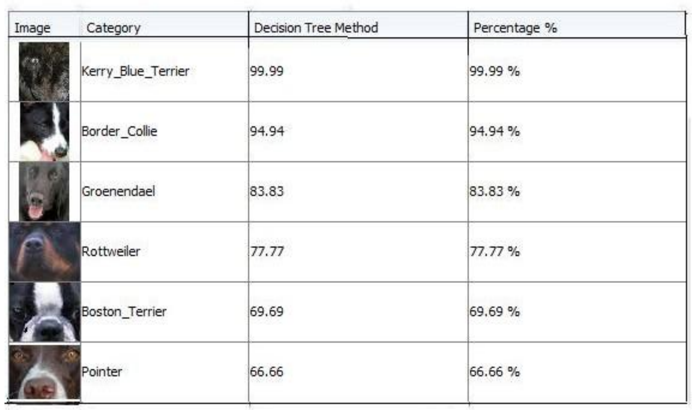
**AVR Method**

| Image | Category | Similarity (Min=4 , Max=401) | Percentage % |
|---|---|---|---|
| | Border_Collie | 17.555555555555557 | 96.61111111111111 % |
| | Kerry_Blue_Terrier | 24.11111111111111 | 94.97222222222223 % |
| | Boston_Terrier | 63.44444444444444 | 85.13888888888889 % |
| | Groenendael | 68.88888888888889 | 83.77777777777777 % |
| | Alaskan_Malamute | 71.66666666666667 | 83.08333333333333 % |
| | Saint_Bernard | 76.33333333333333 | 81.91666666666667 % |



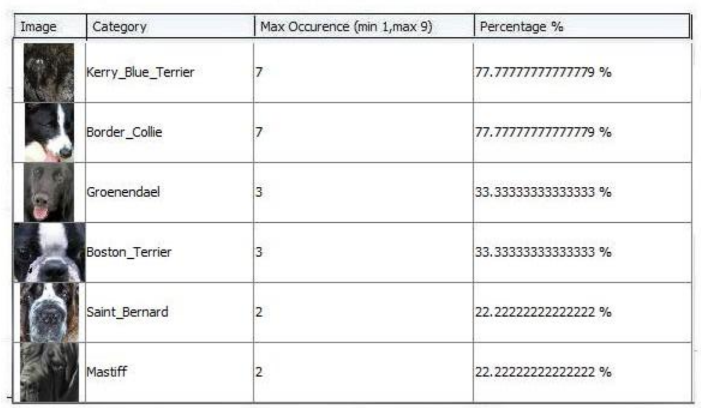
**Summary Of 9 Instances Method**

| Image | Category | Score Summary Of 9 Instances | Percentage % |
|---|---|---|---|
| | Border_Collie | 6.526504166013749 | 72.51671295570831 % |
| | Kerry_Blue_Terrier | 6.363772363828296 | 70.70858182031441 % |
| | Boston_Terrier | 5.8475791931574115 | 64.97310214619347 % |
| | Groenendael | 5.781608052870704 | 64.24008947634115 % |
| | Saint_Bernard | 5.748016300857739 | 63.86684778730821 % |
| | Alaskan_Malamute | 5.702247950311514 | 63.358310559016815 % |



**Decision Tree Method**

| Image | Category | Decision Tree Method | Percentage % |
|---|---|---|---|
| | Kerry_Blue_Terrier | 99.99 | 99.99 % |
| | Border_Collie | 94.94 | 94.94 % |
| | Groenendael | 83.83 | 83.83 % |
| | Rottweiler | 77.77 | 77.77 % |
| | Boston_Terrier | 69.69 | 69.69 % |
| | Pointer | 66.66 | 66.66 % |



**Max Occurence Method**

| Image | Category | Max Occurence (min 1,max 9) | Percentage % |
|---|---|---|---|
| | Kerry_Blue_Terrier | 7 | 77.77777777777779 % |
| | Border_Collie | 7 | 77.77777777777779 % |
| | Groenendael | 3 | 33.33333333333333 % |
| | Boston_Terrier | 3 | 33.33333333333333 % |
| | Saint_Bernard | 2 | 22.22222222222222 % |
| | Mastiff | 2 | 22.22222222222222 % |

Annotation methods results for QUERY3 using LIRE similarity measure

***AVR Method***

| Image | Category | Similarity (Min=4 , Max=401) | Percentage % |
|---|---|---|---|
|  | Border_Collie | 18.88888888888889 | 96.27777777777777 % |
|  | Kerry_Blue_Terrier | 25.666666666666668 | 94.58333333333333 % |
|  | Boston_Terrier | 64.77777777777777 | 84.80555555555556 % |
|  | Groenendael | 65.66666666666667 | 84.58333333333333 % |
|  | Saint_Bernard | 75.55555555555556 | 82.11111111111111 % |
|  | Tervueren | 77.22222222222223 | 81.69444444444444 % |

***Summary Of 9 Instances Method***

| Image | Category | Score Summary Of 9 Instances | Percentage % |
|---|---|---|---|
|  | Border_Collie | 7.137569695754891 | 92.20772787849873 % |
|  | Kerry_Blue_Terrier | 7.012609254695363 | 90.5934083221311 % |
|  | Boston_Terrier | 6.552242132155983 | 84.64608897272205 % |
|  | Groenendael | 6.525289066367888 | 84.29789188861399 % |
|  | Saint_Bernard | 6.458834005614285 | 83.43938256130588 % |
|  | Tervueren | 6.382275808197715 | 82.45035439973795 % |

***Decision Tree Method***

| Image | Category | Decision Tree Method | Percentage % |
|---|---|---|---|
|  | Kerry_Blue_Terrier | 99.99 | 99.99 % |
|  | Border_Collie | 94.94 | 94.94 % |
|  | Groenendael | 83.83 | 83.83 % |
|  | Rottweiler | 77.77 | 77.77 % |
|  | Boston_Terrier | 69.69 | 69.69 % |
|  | Pointer | 66.66 | 66.66 % |

***Max Occurence Method***

| Image | Category | Max Occurence (min 1,max 9) | Percentage % |
|---|---|---|---|
|  | Kerry_Blue_Terrier | 7 | 77.77777777777779 % |
|  | Border_Collie | 7 | 77.77777777777779 % |
|  | Groenendael | 3 | 33.33333333333333 % |
|  | Boston_Terrier | 3 | 33.33333333333333 % |
|  | Saint_Bernard | 2 | 22.22222222222222 % |
|  | Mastiff | 2 | 22.22222222222222 % |

Annotation methods results for QUERY3 using DOGi similarity measure

***AVR Method***

| Image | Category | Similarity (Min=4 , Max=401) | Percentage % |
|---|---|---|---|
|  | Basset_Hound | 13.555555555555555 | 97.61111111111111 % |
|  | Collie_Rough | 33.55555555555556 | 92.61111111111111 % |
|  | Beagle | 56.55555555555556 | 86.86111111111111 % |
|  | Airedale_Terrier | 71.55555555555556 | 83.11111111111111 % |
|  | German_Shepherd | 83.22222222222223 | 80.19444444444444 % |
|  | American_Cocker_Spaniel | 83.77777777777777 | 80.05555555555556 % |

***Summary Of 9 Instances Method***

| Image | Category | Score Summary Of 9 Instances | Percentage % |
|---|---|---|---|
|  | Basset_Hound | 7.099253128034631 | 78.88059031149591 % |
|  | Collie_Rough | 6.871488687716996 | 76.34987430796662 % |
|  | Beagle | 6.743573510969001 | 74.92859456632223 % |
|  | Airedale_Terrier | 6.621324518723088 | 73.57027243025654 % |
|  | American_Cocker_Spaniel | 6.510108980291448 | 72.33454422546053 % |
|  | German_Shepherd | 6.491799378592235 | 72.13110420658039 % |

***Decision Tree Method***

| Image | Category | Decision Tree Method | Percentage % |
|---|---|---|---|
|  | Basset_Hound | 94.94 | 94.94 % |
|  | Collie_Rough | 85.85 | 85.85 % |
|  | Beagle | 79.79 | 79.79 % |
|  | Airedale_Terrier | 78.78 | 78.78 % |
|  | German_Shepherd | 68.68 | 68.68 % |
|  | Blood_Hound | 67.67 | 67.67 % |

***Max Occurence Method***

| Image | Category | Max Occurence (min 1,max 9) | Percentage % |
|---|---|---|---|
|  | Basset_Hound | 8 | 88.88888888888889 % |
|  | Beagle | 4 | 44.44444444444444 % |
|  | Sussex_Spaniel | 3 | 33.33333333333333 % |
|  | Collie_Rough | 3 | 33.33333333333333 % |
|  | American_Cocker_Spaniel | 3 | 33.33333333333333 % |
|  | American_Staffordshire_T... | 2 | 22.22222222222222 % |

Annotation methods results for QUERY4 using LIRE similarity measure

***AVR Method***

| Image | Category | Similarity (Min=4 , Max=401) | Percentage % |
|---|---|---|---|
|  | Basset_Hound | 16.77777777777778 | 96.80555555555556 % |
|  | Collie_Rough | 32.888888888888886 | 92.77777777777777 % |
|  | Beagle | 60.22222222222222 | 85.94444444444444 % |
|  | Airedale_Terrier | 68.44444444444444 | 83.88888888888889 % |
|  | German_Shepherd | 84.66666666666667 | 79.83333333333333 % |
|  | Blood_Hound | 84.66666666666667 | 79.83333333333333 % |

***Summary Of 9 Instances Method***

| Image | Category | Score Summary Of 9 Instances | Percentage % |
|---|---|---|---|
|  | Basset_Hound | 7.5573652218540595 | 97.63091718314193 % |
|  | Collie_Rough | 7.367910745021845 | 95.18342208470555 % |
|  | Beagle | 7.197730478799741 | 92.98492366759992 % |
|  | Airedale_Terrier | 7.142213669260519 | 92.26772172283718 % |
|  | German_Shepherd | 7.007336754329448 | 90.52529476251588 % |
|  | Blood_Hound | 7.0007058189513325 | 90.43963206344776 % |

***Decision Tree Method***

| Image | Category | Decision Tree Method | Percentage % |
|---|---|---|---|
|  | Basset_Hound | 94.94 | 94.94 % |
|  | Collie_Rough | 85.85 | 85.85 % |
|  | Beagle | 79.79 | 79.79 % |
|  | Airedale_Terrier | 78.78 | 78.78 % |
|  | German_Shepherd | 68.68 | 68.68 % |
|  | Blood_Hound | 67.67 | 67.67 % |

***Max Occurence Method***

| Image | Category | Max Occurence (min 1,max 9) | Percentage % |
|---|---|---|---|
|  | Basset_Hound | 8 | 88.88888888888889 % |
|  | Beagle | 4 | 44.44444444444444 % |
|  | Sussex_Spaniel | 3 | 33.33333333333333 % |
|  | Collie_Rough | 3 | 33.33333333333333 % |
|  | American_Cocker_Spaniel | 3 | 33.33333333333333 % |
|  | American_Staffordshire_T... | 2 | 22.22222222222222 % |

Annotation methods results for QUERY4 using DOGi similarity measure

***AVR Method***

| Image | Category | Similarity (Min=4 , Max=401) | Percentage % |
|---|---|---|---|
|  | Labrador_Retriever | 25.555555555555557 | 94.61111111111111 % |
|  | Maltese | 54.44444444444444 | 87.38888888888889 % |
|  | Akita | 74.33333333333333 | 82.41666666666667 % |
|  | Poodle | 77.77777777777777 | 81.55555555555556 % |
|  | Dalmatian | 86.66666666666667 | 79.33333333333333 % |
|  | Bulldog | 101.0 | 75.75 % |

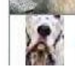
***Summary Of 9 Instances Method***

| Image | Category | Score Summary Of 9 Instances | Percentage % |
|---|---|---|---|
|  | Labrador_Retriever | 7.18400382941892 | 79.82226477132133 % |
|  | Maltese | 6.863632470629514 | 76.2625830069946 % |
|  | Akita | 6.6719492524248425 | 74.13276947138714 % |
|  | Poodle | 6.582408973032157 | 73.13787747813508 % |
|  | Dalmatian | 6.563178804421662 | 72.92420893801848 % |
|  | Collie_Smouth | 6.363684891902928 | 70.70760991003253 % |

***Decision Tree Method***

| Image | Category | Decision Tree Method | Percentage % |
|---|---|---|---|
|  | Labrador_Retriever | 84.84 | 84.84 % |
|  | Akita | 83.83 | 83.83 % |
|  | Poodle | 77.77 | 77.77 % |
|  | Maltese | 77.77 | 77.77 % |
|  | Dalmatian | 76.76 | 76.76 % |
|  | Beagle | 73.73 | 73.73 % |

***Max Occurence Method***

| Image | Category | Max Occurence (min 1,max 9) | Percentage % |
|---|---|---|---|
|  | Labrador_Retriever | 7 | 77.77777777777779 % |
|  | Maltese | 5 | 55.55555555555556 % |
|  | Akita | 4 | 44.44444444444444 % |
|  | Collie_Smouth | 3 | 33.33333333333333 % |
|  | Poodle | 2 | 22.22222222222222 % |
|  | English_Setter | 2 | 22.22222222222222 % |

Annotation methods results for QUERY5 using LIRE similarity measure

*AVR Method*

| Image | Category | Similarity (Min=4 , Max=401) | Percentage % |
|---|---|---|---|
|  | Labrador_Retriever | 26.88888888888889 | 94.27777777777777 % |
|  | Maltese | 49.77777777777778 | 88.55555555555556 % |
|  | Poodle | 71.88888888888889 | 83.02777777777777 % |
|  | Dalmatian | 78.11111111111111 | 81.47222222222223 % |
|  | Akita | 81.0 | 80.75 % |
|  | Bulldog | 96.22222222222223 | 76.94444444444444 % |

*Summary Of 9 Instances Method*

| Image | Category | Score Summary Of 9 Instances | Percentage % |
|---|---|---|---|
|  | Labrador_Retriever | 7.711347406264478 | 99.62015833432778 % |
|  | Maltese | 7.430885834343041 | 95.99697489704539 % |
|  | Dalmatian | 7.195048671194094 | 92.95027834762902 % |
|  | Poodle | 7.193824043863007 | 92.93445782208451 % |
|  | Akita | 7.185071804641716 | 92.82139075208107 % |
|  | Bulldog | 6.96482832199638 | 89.97614342274818 % |

*Decision Tree Method*

| Image | Category | Decision Tree Method | Percentage % |
|---|---|---|---|
|  | Labrador_Retriever | 84.84 | 84.84 % |
|  | Akita | 83.83 | 83.83 % |
|  | Poodle | 77.77 | 77.77 % |
|  | Maltese | 77.77 | 77.77 % |
|  | Dalmatian | 76.76 | 76.76 % |
|  | Beagle | 73.73 | 73.73 % |

*Max Occurence Method*

| Image | Category | Max Occurence (min 1,max 9) | Percentage % |
|---|---|---|---|
|  | Labrador_Retriever | 7 | 77.77777777777779 % |
|  | Maltese | 5 | 55.55555555555556 % |
|  | Akita | 4 | 44.44444444444444 % |
|  | Collie_Smouth | 3 | 33.33333333333333 % |
|  | Poodle | 2 | 22.22222222222222 % |
|  | English_Setter | 2 | 22.22222222222222 % |

Annotation methods results for QUERY5 using DOGi similarity measure

## 6.6 MPEG7 Annotations

We present the actual mpeg7 annotations generated for the five example queries:


```
<?xml version="1.0" encoding="UTF-8"?>
<Mpeg7 xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001 mpeg7-2001.xsd" xmlns="urn:mpeg:mpeg7:schema:2001"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><Description
xsi:type="SemanticDescriptionType"><Semantics id="Dogs"><AbstractionLevel dimension="1"/><Label><Name>Dog
Ontology</Name></Label><SemanticBase xsi:type="AgentObjectType" id="Dog"><Relation type="generalizes" source=
"Dog" target="Dog_Breed"/><Relation type="specializes" source="Dog_Breed" target="Dog"/></SemanticBase><
SemanticBase xsi:type="AgentObjectType" id="Siberian_Husky"><AbstractionLevel dimension="1"/><Label><Name>
High Level Features</Name></Label><Property><Name>hasDescription</Name><Definition>["The Siberian Husky is a
medium-size, dense-coat working dog breed that originated in eastern Siberia. The breed belongs to the Spitz genetic family.
It is recognizable by its thickly furred double coat, sickle tail, erect triangular ears, and distinctive markings\nHuskies are an
active, energetic, and resilient breed whose ancestors came from the extremely cold and harsh environment of the Siberian
Arctic. Siberian Huskies were bred by the Chukchi of Northeastern Asia to pull heavy loads long distances through difficult
conditions. The dogs were imported into Alaska during the Nome Gold Rush and later spread into the United States and
Canada. They were initially sent to Alaska and Canada as sled dogs but rapidly acquired the status of family pets and show
dogs."@]</Definition></Property><Property><Name>hasWordNet</Name><Definition>["breed of sled dog developed in
northeastern Siberia; they resemble the larger Alaskan malamutes"@]</Definition></Property><Property><Name>hasSize
</Name><Definition>Large</Definition></Property><Property><Name>hasCountryOfOrigin</Name><Definition>Siberia
</Definition></Property><Property><Name>hasFur</Name><Definition>Short/Medium</Definition></Property><Property
><Name>hasHabitat</Name><Definition>OutDoor</Definition></Property><Property><Name>hasCoatPattern</Name><
Definition>Bi-Color</Definition></Property><Property><Name>hasMainColor</Name><Definition>Light-Grey</
Definition></Property><Property><Name>hasSecondaryColor</Name><Definition>White</Definition></Property><
Relation type="generalizes" source="Dog_Breed" target="Siberian_Husky"/><Relation type="specializes" source=
"Siberian_Husky" target="Dog_Breed"/></SemanticBase></Semantics></Description></Mpeg7>
```


Mpeg7 annotation for QUERY1 : Siberian Husky

```
<?xml version="1.0" encoding="UTF-8"?>
<Mpeg7 xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001 mpeg7-2001.xsd" xmlns="urn:mpeg:mpeg7:schema:2001"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><Description
xsi:type="SemanticDescriptionType"><Semantics id="Dogs"><AbstractionLevel dimension="1"/><Label><Name>Dog
Ontology</Name></Label><SemanticBase xsi:type="AgentObjectType" id="Dog"><Relation type="generalizes" source=
"Dog" target="Dog_Breed"/><Relation type="specializes" source="Dog_Breed" target="Dog"/></SemanticBase><
SemanticBase xsi:type="AgentObjectType" id="Mastiff"><AbstractionLevel dimension="1"/><Label><Name>High
Level Features</Name></Label><Property><Name>hasDescription</Name><Definition>["The Neapolitan Mastiff, Italian
Mastiff, Mastino or Mastini (plural) is a large, ancient dog breed. This massive breed is often used as a guard and defender
of family and property due to their protective instincts and their fearsome appearance. The breed is reported to have been
used to fight alongside the Roman Legions, by having bladed and spiked leather harnesses tied to their backs and being
trained to run under the bellies of enemy horses, to disembowel them"@]</Definition></Property><Property><Name>
hasWordNet</Name><Definition>["an old breed of powerful deep-chested smooth-coated dog used chiefly as a watchdog
and guard dog"@]</Definition></Property><Property><Name>hasSize</Name><Definition>Large</Definition></Property
><Property><Name>hasCountryOfOrigin</Name><Definition>Italy</Definition></Property><Property><Name>hasFur</
Name><Definition>Short/Smouth</Definition></Property><Property><Name>hasHabitat</Name><Definition>OutDoor</
Definition></Property><Property><Name>hasCoatPattern</Name><Definition>Single-Color</Definition></Property><
Property><Name>hasMainColor</Name><Definition>Grey</Definition></Property><Property><Name>
hasSecondaryColor</Name><Definition xsi:nil="true"/></Property><Relation type="generalizes" source="Dog_Breed"
target="Mastiff"/><Relation type="specializes" source="Mastiff" target="Dog_Breed"/></SemanticBase></Semantics>
</Description></Mpeg7>
```


Mpeg7 annotation for QUERY2 : Mastiff


```
<?xml version="1.0" encoding="UTF-8"?>
<Mpeg7 xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001 mpeg7-2001.xsd" xmlns="urn:mpeg:mpeg7:schema:2001"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><Description
xsi:type="SemanticDescriptionType"><Semantics id="Dogs"><AbstractionLevel dimension="1"/><Label><Name>Dog
Ontology</Name></Label><SemanticBase xsi:type="AgentObjectType" id="Dog"><Relation type="generalizes" source=
"Dog" target="Dog_Breed"/><Relation type="specializes" source="Dog_Breed" target="Dog"/></SemanticBase><
SemanticBase xsi:type="AgentObjectType" id="Border_Collie"><AbstractionLevel dimension="1"/><Label><Name>
High Level Features</Name></Label><Property><Name>hasDescription</Name><Definition>["The Border Collie is a
dog breed developed in the Anglo-Scottish border region for use on farms to assist with the herding of livestock. Their
intelligence has been observed as having an intuitive quality that goes well beyond basic instinct. Such sensitivity calls for an
environment that engages their higher faculties; otherwise, they can become distressed. With this accounted for, they are
excellent companion animals.Typically extremely energetic, acrobatic and athletic, they frequently compete with great
success in dog sports, in addition to their success in sheepdog trials, and are often cited as the most intelligent of all
dogs."@]</Definition></Property><Property><Name>hasWordNet</Name><Definition>["(developed in the area between
Scotland and England usually having a black coat with white on the head and tip of tail used for herding both sheep and
cattle)"@]</Definition></Property><Property><Name>hasSize</Name><Definition>Medium</Definition></Property><
Property><Name>hasCountryOfOrigin</Name><Definition>Scotland</Definition></Property><Property><Name>hasFur</
Name><Definition>Medium/Long</Definition></Property><Property><Name>hasHabitat</Name><Definition>
InDoor/OutDoor</Definition></Property><Property><Name>hasCoatPattern</Name><Definition>Bi-Color</Definition></
Property><Property><Name>hasMainColor</Name><Definition>Black</Definition></Property><Property><Name>
hasSecondaryColor</Name><Definition>White</Definition></Property><Relation type="generalizes" source=
"Dog_Breed" target="Border_Collie"/><Relation type="specializes" source="Border_Collie" target="Dog_Breed"/></
SemanticBase></Semantics></Description></Mpeg7>
```


Mpeg7 annotation for QUERY3 : Border Collie

```
<?xml version="1.0" encoding="UTF-8"?>
<Mpeg7 xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001 mpeg7-2001.xsd" xmlns="urn:mpeg:mpeg7:schema:2001"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><Description
xsi:type="SemanticDescriptionType"><Semantics id="Dogs"><AbstractionLevel dimension="1"/><Label><Name>Dog
Ontology</Name></Label><SemanticBase xsi:type="AgentObjectType" id="Dog"><Relation type="generalizes" source=
"Dog" target="Dog_Breed"/><Relation type="specializes" source="Dog_Breed" target="Dog"/></SemanticBase><
SemanticBase xsi:type="AgentObjectType" id="Basset_Hound"><AbstractionLevel dimension="1"/><Label><Name>
High Level Features</Name></Label><Property><Name>hasDescription</Name><Definition>["The Basset Hound is a
short-legged breed of dog of the hound family. They are scent hounds, bred to hunt rabbits by scent. Their sense of smell for
tracking is second only to that of the Bloodhound. The name Basset is derived from the French word bas, meaning low, with
the attenuating suffix -et, together meaning rather low. Basset hounds are commonly brown and black and most often
spotted, but also exist in a variety of colors"@]</Definition></Property><Property><Name>hasWordNet</Name><
Definition>["smooth-haired breed of hound with short legs and long ears"@]</Definition></Property><Property><Name>
hasSize</Name><Definition>Large</Definition></Property><Property><Name>hasCountryOfOrigin</Name><Definition>
France</Definition></Property><Property><Name>hasFur</Name><Definition>Short/Medium</Definition></Property><
Property><Name>hasHabitat</Name><Definition>InDoor/OutDoor</Definition></Property><Property><Name>
hasCoatPattern</Name><Definition>Bi-Color</Definition></Property><Property><Name>hasMainColor</Name><
Definition>Brown</Definition></Property><Property><Name>hasSecondaryColor</Name><Definition>White</Definition>
</Property><Relation type="generalizes" source="Dog_Breed" target="Basset_Hound"/><Relation type="specializes"
source="Basset_Hound" target="Dog_Breed"/></SemanticBase></Semantics></Description></Mpeg7>
```


Mpeg7 annotation for QUERY4 : Basset Hound


```
<?xml version="1.0" encoding="UTF-8"?>
<Mpeg7 xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001 mpeg7-2001.xsd" xmlns="urn:mpeg:mpeg7:schema:2001"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><Description
xsi:type="SemanticDescriptionType"><Semantics id="Dogs"><AbstractionLevel dimension="1"/><Label><Name>Dog
Ontology</Name></Label><SemanticBase xsi:type="AgentObjectType" id="Dog"><Relation type="generalizes" source=
"Dog" target="Dog_Breed"/><Relation type="specializes" source="Dog_Breed" target="Dog"/></SemanticBase><
SemanticBase xsi:type="AgentObjectType" id="Labrador_Retriever"><AbstractionLevel dimension="1"/><Label><
Name>High Level Features</Name></Label><Property><Name>hasDescription</Name><Definition>["The Labrador
Retriever (also Labrador, or Lab for short) is one of several kinds of retriever, a type of gun dog. A breed characteristic is
webbed paws for swimming, useful for the breed\'s original purpose of retrieving fishing nets. This and their subsequent use
as hunting companions, gave them the name retriever. The dogs of this breed are very loving, kind and compassionate to
their master. The Labrador is the most popular breed of dog by registered ownership in Canada,[citation needed] the United
Kingdom.and the United States (since 1991),[3] and It is also the most popular breed of assistance dog in Australia, Canada,
the United Kingdom and the United States and many other countries, as well as being widely used by police and other official
bodies for their detection and working abilities. Typically, Labradors are athletic and love to swim, play catch and retrieve
games, and are good with young children."@]</Definition></Property><Property><Name>hasWordNet</Name><Definition
>["breed originally from Labrador having a short black or golden-brown coat"@]</Definition></Property><Property><
Name>hasSize</Name><Definition>Medium</Definition></Property><Property><Name>hasCountryOfOrigin</Name><
Definition>England</Definition></Property><Property><Name>hasFur</Name><Definition>Short/Medium</Definition></
Property><Property><Name>hasHabitat</Name><Definition>InDoor/OutDoor</Definition></Property><Property><Name
>hasCoatPattern</Name><Definition>Single-Color</Definition></Property><Property><Name>hasMainColor</Name><
Definition>White</Definition></Property><Property><Name>hasSecondaryColor</Name><Definition xsi:nil="true"/></
Property><Relation type="generalizes" source="Dog_Breed" target="Labrador_Retriever"/><Relation type=
"specializes" source="Labrador_Retriever" target="Dog_Breed"/></SemanticBase></Semantics></Description></Mpeg7
>
```


Mpeg7 annotation for QUERY5 : Labrador Retriever

# Chapter 7

# Conclusion

DOGi is a framework capable of annotating images of a certain domain using image content and the ontology of that domain.Our demo application deals with images of dog breeds but it can be extended to handle any image domain givven the proper domain ontology.Thus,This would require the construction of a different domain ontology and different training stages adjusted to the images of the new domain. Alternatively, the process of image annotation using our framework can be viewed as an attempt for narrowing the semantic gap between low level features which are easily extracted from unknown images and high level concepts related to these images.

The process of annotating an unknown image is implemented in steps. A query image is provided by the user to obtain similar images from the ontology. Image matching is implemented using image content descriptions. 12 visual descriptors are used.The LIRE similarity measure and an overall similarity measure (DOGi similarity measure) between images are proposed as similarity measures.The relative importance of features in this distance (their weights) are computed using machine learning by decision trees.The semantic category of an unknown image is computed based on AVR(Average Retrieval Rank),summation 0f 9 instances ,max occurence and best match. For interoperability reasons ,annotation of the image is genereted in MPEG-7 format. This is achieved through mapping OWL classes and properties, to elements of MPEG-7 MDS(Multimedia Description Schemes) and finally annotation is stored in Exif metadata tags.

Our experiments show that automatic image annotation using ontologies and image content analysis can be succesfull in percentages above 80% for the first answer givven a good and variant training set for the classification stage and a descriptive ontology for the annotation stage.

## 7.1 Future Work

Our ontology has a specific domain however ,this ontology can be easily extended to upper categories of human perception (animals) or picturable nouns hierarchy of Wordnet [37].

To further automate the annotation process,we can use a ROI detection algorithm to automatically detect the region of interest.Thus,the system could annotate images without any help from the user [38].

In a machine learning view,It is possible to use multiple decision trees using a multilayer approach in order to improve classification.The first decision tree decides if the unkown image is of the domain of interest.Then,a second decision tree classifies the image into the less common categories .Finally a third decision tree decides the semantic category of the unkown image.Also,an interesting extension relates to calculating an overall similarity between images using multiple decision trees as weighting scheme of features in regions of images. Next an overall similarity between images is derived with the sum of similarities that came from regions of an image.

Experiments show that better weighting schemes are possible on low-level features for retrieval purposes if a sofisticated training set is used .Thus,in order to improve the training set,we can add more training samples and more instances for each semantic category.Also,an enriched ontology can provide better descriptions ,therefore we can collect more information for the domain of interest and add them to the ontology.

# Bibliography


[1] Dr David Dagan Feng Dr. Fuhui Long, Dr. Hongjiang Zhang. Fundamentals Of Content Based Image Retrieval. *Science*.

[2] Bottom-up Approaches, Jonathon S Hare, Patrick A S Sinclair, Paul H Lewis, Kirk Martinez, Peter G B Enser, and Christine J Sandom. Bridging the Semantic Gap in Multimedia Information Retrieval. *English*.

[3] Jia Li and James Z Wang. Real-time computerized annotation of pictures. *IEEE transactions on pattern analysis and machine intelligence*, 30(6):985–1002, June 2008.

[4] Liu Wenyin, Susan Dumais, Yanfeng Sun, Hongjiang Zhang, Mary Czerwinski, Brent Field, and One Microsoft Way. Semi-Automatic Image Annotation. *Strategy*, 1999.

[5] Thomas C. Jepsen. Just What Is an Ontology, Anyway? *IT Professional*, 11(5):22–27, September 2009.

[6] Thomas R Gruber. Technical Report KSL 92-71 Revised April 1993 A Translation Approach to Portable Ontology Specifications by A Translation Approach to Portable Ontology Specifications. *Knowledge Creation Diffusion Utilization*, (April), 1993.

[7] Mathias Lux. LIRe : Lucene Image Retrieval - An Extensible Java CBIR Library. *Analysis*, pages 1–3, 2008.

[8] Giuseppe Amato. Using MPEG-7 for Automatic Annotation of Audiovisual Content in eLearning. *Learning*.

[9] Xiangdong Zhou. Automatic Image Annotation By An Iterative Approach : Incorporating Keyword Correlations And Region Matching. *Work*, pages 25–32, 2007.

[10] Avril Styrman. Ontology-Based Image Annotation and Retrieval. 2008.

[11] J Jeon, V Lavrenko, and R Manmatha. Automatic Image Annotation and Retrieval using Cross-Media Relevance Models Categories and Subject Descriptors. *Information Retrieval*.

[12] Kyung-wook Park, Jin-woo Jeong, and Dong-ho Lee. OLYBIA : Ontology-Based Automatic Image Annotation System Using Semantic Inference Rules *. pages 485–496, 2007.

[13] Eero Hyv. Ontology-Based Image Retrieval. *Online*.

[14] Huan Wang, Song Liu, and Liang-Tien Chia. Does ontology help in image retrieval?, 2006.

[15] Vasileios Mezaris, Ioannis Kompatsiaris, and Michael G. Strintzis. Region-Based Image Retrieval Using an Object Ontology and Relevance Feedback. *EURASIP Journal on Advances in Signal Processing*, 2004(6):886–901, 2004.

[16] A Th Guus Schreiber, Barbara Dubbeldam, Jan Wielemaker, and Bob Wielinga. Photo Annotation. *IEEE Intelligent Systems*, 2001.

[17] Pyrros Koletsis. SIA : Semantic Image Annotation using Ontologies and Image Content Analysis. *World Wide Web Internet And Web Information Systems*, 2009.

[18] Vinay Modi. Color descriptors from compressed images. *Signal Processing*.

[19] Evaggelos Spyrou, Le Borgne, Theofilos Mailis, and Eddie Cooke. Fusing MPEG-7 visual descriptors for image classification. *Image (Rochester, N.Y.)*, pages 1–6, 2005.

[20] a Vakali. MPEG-7 based description schemes for multi-level video content classification. *Image and Vision Computing*, 22(5):367–378, May 2004.

[21] Syndicate House. Efficient use of MPEG-7 Color Layout and Edge Histogram Descriptors in CBIR Systems. pages 157–163.

[22] Fernando Pereira. MPEG-7 The Generic Multimedia Content Description Standard, Part 1. (June), 2002.

[23] Applied Information Technology. Cbir using color histogram processing 1 1 2. *Image (Rochester, N.Y.)*, 2009.

[24] Ju Han and Kai-Kuang Ma. Fuzzy color histogram and its use in color image retrieval. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 11(8):944–52, January 2002.

[25] Mandar Mitra, Wei-jing Zhu, and Ramin Zabih. Image Indexing Using Corellograms. *Matrix*.

[26] Katarina Trojacanec, Georgina Mirceva, and Danco Davcev. Application of Edge Histogram Descriptor and Region Shape Descriptor to MRIs. *Artificial Life*, pages 1–10.

[27] P Wu. A texture descriptor for browsing and similarity retrieval. *Signal Processing: Image Communication*, 16(1-2):33–43, September 2000.

[28] Zhi Li, Guizhong Liu, Haixia Jiang, and Xuemin Qian. Image copy detection using a robust gabor texture descriptor. *Proceedings of the First ACM workshop on Large-scale multimedia retrieval and mining - LS-MMRM '09*, page 65, 2009.

[29] Savvas A Chatzichristofis and Yiannis S Boutalis. CEDD : Color and Edge Directivity Descriptor . A Compact Descriptor for Image Indexing and Retrieval. *Image (Rochester, N.Y.)*, pages 312–322, 2008.

[30] Savvas a. Chatzichristofis and Yiannis S. Boutalis. FCTH: Fuzzy Color and Texture Histogram - A Low Level Feature for Accurate Image Retrieval. *2008 Ninth International Workshop on Image Analysis for Multimedia Interactive Services*, pages 191–196, 2008.

[31] Konstantinos Zagoris, Savvas A Chatzichristofis, Nikos Papamarkos, and Yiannis S Boutalis. AUTOMATIC IMAGE ANNOTATION AND RETRIEVAL USING THE JOINT COMPOSITE DESCRIPTOR . *Image (Rochester, N.Y.)*.

[32] Evdoxios Baratis, Euripides G M Petrakis, and Evangelos Milios. Automatic Web Site Summarization by Image Content : A Case Study with Logo and Trademark Images. *Image (Rochester, N.Y.)*.

[33] Stephen R Garner. WEKA : The Waikato Environment for Knowledge Analysis. *Computer*.

[34] Suzanne Little and Chrisa Tsinaraki. MPEG-7 based Multimedia Ontologies : Interoperability Support or Interoperability Issue ? *Computer*, (Ddl).

[35] Ana B Benitez, Di Zhong, Shih-fu Chang, and John R Smith. MPEG-7 MDS Content Description Tools and Applications. *Lecture Notes in Computer Science*, (October 2001):1–12.

[36] Standarization Committee. Exchangeable image file format for digital still cameras. 2010.

[37] George a. Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, November 1995.

[38] Tz-huan Huang Kai-yin Cheng Yung-yu Chuang. A Collaborative Benchmark for Region of Interest Detection Algorithms. *Russell The Journal Of The Bertrand Russell Archives*.