UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

« Bahjat Safadi »

Thèse dirigée par **« George Quénot »**

préparée au sein du **Laboratoire d'Informatique de Grenoble**
dans **l'École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique (MSTII)**

Indexation Sémantique des Images et des Vidéos par Apprentissage Actif

Thèse soutenue publiquement le **« 17 septembre 2012 »**,
devant le jury composé de :

M. Denis Pellerin
Professeur, Université Joseph Fourier, (Président)
M. Alan Smeaton
Professeur, Dublin City University, (Rapporteur)
M. Matthieu Cord
Professeur, UPMC Sorbonne Universités, (Rapporteur)
M. Georges Quénot
Chargé de recherche CNRS, CNRS, (Membre)
M. Hervé Jégou
Chercheur INRIA, INRIA- Rennes, (Membre)
M. Stéphane Ayache
Maître de Conférences, Université de la Méditerranée, (Membre)

Acknowledgements

First and foremost, I offer my sincerest gratitude to my supervisor, Mr. Georges Quénot, who has been always available to advise and support me throughout my thesis with his patience and knowledge, while at the same time allowing me the space to work in my own way. One simply could not wish for a better supervisor. Thank you.

I thank all the members of the jury for their acceptance to judge my work. In particular, thanks to Prof. Alan Smeaton, professor at Dublin City University (Dublin, Ireland), as well as Prof. Matthieu Cord, professor at UPMC Sorbonne Universities (Paris, France), for taking the time to report my thesis and for their interest in this work.

I also thank the members of the MRIM team for their sympathy and for the time we spent together during the years of my PhD thesis.

In addition, I thank the partners of the IRIM GDR-ISIS project for their collaboration to make this work as it is.

This research was supported by OSEO, the French state agency for innovation, as part of the Quaero Programme. Part of the experiments presented in this thesis was carried out using the Grid5000 experimental test bed, being developed under the INRIA ALADDIN development action, with support of CNRS RENATER, several universities and other funding bodies.

I especially thank my family. I love them so much. I know I always have my family to count on when times are rough.

I thank all my friends in Grenoble. I had really great times sharing my life with you in Grenoble. Rami, Wendy, Bashar, Nadia, Antje, Ahmad, Moe, Michael, Andy, Trong-Ton... Thank you all for being nice friends.

Finally, many thanks and love to Maaike. There are no words to convey how much I love her. Maaike has been a true and great supporter and has unconditionally loved me during my good and bad times. She has faith in me and my intellect even when I did not have faith in myself. I truly thank Maaike for sticking by my side all the way until the end of this work, I would not have made it this far without her. Maaike Ik hou heel veel van jou :).

Abstract

The general framework of this thesis is semantic indexing and information retrieval, applied to multimedia documents. More specifically, we are interested in the semantic indexing of concepts in images and videos by the active learning approaches that we use to build annotated corpus. Through out this thesis, we have shown that the main difficulties of this task are often related, in general, to the semantic-gap. Furthermore, they are related to the class-imbalance problem in large scale datasets, where concepts are mostly sparse. For corpus annotation, the main objective of using active learning is to increase the system performance by using as few labeled samples as possible, thereby minimizing the cost of labeling data (e.g. money and time).

In this thesis, we have contributed in several levels of multimedia indexing and proposed three approaches that outperform state-of-the-art systems: i) the *multi-learner* approach (*ML*) that overcomes the class-imbalance problem in large-scale datasets, ii) a *re-ranking* method that improves the video indexing, iii) we have evaluated the power-law normalization and the PCA and showed its effectiveness in multimedia indexing. Furthermore, we have proposed the *ALML* approach that combines the multi-learner with active learning, and also proposed an incremental method that speeds up *ALML* approach. Moreover, we have proposed the *active cleaning* approach, which tackles the quality of annotations. The proposed methods were validated through several experiments, which were conducted and evaluated on large-scale collections of the well-known international benchmark, called TRECVid.

Finally, we have presented our real-world annotation system based on active learning, which was used to lead the annotations of the development set of TRECVid 2011 campaign, and we have presented our participation at the semantic indexing task of the mentioned campaign, in which we were ranked at the 3^{rd} place out of 19 participants.

Keywords: Multimedia Indexing, Annotations, Machine Learning, Active Learning, Active Cleaning, Re-ranking, Descriptor Normalization, Evaluation

Résumé

Le cadre général de cette thèse est l'indexation sémantique et la recherche d'information, appliquées à des documents multimédia. Plus précisément, nous nous intéressons à l'indexation automatique de concepts dans des images et des vidéos par des approches basées sur lapprentissage actif, que nous utilisons pour construire des corpus annotés. Tout au long de cette thèse, nous avons montré que les principales difficultés de cette tâche sont souvent liées, en général, au problème dit du "fossé sémantique". Elles sont également liées au problème du déséquilibre entre les classes dans les collections dimages ou de vidéos à grande échelle, où la plupart des concepts recherchés sont rares ou très rares. Pour l'annotation de corpus, l'objectif principal de l'utilisation de lapprentissage actif est d'augmenter la performance du système en utilisant aussi peu que possible déchantillons annotés, minimisant ainsi les coûts dannotations des données (et donc le temps et largent nécessaires).

Dans cette thèse, nous avons contribué à l'indexation multimédia à plusieurs niveaux et nous avons proposé deux approches qui améliorent de l'état de l'art des systèmes : i) l'approche *multi-apprenant* (ML) qui surmonte le problème du déséquilibre entre classes dans les grandes collections, ii) une méthode de reclassement qui améliore la qualité de l'indexation des vidéos. Nous avons évalué la normalisation par une loi de puissance combinée avec l'ACP et montré son efficacité pour l'indexation multimédia. Nous avons proposé l'approche ALML qui combine lapproche multi-apprenant avec l'apprentissage actif et nous avons également proposé une méthode incrémentale qui accélère lapproche proposé (*ALML*). Nous avons enfin proposé l'approche de "*nettoyage actif*", pour lamélioration de la qualité des annotations. Les méthodes proposées ont toutes été validées par plusieurs expériences, qui ont été conduites sur des collections à grande échelle dans le cadre des campagnes dévaluation internationales TRECVid bien connues.

Enfin, nous avons exploité notre système d'annotation basé sur l'apprentissage actif dans des conditions réelles pour effectuer les annotations de l'ensemble de développement des campagnes TRECVid de 2010 à 2012 et nous avons présenté notre participation à la tâche d'indexation sémantique de la campagne 2011 dans laquelle nous nous sommes classés à la 3ème place sur 19 participants.

Mots cles: Indexation et de recherche des documents multimédia, Annotations, Apprentissage Actif, Nettoyage Actif, Réordonnancement, Normalisation des Descripteur, Evaluation.

Contents

Chapter 1

Introduction

In the last decade, digital multimedia databases have grown at a phenomenal speed in many directions, resulting in an explosion in the number and size of multimedia archives to be organized, including images and video databases. In particular, with the widespread use of digital cameras and the increase of the storage of personal computers, which can reach hundreds of gigabytes, individuals can nowadays easily produce thousands of personal images and videos. Further, the revolution in digital media broadcasting, such as TV broadcasts and digital magazines, and the ability to share multimedia through internet (e.g. facebook[1], youtube[2], flickr[3] etc.) with people all over the world have changed the way we take, store and share documents. This has led to great growth in digital multimedia databases. Therefore, the problems of indexing and searching multimedia have grown too. Thus, ensuring that the multimedia collections are indexed, searched and accessed in an efficient way is an important concern.

Multimedia indexing is one of the main issues for searching documents in the vast of digital multimedia collections. This indexing can be done at the signal level (e.g. color, texture, motion, etc.) or at the semantic level (e.g. concepts). Indexing based on signal level (i.e. low-level features) can be useful in certain domains, such as in the medical domain. For example, a doctor may be interested in searching dark or light zones in an image or a video. However, in public videos, such as news, cinema, TV or personal videos, users aim to find particular sequences that semantically represent their needs: for example, ”the fantastic goal of Marco van Basten against the USSR in the final of the European Championship in 1988”. Thus, semantic indexing is so far the most needed functionality for users, however, it is also the most difficult one, since it needs powerful tools to extract the correct content from multimedia documents. Due to the problem of the so-called *semantic gap* between the raw data content and the elements that make sense to human beings, indexing concepts in multimedia documents is a very hard task. One solution to bridge the semantic gap for a given query, is by manually annotating the whole database. Although this approach might be efficient for small collection, it is still far from being optimal, because it needs extremely expensive and time-consuming work, especially, when the dataset is of large scale. Therefore, it is necessary to provide automatic tools to perform this task. These tools are based on machine learning techniques, which deduce semantic knowledge learned from low-level features. These approaches have to be trained on positive and negative samples

[1] https://www.facebook.com

[2] http://www.youtube.com/

[3] http://www.flickr.com/

(i.e. the training set) to generate models that can be used in predicting labels for unlabeled documents (Naphade & Smith [2004]; Snoek *et al.* [2006]; Tahir *et al.* [2009a]).

While the volume of data that can be manually annotated is limited due to the cost of human intervention, there remains the possibility to select a subset of data samples to be manually annotated, in order that their annotations will be as useful as possible (Angluin [1988]). Deciding which samples are the most useful is not trivial.

Active learning is an approach in which an existing system is used to predict the usefulness of new samples. This approach is a particular case of incremental learning, in which a system is trained several times with a growing set of labeled samples. The objective, is to iteratively select as few samples as possible to be manually annotated; these samples will lead to an improvement in classification performance.

In this thesis, we have developed an approach based on active learning, which helps to produce useful training sets of multimedia databases, which are used in classification and indexing processes. Also, we propose some useful approaches that improve the performance of multimedia indexing systems. Moreover, we have built an indexing system that uses the aforementioned approaches for multimedia indexing in a real world application, which was used in building the development sets of TRECVid 2010 and 2011 (the well known international campaign). The terms *annotation* and *label* in this thesis refer to textual keywords that describe the content of a multimedia document.

1.1 Multimedia Indexing based Active Learning System

Information retrieval systems (IR) are designed to meet users' needs in searching documents. Therefore, the goal with such systems is to retrieve the most relevant documents that match the query generated by the user. For multimedia documents, usually, these systems are based on manual annotations or tags of the documents. The more relevant annotations we have, the better IR system we get.

In large-scale databases, the quality and quantity of the annotated documents have great importance. Since they both affect the learning algorithms, training sets with annotations appropriate in quality and quantity are needed. Even though the more documents we annotate the better performance we get, having the dataset fully annotated costs more time and money. Active learning approaches are tools to be considered as promising solutions for creating training sets. Figure 1.1 shows an instance of a multimedia indexing system based on active learning. Given a training set which contains very few labeled samples L and a large number of unlabeled samples U. The samples are represented by their low-level features. The system is iterative, based on learning classifiers from the available labeled samples (L) to predict the usefulness of the unlabeled samples (U). Using the predicted scores the system proposes the most informative unlabeled documents to oracles in order to label them. Then we update the sets L and U. This process is repeated until the breaking condition is achieved (e.g. all the unlabeled samples have been labeled). Further, the performance of the active learner is measured as a function of an actively selected training set. This is done using an independent set (i.e. the test set).

The performance of the system is based on data description techniques, the classification algorithm and the selection strategy for the samples to be annotated. In this thesis, we are inter-

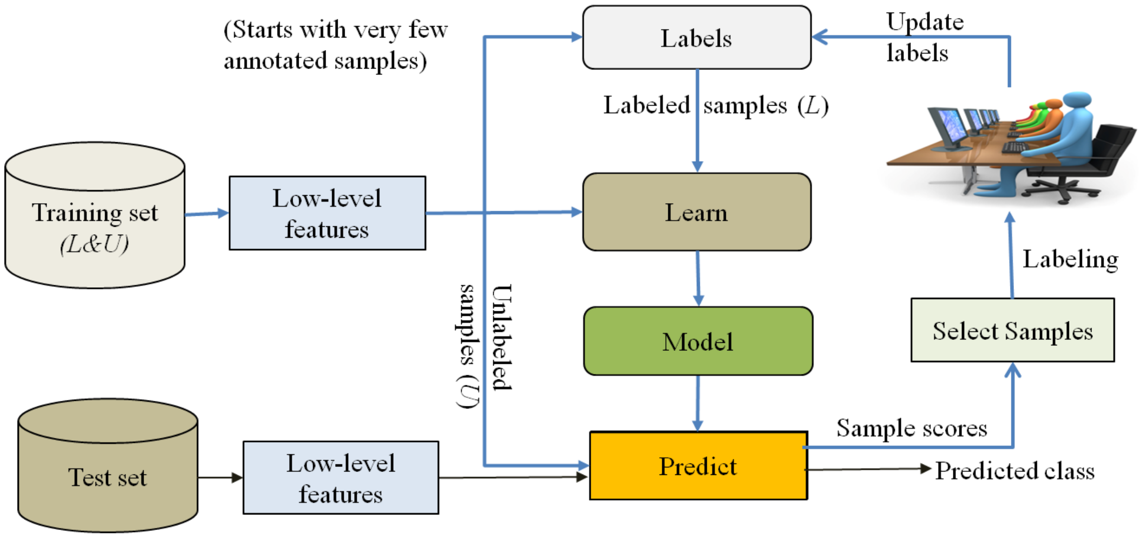


Figure 1.1: Active learning system.

ested in indexing multimedia documents, in particular videos. The objective of the system is to help build training sets for a video corpus by assigning semantic labels to each video. This requires the system to effectively represent videos in a given corpus, so that these representations can lead to the retrieval of preferred data to users. We give an overview of semantic indexing in the following section.

1.2 Semantic Indexing

Systems which use the semantic description of documents are considered the next generation of document indexing, since they allow semantic access to multimedia documents. Rather than interacting with the system using low-level descriptions, users can interact at high-level semantics. This allows users to interact with the system, using key-words of semantic concepts (Lim [2001]; Snoek *et al.* [2005a]).

Due to the difficulties in extracting symbols from a signal (i.e. low-level), these systems become nowadays a very hot topic in the information retrieval community. The advantage of such systems is their ability to abstract representation from low-level features. Indeed, they must be able to index concepts considering visual similarities, since a concept can be described in different colors and shapes. Associating the concepts with their visual features is a non-trivial task. It requires knowledge of the referred concept to avoid ambiguity when detecting it.

In multimedia semantic indexing systems (MSI), passing from low-level to high-level (e.g. concept) is usually done by supervised machine learning approaches. For a given concept, this is based on the use of a labeled set of samples (annotated as positive or negative) to learn a classifier to recognize the same concept in unlabeled documents. Even though these systems better meet the users' needs, they are still difficult to implement. Furthermore, it is difficult to define all known concepts, and it is still impossible to build intelligent MSI systems that fully meet the users' expectation. Moreover, supervised learning algorithms suffer a lot from the data representation and the quality of the training samples. This can be seen while learning concepts from large-scale databases, where they often face the class imbalance problem between the minority and majority classes, since most of the target concepts are very sparse.

In the next section, we discuss the major problems in building effective multimedia indexing systems.

1.3 Problem Description

As our work is based on multimedia indexing, we identify two main problems that multimedia indexing systems usually face. The first is the major problem of bridging the semantic gap between the low-level visual features (e.g. color, shape, texture, etc.) and the semantic concepts identified by the user (Chang *et al.* [2009]; Idrissi *et al.* [2009]; Smeulders *et al.* [2000]). The second problem is the class imbalance problem that appears mostly in large-scale databases such as TRECVid (Smeaton *et al.* [2006]), where most of the target concepts to be learned are sparse. This is still a serious problem for classical supervised learning methods. In addition, another (minor) problem is the mismatch between the evaluation metric and the internal criterion of the classifier. The common metric used in information retrieval is the mean average precision (MAP), which gives more importance to the retrieved samples in the top of the ranked list, whereas the internal criterion of the supervised learning algorithm expects to find classes roughly balanced and is generally not influenced by ranking.

1.3.1 Semantic Gap

In the last decade, the problem of the semantic gap has been highlighted. Smeulders *et al.* [2000] described the semantic gap as follows: *"The semantic gap is the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation"*. Bridging the semantic gap in image retrieval is a very challenging problem yet to be solved (Chang *et al.* [2009]; Idrissi *et al.* [2009]). As human beings, we are able to understand image and video documents in both low-level features (e.g. colour, texture, shape, etc.) and high-level semantics (e.g. concepts, events, etc.). However, machines use only low-level features to interpret these documents. Users prefer to search documents using high-level queries (Liu *et al.* [2007b]; Smeulders *et al.* [2000]), while content-based image and video retrieval (CBIR CBVR) systems use low-level features to index images and videos. An attempt to overcome the gap between high-level semantic and low-level features is needed. For example, can we build machines that can understand semantically the content of the image shown in figure 1.2. As we can see, it is a real challenge to make the machine understand that the picture is taken on a mountain, it is cloudy and there is snow (taken in Winter), the picture focuses on three ladies from the seven people, the activity is skiing.

As an attempt to bridge the semantic gap, automatic multimedia annotation techniques have attracted a lot of interest in recent years. The aim of auto-annotation techniques is to attach textual labels to unlabeled documents, as the descriptions of the documents' content. This labeling should be done using high-level semantics.

Figure 1.2: Skiing at the Belle-Plagn in the French Alpes, an example of the semantic gap.

1.3.2 The Class Imbalance Problem in Large-Scale Databases

A dataset is called imbalanced if at least one of the classes is represented by a significantly fewer number of samples than the others. In other words, data are said to suffer from the class imbalance problem when the distributions of the classes are highly imbalanced; this issue occurs in many real-world databases including those of multimedia. Thus, the class imbalance problem became a hot topic within machine learning in many particular domains, especially in multimedia indexing.

For the binary-class case (positive and negative samples), one assumes that the positive class is the minority class, and the negative one is the majority class. In many cases, the minority class is very infrequent, such as less than 1% in TRECVid databases (Smeaton *et al.* [2006]), as shown in figure 1.3. In this context, standard supervised learning algorithms have very low predictive accuracy for the infrequent class. Thus, when applying them as classification methods on imbalanced databases, they are likely to predict everything as negative (i.e. the majority class). This was often considered as learning from a highly imbalanced dataset (Japkowicz & Stephen [2002]; Weiss & Provost [2001]).

One major research direction to overcome the class imbalance problem, is to re-sample the original training dataset, either by over-sampling the minority class or by under-sampling the majority class. This is done in such a way that classes will be presented in a more balanced way (Bishop [2007]; Chawla *et al.* [2002]; Weiss & Provost [2001]). Over-sampling causes longer training time and inefficiency in terms of memory, due to the increased number of training instances; it suffers as well from high computational costs for preprocessing the data. On the other hand, under-sampling may lead to a loss of information, due to the fact that it may discard useful data that could be important for the learning process.

The underlying motivation for re-sampling methods is to provide a classifier with a training

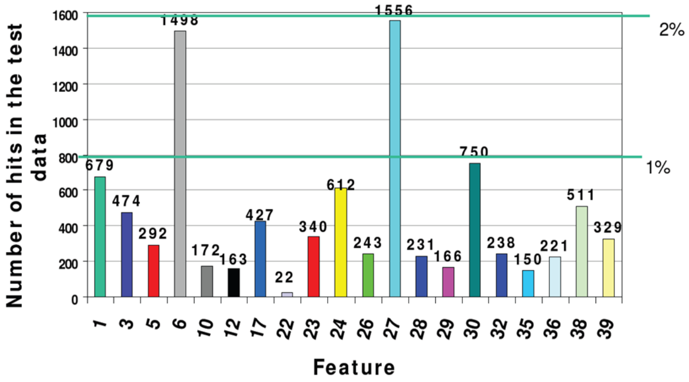


Figure 1.3: Concept frequency, from Smeaton *et al.* [2006].

set, which has more balanced classes. This motivates us to search for or to improve techniques that can overcome the class imbalance problem in multimedia document indexing.

1.4 Aims and objectives

Our work focuses mainly on semantic indexing of multimedia documents. More specifically, we are interested in the semantic indexing of concepts in multimedia documents (e.g. images and videos) by the active learning approaches that we use to build annotated corpora, which are appropriated in quality and quantity. The main aim of this thesis is, to develop and evaluate promising active learning approaches for automatic image and video indexing in large-scale multimedia datasets. The indexing is done by either developing new techniques or improving different techniques in the information retrieval community, especially content-based multimedia retrieval. We hope that the system will be able to index a large number of different concepts. Thus, the approach should be generic that we use the same approach whatever the target concept. Moreover, we hope to apply our indexing system to help in annotating large-scale training sets, such as the evaluation campaign TRECVid.

An important point of this work is the evaluation aspects of our contributions. We wish to validate the proposed models in optimal conditions. Thus participation in large evaluation campaigns (e.g. TRECVid and ImageCLEF) is a major objective of this thesis. This induces that the proposed approaches will be capable to handle large corpus of data.

The efforts toward achieving the objectives consist of three parts: i) the development of suitable multimedia document descriptions (e.g. visual and audio descriptors); ii) the development of an advanced machine learning technique for image and video classification; iii) the development of a system based active learning approach to conduct the creation of the development set, by iteratively asking oracles to annotate unlabeled documents from the dataset. In this thesis, we will explore these three areas, though the focus of the research is on the second and third parts.

1.5 Contributions

This thesis brings a number of contributions to the field of automatic multimedia Indexing and active learning. They are summarized briefly as follows:

- A review of some quality issues of image and video indexing approaches, including image description and classification systems that are used for research on multimedia indexing.
- A proposal for a classification method based on the multi-learner approach for concept indexing in highly imbalanced datasets of multimedia documents, especially those containing images and videos.
- A proposal for a re-ranking approach for improving the performance of semantic video indexing and retrieval systems, based on the homogeneity and nature of the video contents.
- A proposal for a method for image descriptor optimization, which consists of two steps: power transformation and dimensionality reduction using Principal Component Analysis (PCA).
- A proposal for an approach that combines active learning and the proposed multi-learner approach for corpus annotation and concept indexing on highly imbalanced datasets.
- A proposal for a new incremental active learning approach based on the multi-learner for image and video annotations.
- A proposal for an approach denoted as active cleaning to improve the quality of annotations, based on the selection of samples that appear as wrongly annotated, to be re-annotated.
- Finally, integration of the proposed approaches in developing a system based on active learning for video indexing in a real-world application. This was applied to conduct the annotation process of the TRECVid development set for the years 2010 and 2011.

This thesis has led to one refereed journal publication, five refereed international conference papers, ten refereed international workshop papers ans two refereed national conference papers, as detailed in appendix C.

1.6 Thesis Structure

This thesis presents the work carried out to achieve the goals outlined in section 1.4. The structure and content of the thesis are described in the following.

Chapter 2 - Background Introduces the background to this work. It starts by presenting research on content-based multimedia indexing and retrieval. It then reviews a number of different automatic multimedia annotation techniques in the literature, including image and video description and optimization, classification approaches and data fusion techniques. Moreover, it introduces the active learning approach. Finally, it describes the performance metrics for evaluation, and presents an example of the evaluation campaigns, named TRECVid, which is a famous campaign in video indexing.

Chapter 3 - Multimedia Indexing and Retrieval Presents a variety of techniques and challenges regarding a Multimedia Semantic Indexing and Retrieval. It proposes a novel technique based on multi-learner (ML) approach for image and video classification. Moreover, it proposes new techniques for improving the classification performance, including an approach for re-ranking samples in video retrieval system and descriptor normalisation. Evaluation experiments on the proposed methods are also presented.

Chapter 4 - Simulated Active Learning methods for Multimedia Indexing Proposes the integration of the multi-learner approach as a classifier for the active learner, called ALML. It also proposes an incremental method to speed up the performance of the ALML approach, denoted as Inc-ALML. Furthermore, it presents some evaluation experiments on the proposed methods.

Chapter 5 - Active Cleaning For Multimedia Indexing Gives an overview of the problem of noisy annotations and presents the collaborative annotation. It proposes an approach, denoted active cleaning, to reduce the noise in the annotations. It ends with results and conclusions derived from some experiments on the method.

Chapter 6 - Real-World Application This chapter presents our system for video annotations applied to TRECVid (i.e. TRECVid 2010 and 2011). It also gives our results at the TRECVid 2011 and ImageCLEF 2011 evaluation campaigns.

Chapter 7 - Conclusions and Perspectives Discusses and concludes the overall results and contributions from the work presented in the previous chapters. Then it gives some pointers to the perspectives and future work of this thesis.

Chapter 2

Background

This chapter introduces the background of this work. It starts by presenting the research on content-based multimedia indexing and retrieval. It then reviews a number of different automatic multimedia annotation techniques in the literature, including the image and video description and optimization, the classification approaches and the data fusion techniques. Moreover, it introduces the active learning approach. Finally, it describes the performance metrics for evaluation, and presents an example of the evaluation campaigns, named TRECVid, which is a famous campaign in video indexing.

2.1 Content-Based Multimedia Indexing and Retrieval (CB-MIR)

Multimedia indexing approaches can be categorized in two main levels of video content: *low-level* features (e.g. color, texture, etc.) and *high-level* semantics (e.g. concepts, events). Low-level features can be fully automated by machines, and a user can interpret them as well. However, low-level indexing ignores the semantic content, whereas users are mostly interested in this content when searching multimedia documents. It is usually difficult for users to describe low-level characteristics of certain objects they are searching for. The high-level semantic-based indexing approaches have benefits, mainly the support of flexible, powerful, and more natural querying. Even though it can be achieved by manually labelling them through users, such as the process of mapping the low-level features to semantic concepts (annotating), it faces the problem of a semantic gap. Manual annotation should be minimized as it can be very expensive and time-consuming.

Concept indexing in image and video documents is important for a content-based search. Such indexing techniques are generally performed by using classifiers. These classifiers are usually trained by supervised learning methods using low-level descriptors of the training set (i.e. the set of the labeled samples). The samples of the training set are, for each target concept, manually annotated as positive or negative.

The semantic indexing of multimedia documents is based on labeling documents using their low-level features. Thus, in order to achieve a good performance on this task, the use of supervised learning is typically more appropriate. After learning, the problem of semantic char-

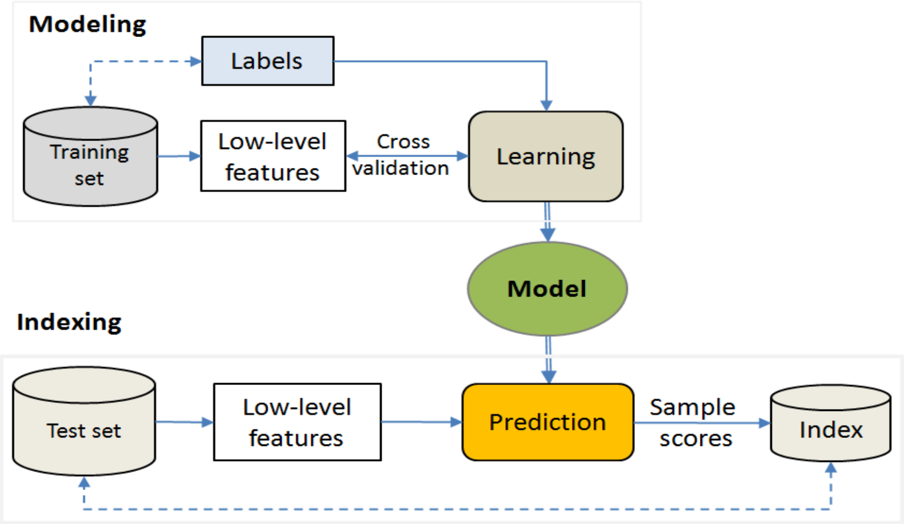


Figure 2.1: Architecture of a basic multimedia indexing system.

acterization can be given as: ”Given a set of low-level features (X) and a set of concepts (C); for each sample ($x \in X$), what is the most likely concept c that x belongs to?” . Typically, a set of N training samples $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ is given. The input x_i (i.e. low-level features vector) forms a feature space X, and the output y_i (i.e. the target class) has a class label c, which belongs to a infinite set C. A classification rule is designed based on the training data so that, given a new input x, a class $c \in C$ with the highest probability score is assigned to it.

An approach of the automatic indexing system, as shown in figure 2.1, consists of two main phases: modeling and indexing. For a given target concept, in the modelling phase the system learns the relations between the descriptors (i.e. low-level features) of the training set and their associated labels, and it generates a classification model. Such a model can then be used to assign scores to new samples by providing their descriptors. In the indexing phase, the model is applied on the unlabeled samples (i.e. test set). It will generate the predicting scores, as the likelihood to contain the learned concept, for each of these samples. The retrieval task can be then achieved by ranking the test samples according to their scores, in which samples in the top of the ranked list are considered to be positive samples.

In the following sections, the major components for multimedia indexing are discussed, including the structure of video document, description and classification.

2.2 Video Documents

A video document is defined as a combination of different data streams (flows), where visual and audio are the two main data flows. The visual flow consists of a sequence of images which are animated, usually, in 24 to 30 images per second. The audio flow is a mix of different sounds (e.g. mono, stereo), which are, typically, sampled from 1600 to 48000 kHertz. A third flow which can be associated, is text. It may appear in some video documents to help users understand the audiovisual presentation.

There are some significant characteristics that distinguish video from other types of data. Video has higher resolution, larger data volume, a larger set of data that can be originated, higher interpretation ambiguity, and needs more interpretation efforts. Furthermore, video has spatial and temporal dimension, whereas text is only non-spatial static and image is only spatial static. Moreover, video semantic is unstructured and generally contains complex relationships.

2.2.1 Video Segmentation

In general, concept indexing can be performed on the whole video (one label for each video), but this would be very rough. On the other hand, if the indexing is based on each frame in the video, it would be too dense, as a frame does often not contain any important information. Therefore, researchers have regularly indexed videos of the shot level, which consists of sequential frames of a continues camera view. A commonly used model is the *model based layers* proposed by Smith & Davenport [1993], which was widely applied to segment video documents (Chua *et al.* [2002]; Kankanhalli & Chua [2000]; Weiss *et al.* [1995]). We design each layer as a list of non-overlapping sequences according to the time axis. The layers are hierarchically organized. Thus, the first layer contains all the frames within the video, and then, the more we go up on the hierarchy structure, the more the sequences are grouped together according to their semantic contents. Hence, it makes a useful tool for video indexing; this approach is the basis of automatic indexing. Video segmentation is therefore considered the major step in video indexing.

The hierarchy of video segmentation is given in figure 2.2. The first layer contains the images of the video; these images are grouped to form shots, then the shots are grouped to represent scenes; lastly, the scenes are grouped to form stories. In this context, we are interested in indexing the shot and story layers. Essentially, a shot consists of sequential frames sharing common semantics, while a story is the most abstract unit of the video, which can have different human interpretation. Thus, one can say that several segments of stories are valid for the same video.

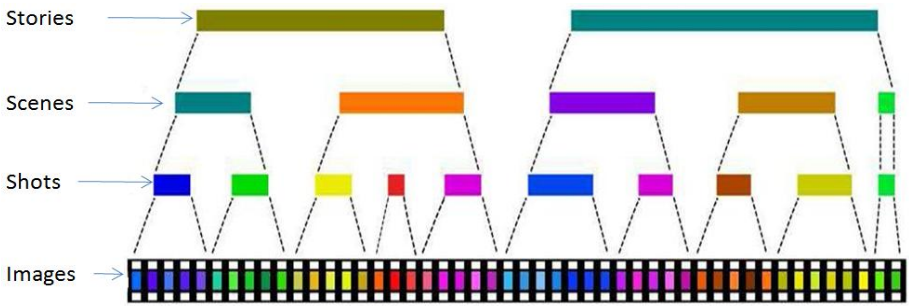


Figure 2.2: An example of video document structuring, from (Ayache [2007]).

The shots are identified by the video segmentation process, where it depends mostly on the visual flow (Boreczky & Rowe [1996]). It can be done by grouping images according to their low-level features. Video segmentation is a problem which has attracted much attention since video became available in digital form as it is an essential pre-processing step to almost all video analysis, indexing, summarization, search, and other content-based operations. Many works have been realized in this domain in the last decade, especially in the context of TRECVid. Since, automatic shot boundary detection (SBD) task was one of the tracks of activity within the annual TRECVid benchmark tasks (i.e. from 2001 to 2007). It is defined as a process of automatically detecting the boundaries between shots in video. The work of Smeaton *et al.* [2010] summarizes the work of 7-years history of the SBD task in TRECVid, in which 57 different research groups from across the world had participated to determine the best approaches to SBD, while using a common dataset and common scoring metrics. The authors present an overview of the TRECVid shot boundary detection task, a high-level overview of the most significant of the approaches taken, and a comparison of performances, focusing on one year (i.e. 2005) as an example. Among these systems, for example the system presented by Quénot [2001], which provides since several years the official video segmentation tool for TRECVid. This approach identifies the transition type (e.g. cut, degraded) with good precision. In addition, the segmentation algorithms extract one representative image of each shot. This step is essentially important, since the shot is the basic unit for the classification, searching and browsing systems based on the content of video datasets (Naphade & Smith [2004]; Over *et al.* [2005]). In practice, the automatic indexing process of videos is to detect concepts in the extracted shots. Recently, story segmentation in video became a hot research, in which the goal is to organize videos into meaningful units, based on stories. Therefore, the story units are indexed and organized to facilitate the retrieval and browsing of video documents.

2.3 Multimedia Document Description

Digital multimedia documents are generally considered as two or three dimensional values, such as images and videos respectively. Before they are automatically analysed (e.g. indexing, detection or recognition), they need to be condensed from Visual and audio data into feature values, so that the important information of the problem that has to be solved, can be retained while the redundancy can be removed. For example, to distinguish videos with red cars from those with blue cars, we can describe the videos by their colors rather than using shape or audio description. Thus, we define the multimedia document description as the process of generating descriptors that represent the visual and audio content of documents in a certain manner.

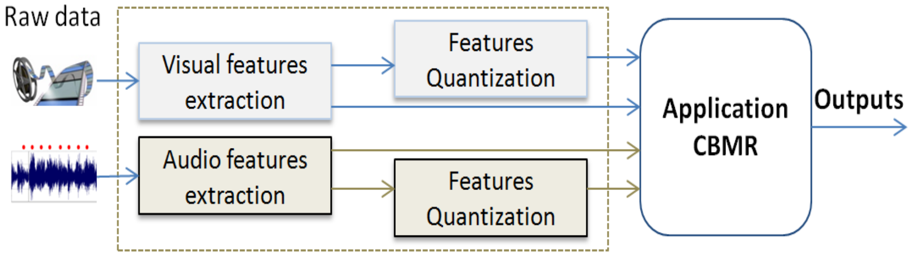


Figure 2.3: Multimedia document description.

Figure 2.3 shows the process of multimedia description, which consists mainly of two ways: i) the visual features extraction, which condenses the pixel color-values into feature values, and ii) the extraction of the audio feature values. In addition, a feature quantization step (i.e. aggregation), which projects feature values from continuous space into discrete space, can be applied on both visual and audio feature values. For each document, The description step results one or more descriptors that will be used to represent it in the designed application (e.g. indexing or retrieval). However, it is clear that for image description we only follow the visual features extraction, while visual and audio descriptors are used in representing videos.

In multimedia indexing and retrieval, it is recommended to use both visual and audio descriptions. Theoretically, the combination of different kinds of features (i.e. descriptors) will produce a more robust multimedia indexing system. Thus, the basic system of multimedia indexing is extended to contain more low-level descriptors and a fusion step, which is applied on the predicted scores and produces one final score for each sample, as shown in figure. 2.4. The indexing process is done by ranking the samples regarding to their final scores.

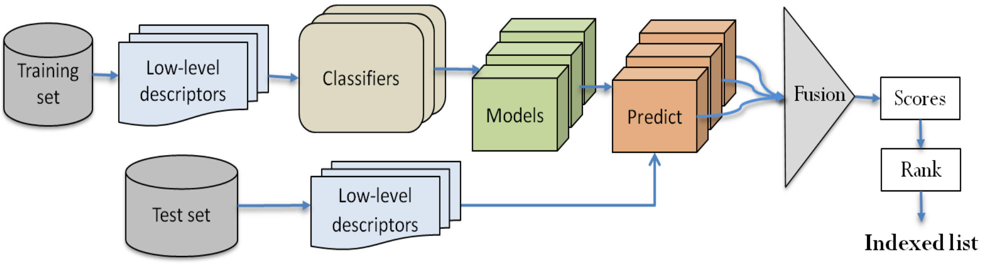


Figure 2.4: A standard content-based multimedia indexing system.

In the following, we will give an overview of the most used multimedia descriptors, including visual and audio descriptors.

2.3.1 Visual Descriptors

Visual features can be extracted to describe the visual content of a multimedia document globally or locally. When the region is chosen to be the whole document (e.g. image), features are global. When the region is chosen to be a part of the document, features are local and they describe individual parts of the image. Features can also be categorized as being general or domain-specific. General features include commonly used features such as color, shape and texture. For special applications (e.g. human faces, fingerprints and human skin) however, general features are not applicable, so there is a need to develop domain-specific features. Visual description is mostly based on image processing techniques, since a video is considered as a sequence of images. A video can be processed, visually, as images unless if we consider the motion description then special feature extraction are needed, for instance the space-time interest points descriptor (STIP) proposed by Laptev & Lindeberg [2003]. Thus, in video documents, the methods of image representation are applied on each video shot (Naphade & Smith [2004]; Over *et al.* [2005]). Certain studies of video indexing use object or face tracking in particular contexts. In the following, different visual descriptors, which are used in content-based multimedia indexing and retrieval (e.g. color, shape and texture), are discussed, including the details of those used in this thesis that are given in appendix.B.

2.3.1.1 Color

Color is perhaps the most popular visual feature used in image representation. It can be expressed in different types of color-spaces, such as RGB. Color indexing is normally based on two aspects, the color-space and its representation. The most common color-based feature representation is the *color histogram*, which characterizes effectively the global distribution of an image (or a region of the image) without knowing the objects that form the image. Because color histograms represent images with no information about locations, they are invariant to translation and rotation of objects, and they are not stable in case of change of illumination. The first use of color histograms for image retrieval was proposed by M.J. Swain [1991]. It is calculated by discretizing the colors in the image into a number of bins, then counting the number of pixels in each bin. Their efficiency depends mostly on the color-space and the quantization method. Color histograms can be applied on several color spaces including RGB, YUV, HSV and CYLAB (Wan & Kuo [1998]). Image descriptors based on color histograms are often computed globally; however, some researchers used it as locally region-based.

Another way to characterize the color is by applying statistical methods, such *color moments*, which have been applied in image and video retrieval. The fundamental idea behind this approach is that any color distribution can be characterized by its moments. As most of the information is concentrated in the low-order moments, only the first three moments (i.e. mean, variance and skiwness) can be used for robust and compact color content representation. Jau-Ling & Ling-Hwei [2002] have characterized the image blocks by the first three statistical moments of the three components of HSV. Weighted Euclidean distance can then used to compute color similarity. Stricker & Orengo [1995] compared the performance of an image search system according to several color characterization, and showed that the use of color moments significantly increases the rapidity of the system (for the similarity calculation) with accuracy comparable to the color histograms.

2.3.1.2 Shape

Shape is another important visual feature, which is considered, however, as a hard task to be fully automated. To extract shape features of a visual object or a region, firstly, image segmentation techniques are required, then a geometric description on the segmented objects or the regions are applied. In many cases, especially when an accurate detection is required, a human intervention is needed. A shape descriptor attempts to quantify the form of an object in the way of human perception. There are several techniques to characterize forms, such as perimeter, area, bounding box, convex hull, etc.

2.3.1.3 Texture

Texture is an important image feature, as it describes visual patterns that are homogeneous and not produced from a single color. These visual patterns distinguish structural arrangements of surfaces from the surrounding environment. Therefore, texture is a natural property of all object surfaces, such as woods and bricks. In general, the texture is characterized by a spatial ordering of grayscale image pixels.According to Tuceryan & Jain [1998] *"we recognize texture when we see it but it is very difficult to define"*; they distinguish four main types of approaches used in

CBIR:

1. Statistical approaches: these characterize the statistical properties based on the occurrences of grayscale values in the image. The best known is the statistical approach of the co-occurrence matrices (Haralick [1979]), in which the authors extract statistical parameters such as mean, variance and correlation. The co-occurrence matrices are very costly to calculate, but they have a strong discriminative power. QBIC (Flickner *et al.* [1995]) uses one of these approaches.

2. Geometrical approaches: these are based on the study of human perception, such as the descriptor of Tamura *et al.* [1978], which characterizes the granularity, direction and contrast in the image.

3. Spectral approaches: these are applied on signal processing. Wavelets and Gabor filters (Manjunath & Ma [1996]; Turner [1986]) are widely used in image description. They can capture frequency and the main directions in an image. The extraction can be done at different scales. Moreover, they are effective in image classification (Turner [1986]).

4. Modeling approaches: these attempt to model a predefined texture. Although it is difficult to adapt these approaches to natural textures, they are being used in generating textures (Elfadel & Picard [1993]).

2.3.1.4 Motion

Motion characterization is one of particularities of video analysis; especially at the base of the segmentation process in shots. Typically, a sequence of images containing little movement is grouped to form a video shot. In addition to being useful for the segmentation into shots, the characterization of the movement will estimate the motion of objects in the video stream (Courtney [1997]), as well as those of the camera (i.e. zooming, translation, rotation). Finally, the characterization of the movement is a discriminative index to classify different types of scenes. There are two main approaches for the motion characterization within a video: i) approaches based on the MPEG (Moving Picture Experts Group) stream, and ii) approaches for optical flow (Horn & Schunck [1981]). The movement extraction, directly, in the compressed domain has the great advantage to limit the calculation time using already existing information. We do not care about compression modes because they are numerous and has dedicated to change. We just note that there are methods that exploit the information extracted during the compression to characterize the movement (Kobla *et al.* [1997]; Shen & Delp [1995]). Although some costly, treatments performed in the compressed domain are of low quality due to a coarse-grained depending on the type of compression. In contrast, the optical flow is a vector field on the complete image, a velocity vector for each pixel of the image. Many technical computing optical flow have been developed from the 80s (Barron *et al.* [1994]). Generally, optical flow is represented by a histogram or by the average movement (Wang *et al.* [1999]).

2.3.1.5 Local Descriptors

Recently, local invariant features based image descriptors have shown interest and successful for many computer vision fields and multimedia document analysis (Ferrari *et al.* [2006]; Leibe &

Schiele [2003]; Lowe [2004]; van de Sande *et al.* [2008]; Yang & Newsam [2010]). For image description, approaches using locale invariant features perform in to steps. First, the detection step, which identifies interesting locations in the image (termed interest points), is needed. These are done usually depending on some measures of saliency. Second, is to calculate a descriptor for each of the image patches centred at the detected locations. There are many desirable properties of the detection and descriptor components of local invariant features, they are summarized in the following:

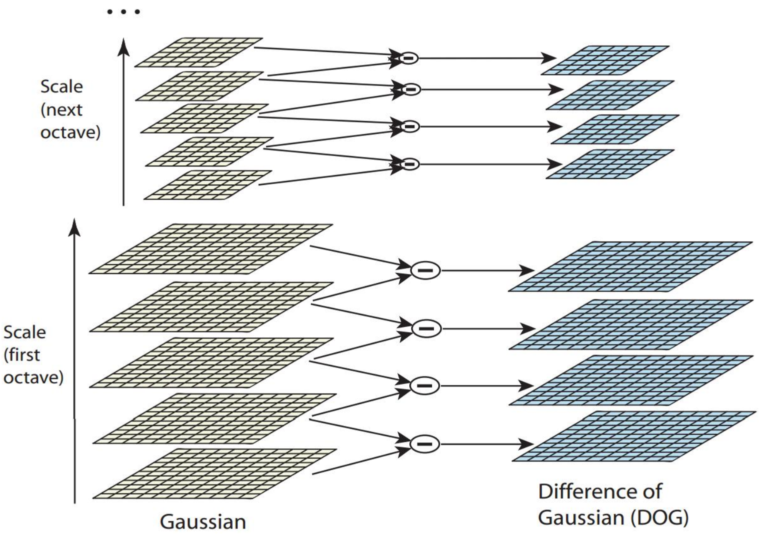


Figure 2.5: The Difference of Gaussian (from Lowe [2004])

Local: The local property of the features makes their use robust to two common challenges in image analysis. First, they do not require any segmentation step, since they are not calculated for image regions corresponding to objects, which is a serious challenging issue. Second, since objects are not considered as a whole document, the features provide robustness against occlusion.

Invariance: Local image analysis has a long history including corner and edge detection. However, the success of the more recent approaches to local analysis is largely due to the invariance of the detection and descriptors to geometric and photometric image transformations. Geometric transformations result from changes in viewing geometry and include translation, Euclidean, similarity, affine, and projective, the most general linear transformation in which parallel lines are not guaranteed to remain parallel. Photometric image transformations result from variations in illumination intensity and direction. Invariance is typically obtained in both the detector and descriptor by simply modeling the transformations as being linear and relying on changes in intensity rather absolute values. The invariance is then important; as an invariant detector will identify the same locations. Thus, an invariant descriptor will remain the same.

Robust: The features should be robust and not be greatly affected by other transformations (e.g. image noise, blur, compression, etc.). The detection should be sufficiently sensitive to the underlying image signal, therefore, it makes the descriptors to be sufficiently distinctive.

Density: Hence the detection is image dependent, it normally results in a large number of features. This density of features is important for robustness in classifying and indexing documents. However, the large number of features leads to representation challenges. The histograms of quantized descriptors (see section 2.3.3), have shown to be effective and efficient methods for summarizing these features.

Efficient: The extraction of local invariant features can be made computationally efficient. This is important when processing large-scale collections, such as in multimedia analysis and real-time applications.

A variety of work on interest point detection, in images and videos, are proposed at both fixed (Harris & Stephens [1988]; Würtz & Lourens [1997]), and at varying scales (Lindeberg [1998]; Mikolajczyk & Schmid [2001, 2002]). An interesting study on the evaluation of interest point detectors is given by Schmid *et al.* [2000]. Lowe introduced an interest point detector invariant to translation, scaling and rotation and minimally affected by small distortions and noise (Lowe [2004]). He also uses the scale space, built with a Difference of Gaussian (DoG), in which the candidate locations are selected from local extrema in DoG filtered images in scale space. The DoG images are derived by subtracting two Gaussian blurred images with different σ, therefore,

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

where $L(x, y, \sigma)$ is the image convolved with a Gaussian kernel with standard deviation σ, and k represents the different sampling intervals in scale space. An example of constructing $D(x, y, \sigma)$ (the DoG process) is shown in figure 2.5, where for each octave of scale space, the initial image is repeatedly convolved with Gaussians to produce the set of scale space images shown on the left. Adjacent Gaussian images are subtracted to produce the difference of Gaussian images on the right. After each octave, the Gaussian image is down-sampled by a factor of 2, and the process is then repeated.

A great number of different local visual feature descriptors have been proposed for describing the content of a multimedia document. Examples of descriptors using this measure are: SIFT (Lowe [1999]), SURF (Bay *et al.* [2006]), GLOH (Winder & Brown [2007]) and STIP (Laptev & Lindeberg [2003]). Mikolajczyk & Schmid [2005] have compared several local descriptors and have showed that SIFT (Scale Invariant Feature Transform) based descriptors (Lowe [1999, 2004]) perform the best.

2.3.1.6 Scale Invariant Feature Transform (SIFT)

We give the scale invariant feature transform (SIFT) as an example of local descriptors, because it is the most used local descriptor for image and video indexing. The approach was named SIFT, as it transforms visual data (e.g. image) into scale invariant coordinates relative to local features. The SIFT descriptor (Lowe [2004]) is designed to be invariant to image scaling, translation and rotation, and partially invariant to change in illumination. It encapsulates the information on gradient magnitude and orientation at each salient region. An extensive comparison with other local descriptors found that the SIFT descriptor performed the best in an image matching task (Mikolajczyk & Schmid [2005]).

In general, interest point based image and video analysis, including SIFT, is a two-step process. First, a detection step locates points that are identifiable from different views. This process

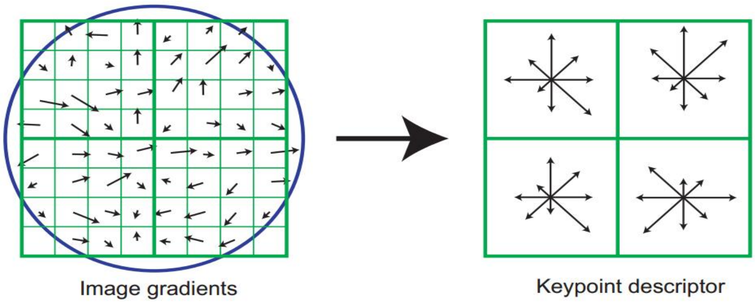


Figure 2.6: The SIFT feature descriptor (from Lowe [2004])

ideally locates the same regions in an object or scene regardless of viewpoint or illumination. Second, these locations are then described by a descriptor that is distinctive yet invariant to viewpoint and illumination. The SIFT-based analysis exploits image patches that can be found and matched under different imaging conditions. The construction of the SIFT description is presented in figure 2.6, and performed as follows:

1. calculate the maximum orientation of each point (p_{xy}) of the detected points of interest (e.g. DoG);

2. divide the space of each p_{xy} in a 4×4 square, the side of each square is equal to N^2, where N is an integer (typically equal to 2);

3. calculate the gradient for the $4 \times 4 \times N^2$ pixels in the square;

4. for each square, compute a histogram of orientation, which is quantized into 8 directions.

To be invariant to rotation, the direction of the local point of interest (p_{xy}) is used as the source (zero orientation) histograms.

The four steps are repeated on several scales of the image. Let E be the number of scales, the descriptors are formed, therefore, vectors of size $E \times 8 \times N^2$. Lowe [2004] suggests to quantize gradient location into a 4×4 location grid ($E = 4$ and $N = 2$), and orientation into 8 orientation bins. This generates a descriptor of $4 \times 8 \times 4 = 128$ dimensions.

The normal SIFT is a texture descriptor, it considers only the grayscale pixels of a multimedia document (e.g; image); it therefore includes no information on the colors of regions of interest. To overcome this drawback, approaches trying to add to the description of SIFT descriptions of colors, which can benefit both the power of SIFT in texture and have color information. Rescently, van de Sande *et al.* [2008] proposed a color-based SIFT descriptors (OpponentSIFT, W-SIFT, rgSIFT, Transformed color SIFT), which incorporate color information, and compared them to other existing SIFT approaches (HSV-SIFT (Bosch *et al.* [2008]) and HueSIFT (van de Weijer & Gevers [2005])). The best results were obtained using the rgSIFT and Opponent-SFIT descriptors compared to the original SIFT for image classification.

2.3.2 Audio Descriptors

Earlier researches in multimedia indexing have focused on the use of visual features for classification and summarization. Recently, researchers have begun to realize that audio characteristics are equally, if not more, important when it comes to understanding the semantic content of a video. For example, when interesting events occur, people are likely to talk, laugh or cry. So, these events can be detected easily using the audio content rather than the visual content.

Representing an audio signal can be achieved by using several audio features, from which many are based on the automatic speech recognition (ASR) methods. ASR is reliable and largely indicative for the topic of videos. A semantic video retrieval approach using audio analysis is presented by Bakker & Lew [2002], in which the audio can be automatically categorized into semantic categories, such as explosions, music, speech, etc. Usually, characterization is done through audio sampling in different sizes. Short samples are suitable for speech recognition, while for analyzing the semantic content of an audio signal, the larger the samples are, the better. Generally, the audio features are grouped in two main categories (Wang *et al.* [1999]): temporal coefficients, which are calculated directly from the audio signal, and frequency coefficients. The spectral coefficients are directly derived from a Fourier transform; they describe the spectrum of an audio signal and are often used for the semantic analysis of audio (Lu *et al.* [2001]; Zhang & Kuo [1998]). These coefficients are commonly used for the characterization and recognition of speakers. Among them, the MFCCs (Mel Frequency Cepstrum Coefficients) have the advantage of considering the non-linear properties of the human perception for sounds according to several frequencies. MFCCs are commonly used as features in speech recognition systems, such as the systems which can automatically recognize numbers spoken into a telephone. They are also common in speaker recognition, in which the goal is to recognize people through their voices (Ganchev *et al.* [2005]).

Among the temporal characteristics, volume is the most widely used. The mean volume of a video does not necessarily reflect on anything; however, the temporal variation of volume often provides important features of a video. Easily calculated, it is a good indicator of silence, and useful for audio segmentation (Wang *et al.* [1999]). The zero-crossing rate, which has been used heavily in both speech recognition and music information retrieval, is very effective in distinguishing dialogues from music. The voice is characterized by low volume and a high zero-crossing rate. Finally, the pitch - a fundamental frequency of the audio signal - is a major auditory attribute of musical tones, which makes it an important parameter for the analysis and synthesis of sound and music. Normally, only the voice and music pitch have harmonic characteristics; the pitch is often used to characterize other sound types (Wang *et al.* [1999]).

2.3.3 Descriptor Aggregation

Feature descriptors generated from the aforementioned description methods, can be processed directly by some applications for the problem to be solved. For example, by using video color histograms as descriptors (represented as vectors), we can measure the similarity of two shots by the similarity of the corresponding vectors. This can be further calculated in different ways, such as the Euclidean distance. Then, given a query in video retrieval systems, all the video shots in the database are ranked according to their distance to the query. However, a further step needs to be applied, the aggregation. This has to be applied, particularly, when using the local feature as descriptors. Normally, a video shot (or an image) can contain many interest points, so that it is not convenient for the video retrieval algorithms to process directly. Descriptor aggregation is a process of clustering similar feature descriptors into classes. As a result, shots can be described by their membership - a single number - of descriptors, instead of the actual high dimensional values. Recently, a very effective method for visual feature aggregation, denoted as the bag of visual words (BoVW), was presented first by Sivic & Zisserman [2003]. It has become very popular and showed its effectiveness in the field of image classification and indexing (Fei-Fei & Perona [2005]; Jiang *et al.* [2007]; Schmid [2006]; Sivic & Zisserman [2003]; van de Sande *et al.* [2008]). Moreover, Liu *et al.* [2010] proposed the use of bag of audio (BoA) to characterize each audio frame.

In the following, we briefly review two popular approaches that are used to represent an image (or video shot) by aggregating a set of local descriptors. Namely, the bag of visual words (BoVW) and the Fisher kernel.

2.3.3.1 Bag of Visual Words

The bag of visual words (BoVW) approach is derived from the BoW approach in text analysis, wherein a document is represented by word frequencies and not regarding to their order. These frequencies are then used to perform document classification. Therefore, it is necessary to identify the visual equivalent of a word before the method can be applied to images. This is commonly done by extracting and quantizing local invariant features. As we have showed, in previous section, the motivation behind the use of local invariant features especially the SIFT, in the following, we will describe how they are transformed into visual words.

Most local feature detectors result in a large number of interest points. This density is important for classification robustness, However, it illustrates a representation challenge particularly since they are of high dimensionality (e.g. SIFT has 128 dimensions). The BoVW can be regarded as a classification problem, in which the membership of each feature is to be determined. K-means (MacQueen [1967]) algorithms are widely used as clustering methods for image and video descriptor aggregation, especially with BoVW. It takes the descriptors as input and creates a set of clusters. Hence, each cluster centroid is denoted by an indicator to a visual word, and a set of visual words is then called visual vocabulary *Codebook*.

We adopt a standard approach, which was given by Sivic & Zisserman [2003], to summarize the SIFT descriptors by quantizing and aggregating the features without regard to their location. The analogy to representing a text document by its word count frequencies is made possible by labelling each 128 dimension SIFT feature as a visual word. We apply standard K-means clustering to a large number of SIFT features (e.g. 10 millions) to create a dictionary of visual words. This visual dictionary is then used to quantize the extracted features by simply assigning the label of the closest cluster centroid. The final representation for an image is a histogram (frequency counts) of the labelled SIFT features.

$$BoVW = [w_1, w_2, \ldots, w_N]$$

where N is the dictionary size, w_i is the number of occurrences of visual word i in the image. To account for the difference in the number of interest points between images, the BoVW histogram can be then normalized in different way. Furthermore, Lazebnik *et al.* [2006] proposed to use the spatial pyramid techniques. In which, the fundamental idea is to partition the feature space into a sequence of increasingly coarser grids and then compute a weighted sum over the number of matches that occur at each level of resolution. More recently, Albatal [2010] proposed to group the vocabularies (BoVWs) to form visual phrases, which are then used in the classification.

2.3.3.2 Fisher Kernel

The basic idea behind the Fisher kernel method (Jaakkola & Haussler [1999]) is to train a generative model (e.g. hidden Markov model (HMM)) on data to derive a Fisher kernel for discriminative classifiers, such as support vector machine (SVM). The Fisher kernel gives a natural similarity measure that takes into account the underlying probability distribution. If each data item is a (possibly varying length) sequence, each may be used to train a HMM, with the average of the models in the training set used to construct a global HMM. It is then possible to calculate how much a new data item would ”stretch” the parameters of the existing model. This is achieved by, for two samples, calculating and comparing the gradient of the log-likelihood of the data item, with respect to the model with a given set of parameters. If these ”Fisher scores” are similar it means that the two samples would adapt the model in the same way. Many works have been shown the effectiveness of the use of Fisher kernels, in the literature (Fine *et al.* [2001]; Jaakkola *et al.* [2000]; Jaakkola & Haussler [1999]; Perronnin & Dance [2007]; Smith *et al.* [2001]; Vinokourov & Girolami [2001]).

We are interested in the work of Perronnin & Dance [2007], in which the authors applied the Fisher kernel framework to the visual vocabulary in the context of image classification. They model the visual words with a Gaussian mixture model (GMM), which models the generative process of the low-level feature extracted from images, restricted to diagonal variance matrices for each of the k components of the mixture. Deriving a diagonal approximation of the Fisher matrix of a GMM, they obtain a representation of $(2d+1)\times k-1$ dimensional vector of an image feature set, or $d \times k$ dimensional when considering only the components associated with either the means or the variances of the GMM. The Fisher kernel representation was shown to extend the BoVW: it is not limited to the number of occurrences of each visual word but it also encodes additional information about the distribution of the descriptors. In comparison with the BoVW representation, fewer visual words are required by this more sophisticated representation.

2.3.4 Description Normalization

The main goal of feature normalization, is to independently normalize the feature components, for instance, in such a way that their values lie within a similar range (e.g. $[0, 1]$ range). The normalization is often done by either using the component values of each vector independently (e.g. the L_1 and L_2 normalization), or by normalizing the vectors using their components values (e.g. min-max normalization). However, other normalization techniques work directly on the values independently, and it does not consider any of the other related values (e.g. power normalization). We will give more details on several normalization techniques in section 3.3.

2.3.5 Dimensionality Reduction

In general, algorithms of dimensionality reduction of descriptors aim to project the data in a high-dimensional space into a sub-dimensional space, formed by a statistical analysis of the training samples. For this, several methods exist. The goal is to transform the original space into a new space; the dimension reduction is then obtained by keeping only the dimensions that contain the maximum information. It should be noted that this may lead to a loss of information, so it is not always recommended to use such algorithms. In the following, we give an overview of some techniques, which are used for dimensionality reduction of descriptors in multimedia indexing.

Principal Component Analysis The principal component analysis (PCA) is the main linear technique of dimensionality reduction; it performs a linear mapping of the data to a lower dimensional space in such a way that the variance of the data in the low-dimensional representation is maximized. PCA was invented by Pearson [1901]. Let $(X_1, X_2, \ldots, X_p)$ be the initial components in space p, PCA aims to find the components $(C_1, C_2, \ldots, C_k)$ so that:

$$C_k = \alpha_{1k}X_1 + \alpha_{2k}X_2 + \ldots + \alpha_{pk}X_p$$

C_k is a 2×2 uncorrelated matrix, it has maximum variance and it is ordered by decreasing variance. Geometrically, this transformation forms a new space whose axes follow the extensions of the main cloud of the points formed by the analysed vectors.

In the information retrieval community, a popular technique to perform this transformation is achieved by applying Singular Value Decomposition (SVD) of the variance-covariance matrix, based on the eigenvector decomposition. Hare & Lewis [2005] used SVD for image annotation using semantic propagation. The premise of their approach is based on the perception that visually similar images often have similar meaning or semantics. SVD decomposes a $m \times n$ matrix A into the product of a $m \times r$ matrix T, a $r \times r$ matrix S, and a $r \times n$ matrix D:

$$A = TSD^T$$

So that $TT^T = DD^T = D^Td = I$, where I is the identity matrix. S is a diagonal matrix, in which diagonal elements are called singular values of matrix A, in monotonically decreasing order. It is proved that the k largest singular values, together with the corresponding left and right eigenvectors, encode the most important information of A (Deerwester *et al.* [1990]). Therefore, A is usually approximated by A^* (i.e. $A = A^*$), which is thought to contain less noise or even be noise-free:

$$A^* = T_kS_kD_k^T$$

Thus, the axes of the greatest variance correspond to the eigenvectors associated with higher values. The new basis of reduced dimension is finally formed by the k-eigenvectors of the largest eigenvalue. There are no methods to select automatically the number k; typically, it is decided a priori or it is selected by thresholding the eigenvalues.

Independent Component Analysis Whereas PCA only considers the moments of the second degree to reduce the dimension space, independent component analysis (ICA) considers higher order moments to make independent components (Jutten & Herault [1991]). The goal of ICA is to minimize the statistical dependence between the analyzed vectors. The key difference between PCA and ICA is that ICA does not reduce the number of dimensions, but makes them independent, which may be useful in Bayesian classifiers, which assume that the presence (or absence) of a particular feature component is unrelated to the presence (or absence) of any other feature component. However, as a complement to PCA, ICA allows to obtain independent components in a reduced dimensional space.

Linear Discriminant Analysis Another statistical method used for dimensionality reduction is the linear discriminant analysis (LDA). In contrast to PCA, LDA takes into account the organization of data into classes. Thus, LDA maximizes the interclass variance without increasing the total variance of the analyzed descriptors. A dimension reduction by LDA often increases the performance of classifiers. M.H. Song & Yoo [2005] showed that the performance of their classification system had better accuracy after the analysis of the data by LDA than by PCA.

2.3.6 MPEG-7 based Multimedia Indexing

With the large scale introduction of digital images and video to the market (Chang *et al.* [2001]). in 1997 the ISO MPEG Group initiated the MPEG-7[1] work item, formally named ”Multimedia Content Description Interface”. MPEG-7 is known as 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. The ultimate objective of MPEG-7 is to provide standardized descriptions of both real-time and archived audiovisual data. Such media descriptions can be used to identify, categorize, compare, filter, or browse image or video purely based on no-text visual descriptions of the content, or if required, in combination with common text-based queries (Chang *et al.* [2001]). Because of their descriptive features, the challenge for developing such MPEG-7 Visual non-text descriptors is that they must be meaningful in the context of various applications. They will be used differently for different user domains and different application environments.

To achieve the goals that MPEG-7 aims to standardize, the following components (which forms the six parts of MPEG-7 standards (Chang *et al.* [2001]; Martinez *et al.* [2002]; Nack & Lindsay [1999]):

MPEG-7 Systems, which defines the architecture standard. For example, the scheme of preparing MPEG-7 description to achieve efficient transport/storage and to allow synchronization between content descriptions.

MPEG-7 Description Definition Language (DDL), which is a standard language for defining new or extending description schemes and descriptors. MPEG-7 DDL has chosen to adopt XML Schema Language with some MPEG-7 specific extensions, to meet the MPEG-7 requirements.

[1] More details about MPEG-7 can be found at: http://www.chiariglione.org/mpeg.

MPEG-7 Visual, which contains the basic structures and descriptors to represent the basic visual features of multimedia data. For example color, texture, shape, motion and location.

MPEG-7MPEG-7 Audio, which contains the audio descriptions of multimedia data. MPEG-7 Audio comprises six technologies: the audio description framework, sound effect-, instrumental timbre-, spoken language content-, uniform silence segment- and melodic-description tools.

MPEG-7 Multimedia Description Schemes, which aims to standardize a set of description tools (descriptors and description schemes) to produce a generic description of multimedia data (including audio, visual, and text).

MPEG-7 Reference Software, is also called as the experimentation model, which is the simulation platform for the MPEG-7 descriptors, description schemes, coding schemes, and DLLs.

As shown in Figure 2.7, MPEG-7 describes multimedia content by its structural aspects and semantic aspects. Semantic description scheme emphasizes on describing semantic entities such as objects, events, semantic concepts, etc. in narrative world. A narrative world is the context in which the description makes sense, which may cover the world depicted in the multimedia data.

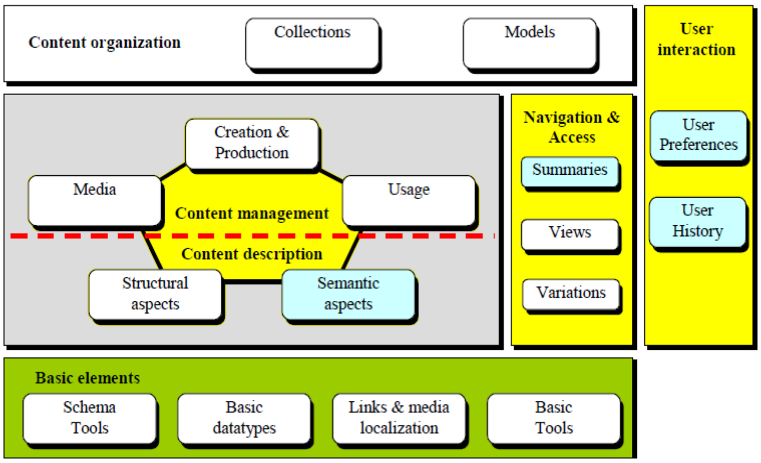


Figure 2.7: MPEG-7 multimedia description schemes components, from (Chang *et al.* [2001]).

Semantic description schemes can be used to form the description of abstractions which refer to the process of using a description for a single instance of multimedia content to be generalized into a set of multiple instances of multimedia content or a set of specific descriptions. There are two abstractions, namely media- and standard- abstraction. *Media abstraction* is a description that describes multiple instances of similar multimedia content. An example is how a TV news program description can be applied to describe other TV programs. *Standard abstraction* is generalization of media abstraction that describes a general class of semantic entities of descriptions. The general approach is to replace specific objects, events or other semantic entities by classes (Chang *et al.* [2001]). For example, ”Lionel Messi scores a goal in the soccer match between Barcelona and Manchester United during a Champions league" can be replaced by "A soccer player scores a goal in a soccer match". Hence standard description supports instantiation of a description template.

Data scheme of MPEG-7 has already considered many alternatives methods to describe video such as a trees, graph, hierarchical and sequential models(Abdel-Mottaleb & Krishnamachari [2004]). However, the semantic graph can be used to organize semantic annotations.

2.4 Classification Approaches

Classification is the process, in which similar entities are combined comprehensively into different classes; thus, a class is defined as the collection of similar entities. For example, books may be classified by their titles, authors or publishing year. In computer science, the term classification refers to processing this task automatically. Essentially, it is based on learning; the aim is to learn the relations between the target class and the features of each sample. Consequently, it needs some examples to learn these relations; we refer to this as a learning set. There are mainly two types of learning algorithms: the supervised and the unsupervised. In supervised learning, each example in the training set is a pair, which contains an input sample (e.g. a low-level feature) and a target output value (e.g. label). The learning algorithm analyses the training set to produce classifiers, which should predict the usefulness of any valid input sample. The learning is based on minimizing the error of the classification on the training set. A wide range of supervised learning algorithms is available, and there is no single learning algorithm which works best on all supervised problems. The second type of classification algorithms, the unsupervised learning, refers to the problem of finding hidden structures in unlabeled data. Since the data are unlabeled, there is no error to be minimized. This is useful in data clustering problems; the most used algorithm of this type is K-*means*.

For the supervised learning algorithms, there are two kinds of models: the discriminative and the generative models. Generative models specify a joint probability $P(x, y)$, which consists of x, the low-level feature vector and its associated label y. A common method used for estimating the probabilities is maximization of the likelihood of the data to containe the target concept. Then, Bayesian rule can be used to determine the most likely class. Some examples of these methods are the Gaussian models, the Bayesian networks and the Markov models. These methods have advantages: they can learn from partially annotated data, and they can be used in incremental learning, in which adding new examples to the training set does not need retraining the whole data. On the other hand, the discriminative models are used for modeling the dependence of an unobserved variable y on an observed variable x. This is done by modeling the conditional probability distribution $P(y|x)$, which can then be used for predicting y directly from a given x. Discriminative models generally yield superior performance for classification and regression tasks that do not require the joint distribution. Vapnik [1998] gave a good reason why to use discriminative methods: "one should solve the (classification) problem directly and never solve a more general problem (class-conditional) as an intermediate step".

Generative models are, typically, more flexible than discriminative models in expressing dependencies in complex learning tasks. However, they need often more time for training than the discriminative models (Ulusoy & Bishop [2005]). Furthermore, it is shown that for many classification tasks, such as multimedia indexing, discriminative models perform better than generative models (Iyengar & Nock [2003]; Jebara [2004]; Naphade [2004a]). Therewith, the rise of kernel methods has benefited the discriminative models; an example is the use of the Support Vector Machines (SVM). The idea behind this is to use special kernels to overcome the problem of non-linearly separated data; this is done by projecting the initial data in a higher dimensional space, in which the problem can be solved linearly. It is also important to note that, recently, hybrid algorithms for supervised learning have been proposed to realize some benefits of generative and discriminative models (Bellili [2001]; Li *et al.* [2005]; Prevost *et al.* [2005]; Raina *et al.* [2003]).

In the following, we present some supervised learning approaches commonly used in multimedia indexing.

2.4.1 K-Nearest Neighbors Approach

The K-nearest neighbors approach (KNN) is one of those algorithms that is very simple to understand but works incredibly well in practice. It is considered as the most natural classification model; it is part of the discriminative approaches because it directly assesses the class of a document from its low-level features (Cover & Hart [1967]). KNN is a lazy algorithm; hence, it does not use the training data points to do any generalization. In other words, there is no explicit training phase or it is very minimal. This means that the training phase is quite fast. However, the lack of generalization means that KNN keeps all the training set. More exactly, this approach is based on memorization of the training examples, which contrasts with other probabilistic approaches. The number K decides how many neighbors influence the classification and it is defined, based on the distance metric. If $k = 1$, then the algorithm is simply called the nearest neighbor algorithm.

This approach has several problems: it is resource intensive; it requires a lot of memory to store the samples; and it is a very time consuming classification, given the need for each document to calculate its distance to all the examples. Moreover, the accuracy of the algorithm degrades in the presence of noisy data. This makes it difficult to generalize. One solution is to first select a subset of non-noisy features (Blum & Langley [1997]), which enhances the system accuracy and speeds up the calculation of distances. Although it is quite obvious that the accuracy may increase when K is increased, the computation costs also raise.

During the classification phase, a document is faced with all the stored samples, as a function of distance; the algorithm evaluates the similarity between the given document and all samples in the training set. Then, the document class is determined by linear combination, or voting classes, of the K nearest samples, weighted by their similarity to the given document. The common similarity measure used is the Euclidean distance; however, according to the representation of the features, more appropriate similarity functions can be used. In particular, the case of $K = 1$ simply assigns for each test sample the class of its unique nearest sample from the training set. In other words, let x be the sample to be labeled, find its closest sample (y); 1NN assigns the label of y to x. It turns out that 1NN is a reliable technique. It is also recommended to compare the results of new learning algorithms with those of 1NN, because its average performances, in many contexts, are constant and often good (Jain *et al.* [2000]).

2.4.2 Support Vector Machine Approach

The Support Vector Machine (SVM) is a very popular and powerful learning technique for data classification. The basic idea of SVM is, given a set of data samples that belongs to one or two classes, SVM finds hyperplane(s) that separate(s) perfectly the d-dimensional data (into its two classes) by a clear gap that is as wide as possible, thus maximizing the distances of both classes to the hyperplane. The SVM was invented by Cortes & Vapnik [1995], and was used for text classification. Since then, the approach became one of the most popular and used approaches in classification and indexing tasks, especially in the information retrieval and computer vision communities. Because of its high generalization performance, it has been introduced to the image and video community in which the visual features are usually of very high dimensionality. One of the earliest researches that applied SVM to image classification was Chapelle *et al.* [1999].

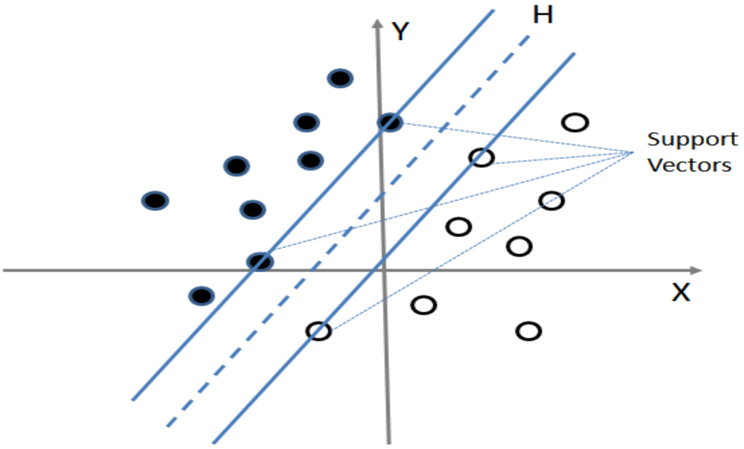


Figure 2.8: Linear separation in a two-dimensional space.

Maximum-margin hyperplane and margins for SVM are trained with samples from two classes. Samples on the margin are called the support vectors. Figure 2.8 shows an instance of linear SVM applied to data classification in a two-dimensional space. H denotes the hyperplane, which separates the black and white samples. However, since data samples are often not linearly separable, SVM's introduce the notion of a ”kernel induced feature space”, which maps the data into a higher dimensional space where the data can be separable. This can be done by using a *kernel tric* (Scholkopf & Smola [2001]). Typically, the kernel function is based on the similarity of the samples (descriptors) that provides more information on a given class than the values of these descriptors themselves.

When searching the hyperplane separator, the use of Lagrange multipliers leads to the selection of a subset of support vectors that define the hyperplane (i.e. the support vectors). This process has great benefits, as the support vectors simplify and accelerate the first phase of the indexing (i.e. the modeling), in which only the subset of support vectors affects the indexing of new samples.

For a binary classification problem, given a training data set T of size n:

$$T = \{(x_i, y_i) | xi \in \Re^d, y_i \in \{1, -1\}\},$$

where x_i and y_i indicate respectively the training vector and the target value of the i^{th} sample in T, and $i = 1, \ldots, n$. The classification hyperplane is defined as:

$$\langle w, \Phi(x) \rangle + b = 0$$

where $\Phi(\cdot)$ is a mapping from $\Re^d$ to the (usually) higher dimension Hilbert space H, and $\langle \cdot, \cdot \rangle$ denotes the dot product in H. Thus, the decision function $f(x)$ is:

$$f(x) = sign(\langle w, \Phi(x) \rangle + b)$$

The goal of the SVM's is to find an optimal hyperplane with the maximum margin between the two predefined classes. This can be obtained by solving the following quadratic optimization problem:

$$\overset{\min}{w, b, \xi} \quad \frac{1}{2} \|w\|^2 + C \sum_{i=0}^{n} \xi_i$$

$$\overset{subject\ to: \quad y_i(\langle w, \Phi(x) \rangle + b) \geq 1 - \xi_i}{\xi_i \geq 0, \forall i = 1, \ldots, n}$$

This is done by defining a mapping $z = \phi(x)$ that transforms the d dimensional input vector x into a (usually higher) d dimensional vector z. We aim to choose a $\phi()$ so that the new training data $\{\phi(x_i), y_i\}$ are separable by a hyperplane. The important observation here is, that every time a $\phi(x_i)$ appears, it is in a dot product with some other $\phi(x_j)$. That is, if we know the formula (i.e. kernel) for the dot product in the higher dimensional feature space:

$$K(x_i, x_j) = \phi(x_i)\dot{\phi}(x_j)$$

we would not have to deal with the mapping $z = \phi(x_i)$ directly. The most popular kernel is the Radial Basis Function (RBF), which is also called Gaussian kernel, defined by:

$$K(x_i, x_j) = Exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

where $||\dot{}||$ is the L_2 norm, x_i, x_j are two different input vectors, and σ is a tunable Gaussian parameter that can be fixed by cross-validation. This leads to a symmetric matrix called kernel matrix or ”Gram matrix”, which indicates the similarity between each pair of input vectors. In principle, only similarity functions that lead to a kernel matrix satisfying the *Mercer* conditions (positive-definite eigenvalues) can be used.

However, some studies have successfully used dissimilarity functions that do not satisfy the Mercer conditions. For instance, Zhang *et al.* [2006] used the Earth Mover Distance (EMD) instead of the Euclidean distance in a Gaussian kernel. The authors could not provide evidence that this kernel satisfies the Mercer conditions in all cases. However, a high classification performance rate was reported through their experiments.

2.4.3 Ensemble Learning

An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way (typically by weighted voting) to classify new instances. It is referred to by different names in the literature (i.e. committees of learners, mixtures of experts, classifier ensembles, multiple classifier systems, consensus theory, etc.) (Kuncheva & Whitaker [2003]). The strategy in ensemble systems is to create many classifiers, and combine their outputs in such a way that the combination improves upon the performance of a single classifier. The intuition is that, if each classifier makes different errors, then a combination of these classifiers may reduce the total error. The overarching principal in ensemble systems is therefore, to make each classifier as unique as possible. Specifically, we need classifiers whose decision boundaries are adequately different from those of others. Such a set of classifiers is said to be diverse (Dietterich [2000]; Polikar [2006]).

Classifier diversity can be achieved in several ways. The most popular method is to use different training datasets to train individual classifiers. Such datasets are often obtained through re-sampling techniques, such as bootstrapping or bagging, where training data subsets are drawn randomly, usually with replacement, from the entire training data. Another way to achieve diversity is to use different training parameters for different classifiers. Adjusting such parameters allows one to control the instability of the individual classifiers, and hence contribute to their diversity. Furthermore, diversity can also be achieved by using different features, this is mostly used in image and video classification, as researchers usually combine the predicting scores of classifiers trained on several features.

There have been many methods developed for the construction of ensemble learners. Some of these methods, such as Bagging, Boosting and Random Forest, can be applied to any base learner. In the following, we present some ensemble approaches that we have used and compared to other classification methods used in this thesis.

Bagging Bagging was proposed by Breiman [1996] to improve the classification by combining classifications of randomly generated training sets. The name is derived from bootstrap aggregation, and was one of the first effective methods of ensemble learning. It is also one of the most intuitive and simplest to implement, with a surprisingly good performance (Breiman [1996]). Diversity in bagging is obtained by randomly drawning - with replacement - different subsets as training data from the original training data. Such a training set is called a bootstrap replicate of the original set. Each training data subset is then used to train a different classifier of the same type. Individual classifiers are then combined by taking a majority vote of their classification decisions (scores). For any given data sample, the ensemble decision is defined by the class chosen by most classifiers. Bagging can be used with any type of model for classification or regression. Neural networks and decision trees are good candidates for this purpose, as their instability can be controlled by the selection of their free parameters. Bagging is, typically, applied to learning algorithms that are unstable, where a small change in the training set can cause a significant change in the model.

Boosting Schapire [1990] proved that a weak learner, based on an algorithm that generates classifiers, which merely does better than random guessing, can be turned into a strong learner that generates a classifier to classify new unlabeled samples well. This is officially done by *boosting* the performance of a weak learner to the level of a strong one. The algorithm is now considered as one of the most important developments in the recent history of machine learning. Boosting creates an ensemble of classifiers by re-sampling the data, similar to Bagging, in which classifiers are then combined by majority voting. However, similarity ends here. In boosting, re-sampling is applied to provide the most informative training data for each consecutive classifier. The training set can be sampled according to weight distributions to produce a new training set, to be used by the next learner.

Freund & Schapire [1997] has introduced the *AdaBoost* approach, which is a more general version of the original boosting algorithm. AdaBoost maintains a set of weights over the training samples; in each iteration i, the classifier C_i is trained to minimize the weighted error on the training set. The weighted error of C_i is computed and used to update the distribution of weights on the training samples. This distribution update ensures that samples, misclassified by the previous classifier, are more likely to be included in the training data of the next classifier. Thus, the weights of correctly classified samples are decreased and the weights of misclassified samples are increased. Hence, consecutive classifiers' training data are geared towards increasingly hard-to-classify instances, and the process is repeated.

For example, when boosting three classifiers (C_1, C_2 and C_3), the process is done as follows: the first classifier C_1 is trained on a randomly chosen subset from the available training set and maintains a set of weights over the training samples. The second classifier C_2 is trained on samples chosen as the most informative subset, with the updated distribution obtained by C_1. The third classifier C_3 is trained with samples on which C_1 and C_2 disagree. Finally, in order to give the final score for each sample, the three classifiers are combined through a majority vote.

Random Forest The Random Forest approach was introduced by Breiman [2001]; he combined bagging with random feature selection for decision trees. A random forest can be created from individual decision trees, whose certain training parameters vary randomly. Such parameters can be bootstrapped replicas of the training data, as in bagging, but they can also be different feature subsets as in random subspace methods. Dietterich [2002] recommends random forests as a method to choose for decision trees, as it compares favourably with *AdaBoost* and works well even with noise in the training data.

Stacking Generally, certain data samples may have a high likelihood of being misclassified because, for example, they are very close to the decision boundary. Contrarily, certain samples may have a high likelihood of being correctly classified, because they are primarily far away from their respective decision boundaries. A question arises: can we learn that certain classifiers consistently classify correctly or misclassify certain data samples? (Polikar [2006]). Stacking (stacked generalization) was proposed by Wolpert [1992], and it works on combining the models rather than choosing among them, thereby typically getting a performance better than any single one of the trained models. Thus, in stacking an ensemble of classifiers is first created, whose outputs are used as inputs to a second level meta-classifier, to learn the mapping between the outputs of the ensemble and the correct class. In multimedia indexing, stacking is widely used as data fusion, as it is a good strategy for fusing the scores of classifiers, which are learned on different multi-modalities.

2.5 Fusion

As a video document can be represented by different sources (i.e. low-level features), such as visual, audio and text, a major step in multimedia indexing is fusion. Data fusion is the study of efficient methods, which automatically transform information from different sources into a representation that provides effective support for decision making. Bloch *et al.* [2003] give a general definition of information fusion: ”Information fusion is to combine information from multiple heterogeneous sources to improve decision making”.

This definition allows to a large variety of data fusion problems. In the case where data come from automatic processes, it is important that the merging phase is able to handle uncertain or missing data. Typically, uncertainty leads to two problematic situations as given by Bloch *et al.* [2003]: the ambiguity reflects the ability of information to conduct other information, and the conflict characterizes the conflict interpretations between two or several information.

Furthermore, in automatic multimedia indexing, the fusion can be applied at two levels of indexing: low-level fusion (denoted as *early fusion*), where the fusion is a combination of different low-level descriptors, and high-level fusion (denoted as *late fusion*), where the fusion is done by combining the outputs of several classifiers. The different classifiers can be deployed on a single or on different descriptors. Moreover, a third way of data fusion is the *kernel fusion*, where the uni-modal kernels are combined in order to create a multi-modal kernel.

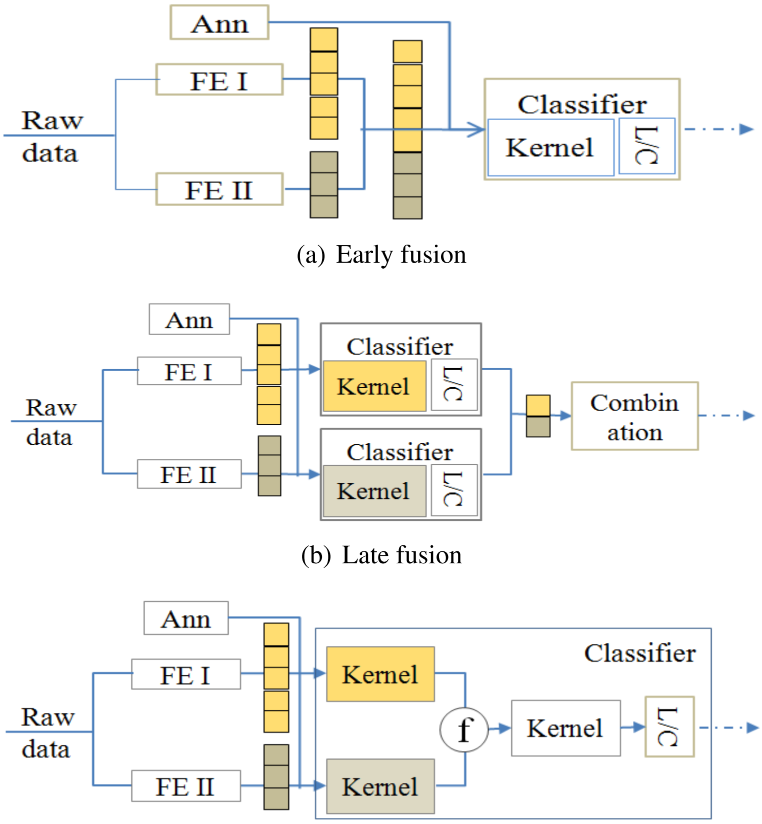
2.5.1 Early Fusion

Early fusion is a fusion algorithm that gathers uni-modal descriptors into a multi-modal representation (i.e. one larger descriptor) before learning a concept. After combining the descriptors in a multi-modal representation, early fusion methods rely on supervised learning to classify semantic concepts (Naphade [2004b]; Snoek *et al.* [2005b]). In practice, early fusion is based on concatenation of feature vectors from different media (e.g. visual, audio) or descriptor types (e.g. color, texture) in a single vector of larger dimension. The scheme of early fusion is shown in figure 2.9(a).

Early fusion yields a truly multimedia feature representation, since the features are integrated from the beginning. The simplicity of its implementation (i.e. simple concatenation) makes it popular in multi-modal indexing. Indeed, it requires only one learning phase. Moreover, the concatenation of many feature vectors may lead to a large space, in which the learning phase may not be converged. In such cases, it is necessary to reduce the number of features in the final vector, typically by applying dimensionality reduction algorithms, such as PCA.

2.5.2 Late Fusion

Late fusion is a fusion scheme that combines prediction scores of different classifiers that were learned on uni-modal descriptors, to yield a final classification score. Indexing approaches



(a) Early fusion

(b) Late fusion

(c) Kernel fusion

Figure 2.9: General schemes of data fusion. In all schemes (L/C) refers to Learning and Classification, from Ayache *et al.* [2007].

that rely on late fusion also start with extraction of uni-modal features. In contrast to early fusion, approaches for late fusion learn semantic concepts directly from uni-modal features. Late fusion focuses on the individual strength of modalities. Uni-modal concept classification scores are fused into a multi-modal semantic representation rather than a feature representation.

The advantage of late fusion is, that it allows the use of a specific method of classification for each modality, which takes into account the specificity of each classifier and each modality. The decisions from these classifiers are then combined into a smaller space. Thus, the complexity of learning is reduced, compared to early fusion. However, the multi-modal combination yields a much smaller dimension that greatly limits the use of correlations between the features from each modality. In addition, the combination of several classifiers provides more accurate decisions, since the individual classifiers do not make the same prediction errors. A big disadvantage of late fusion is its expensiveness in terms of the learning effort, as every modality requires a separate supervised learning phase. Moreover, the combined representation may require an additional learning phase. Another disadvantage of the late fusion approach is the potential loss of correlation in mixed feature spaces. Figure 2.9(b) shows an instance of the late fusion scheme.

Late fusion has been presented in several formalisms. One of the most used method is done by applying the majority vote on the outputs of the classifiers. Another method is 'stacking', which uses another classifier to combine the outputs of weak classifiers. This is also called "Meta-Classifiers" (Lin *et al.* [2002]). The authors have learned an SVM on the output of two classification systems, one for face recognition and the other for speech identification. The meta-classifier can then identify a person more accurately than the two uni-modal classifiers. They also showed that using meta-classifiers for data fusion is more effective than using the majority vote. Wu *et al.* [2004] classified concepts in a video corpus regardless of visual and sound modalities. As output, a kernel matrix called "super kernel" was created and learned using SVM with RBF kernel. The authors compared their approach with a linear combination and fusion product, and showed that fusion by classification is more accurate.

2.5.3 Kernel Fusion

The Kernel combination is an active topic in the field of machine learning and data fusion. It takes benefits of Kernel-based classification algorithms. The fundamental advantage of merging modalities at the kernel level is, to allow choosing the kernel functions according to the modalities. For instance, color-histograms can take advantage of specific histogram matching distances, while textual modality can be categorized using more appropriate kernels, such as String Kernels (Lodhi *et al.* [2002]).

Kernel fusion also allows modeling the data with more appropriate parameters. Merging modalities using an early fusion scheme leads to model the data using a single kernel function. Consequently, when using a Gaussian kernel (i.e. RBF), a single σ parameter is expected to properly fit the relations of sample vectors, whereas it makes much more sense to train a combined RBF kernel using one σ per modality. Combination of uni-modal kernels leads to keep as much information as possible from each modality. A combined RBF kernel has the following form:

$$K_c(x, y) = F(K_m(x_m, y_m)_{(1 \leq m \geq M)})$$

where $K_c(x, y)$ is the combined kernel value for the samples x and y, $(K_m)_{(1 \leq m \geq M)}$ are the considered uni-modal RBF kernels, F is the combining function over the M modalities, x_m and y_m are the sample vectors for modality m. The kernel fusion process is illustrated in figure2.9(c), the uni-modal kernels are merged using a fusion function in order to create the multimodal kernel. Then, the learning and classification steps aim to assign a classification score to each of the video items. One of the main issues in the current kernel research is the learning of such combined kernels. It is denoted as *Multiple Kernels Learning*, and aims to learn, at the same time, the parameters of the all uni-modal kernels and the parameters of the combining function (Gönen & Alpaydin [2011]; Sonnenburg *et al.* [2006]).

Ayache *et al.* [2007] used a very simple strategy to create combined kernels. Their algorithm assumes that the best parameters of uni-modal kernels are suitable enough to allow efficient generalization of the combined kernel.

2.6 Active Learning

Active learning (sometimes called ”query learning” in the literature) is a sub-field of machine learning and, more generally, artificial intelligence. The key hypothesis is that if the learning algorithm is allowed to select the data from which it learns, it will perform better with less training samples. Active learning can be applied in different applications: i) corpus annotations, where the goal is to lead the process of annotating unlabeled corpus; ii) training classification systems, where its objective is to select a small set of the most useful samples (from the training set) to train a classification model; iii) searching documents, the goal is to learn models from the user feedback, in order to enhance the retrieval result (e.g. relevance feedback). The classifier with an active learning algorithm is called a *learner*, and an human annotator or expert who gives the correct labels is called an *oracle*.

For multimedia corpus annotation, the task is to assign relevant labels or tags to multimedia documents. Active learning systems attempt to overcome the labeling bottleneck by querying samples from unlabeled instances to be labeled by an oracle. In this way, the active learner aims to achieve high accuracy using as few labeled samples as possible, thereby minimizing the cost of obtaining labeled data. Active learning is well-motivated to be used in many modern machine learning problems, where data may be abundant but labels are sparse or expensive to obtain. Machine learning techniques allow us to create new rules and classification techniques from the labeled and sometimes unlabeled training data for the development of generic algorithms with a wide range of applications. Most of the current techniques, which we showed in the aforementioned classification methods, are passive in nature, in which the learning function does not interact with the training data and works only on the given data.

In multimedia indexing, the amount of unlabeled data is abundant while the labeled data needed for training an accurate classifier, is hard to acquire, as labeling can be an expensive and time consuming task. Active sampling is an alternative approach to automatic learning: given a pool of unlabeled data U, one tries to select, in an active way, a set of training samples to reach the best performance with a minimum number of labeled data. Ideally, the same performance is achieved as when labeling all the training samples, with a smaller number of manual labeled samples.

Given a pool of unlabeled data U and a given concept c to be labeled, an active learning function F assigns a real value to each unlabeled sample as likelihood to contain an instance of c, using the following formula: $F(x_i) \to \Re, x_i \in U$. Based on this criterion, we can rank the unlabeled samples and select the most informative samples ($\tilde{X}$) to be annotated by an oracle, according to F:

$$\tilde{X} \approx \arg\max_{x_i \in U} F(x_i)$$

Thus, after labeling the most informative samples $\tilde{X} \in U$ and adding it to the training set, the knowledge and performance of the classification system will be improved the most. The optimal active learning function selects the most informative samples, which improve the system performance when adding them to the training set. Therefore, the performance of the classification system can be improved by enlarging the current training set through iteratively adding selected samples.

The performance of a learner A is usually measured on an independent test set, as a function of an actively selected training set, which generates a learning curve. This is denoted as *simulated active learning* (Ayache & Quénot [2007a]). The resulting learning curve is usually compared to the learning curve based on random sampling from U. A good query function should at least outperform random sampling during the learning process. Moreover, a satisfied performance should be reached with small fraction of labeled data.

There are several scenarios in which active learners may query, as well as several query strategies (i.e. sampling) that have been used to decide which samples are most informative, thus need to be labeled. In the following, we describe some active learning scenarios and query strategies.

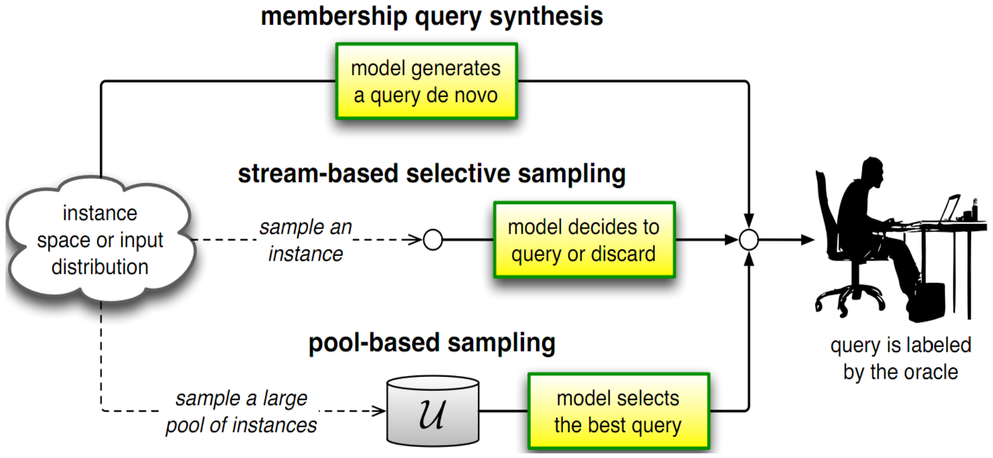

membership query synthesis
model generates a query de novo
stream-based selective sampling
instance space or input distribution
sample an instance
model decides to query or discard
pool-based sampling
sample a large pool of instances
$\mathcal{U}$
model selects the best query
query is labeled by the oracle

Figure 2.10: Diagram illustrating the three main active learning scenarios, from Settles [2009].

2.6.1 Active Learning Scenarios

There are several different problem scenarios in which the learner is able to ask queries. The three main settings that have been considered in the literature are (i) membership query synthesis, (ii) stream-based selective sampling, and (iii) pool-based sampling. Figure 2.10 illustrates the differences among these three scenarios.

Membership Query Synthesis The scheme of active learning with membership queries (Angluin [1988]), is one of the first active learning scenarios. In each iteration, the learner generates an instance in an input space and requests its label. There is no predefined set of unlabeled samples. Efficient query synthesis is often tractable and efficient for finite problem domains (Angluin [2001]).

Stream-Based Selective Sampling An alternative to synthesizing queries is selective sampling (Atlas *et al.* [1989]; Cohn *et al.* [1994]; Freund *et al.* [1997]; Seung *et al.* [1992]). The key assumption is that obtaining an unlabeled instance is inexpensive, so it can first be sampled from the actual distribution, and then the learner can decide whether or not to request its label. This approach is sometimes called *stream-based active learning*.

Pool-Based Sampling For many real-world learning problems, large collections of unlabeled data can be gathered at once. This motivates *pool-based* sampling (Baram *et al.* [2004]; Lewis & Gale [1994]; McCallum & Nigam [1998]; Roy & Mccallum [2001]; Tong & Koller [2002]), which assumes that there is a small set of labeled data L and a large pool of unlabeled data U available. Given an unlabeled pool U, an active learner (AL) has three components: (L, A, Q) where A is a classifier, $A : X \leftrightarrow \{-1, 1\}$, trained on labeled instances L, and Q is the selection (querying) function that, given L, decides which samples in U to query next. Queries are selectively drawn from the pool, which is usually assumed to be fixed. Typically, samples are selected according to an informativeness measure used to evaluate all the samples in the pool. The pool-based scenario has been studied for many real-world problem domains in machine learning, such as text classification (Cohn *et al.* [1996]; Hoi *et al.* [2006]; Lewis & Gale [1994]; McCallum & Nigam [1998]; Roy & Mccallum [2001]; Tong & Koller [2002]), information extraction (Settles & Craven [2008]; Thompson *et al.* [1999]), image classification and retrieval (Gosselin & Cord [2008]; Tong & Chang [2001]; Zhang & Chen [2002]), video classification and retrieval (Hauptmann *et al.* [2006]; Yan *et al.* [2003]) and speech recognition (Tur *et al.* [2005]).

In this thesis, we focus on the pool-based sampling approach. Essentially, the active learning algorithm consists of two main steps. First, a small number of labeled samples and a pool of unlabeled samples are collected. Then the querying function (i.e. selective sampling process) and the retraining of a learner are adopted. The general active learning algorithm can be summarized in algorithm 1. Assume that an initial labeled set L is given, a pool of unlabeled data U, a classifier A, and a query function (of the active learning) Q, then a set of unlabeled samples $\tilde{X} \in U$ is actively selected according to the criterion of Q, and proposed to an oracle to be labeled. After their labels are obtained, the training set is enlarged and the classifier A is retrained. In practice, there are four crucial issues in developing an active learning system:

Algorithm 1 General Active Learning Algorithm

```
Input: initial small labeled set $L$, a pool of unlabeled data set $U$,
and the selection function $Q$ .
repeat
    Train classifier $A$ on $L$
    $P_{un} \leftarrow$ Predict($U$,$A$)
    Select the most informative samples $\tilde{X} \in U$ according to $Q$
    Ask oracle(s) to label $\tilde{X}$
    Add $\tilde{X}$ to $L$
    Remove $\tilde{X}$ from $U$
until The predefined breaking criterion is achieved (e.g. $U = \phi$)
```

The cold start: This issue is defined as the construction of the initial training set, which will bootstrap the active learner. Traditionally, the initial training data set is generated randomly, based on the assumption that random sampling is likely to build an initial training set with data distribution similar to that of the whole corpus. Zhu *et al.* [2007] have proposed a technique of sampling by clustering. The cold-start problem is considered as one of the future perspectives of this thesis.

The learning algorithm: This issue is mainly related to a classification issue, in which the performance varies when using different classifiers. It is difficult to find out which classifier is better for a given problem. Recently, researches have focused on the use of support vector machines (SVMs) as classifiers.

The stopping criterion: In principle, defining a stopping criterion is a problem of estimating the effectiveness of a classifier during active learning (Lewis & Gale [1994]). Actually, defining an appropriate stopping criterion for active learning is a trade-off issue between labeling cost and effectiveness of the classifier. However, in some cases the annotation process is limited to a time schedule. For example, the production of the annotations for the development sets of TRECVid is limited to less than one month. Thus, the challenge is to produce as many effective annotations as possible, in a given scheduled time.

The selective sampling scheme: The third issue of active learning is how to select the most informative sample for human annotation at each learning iteration. In the next section, we introduce some of the most used selective sampling (i.e. querying) strategies.

2.6.2 Querying Strategy

The main difference between an active learner and a regular passive learner is the querying function Q. This brings us to the issue of how to choose the next unlabeled samples from the unlabeled pool. All active learning scenarios evaluate the informative unlabeled samples, depending on different querying (i.e. selection) strategies. The number of the selected samples to be labeled can be given as constant (e.g. fixed number at each iteration) or as a function of the size of the labeled set. As the labeled data increase, the number of samples to be labeled grows.

There have been many proposed ways in the literature to formulate such selection strategies. In the following, we give three general strategies that are widely used.

Uncertainty Sampling Perhaps the simplest and most commonly used query framework is uncertainty sampling (Lewis & Catlett [1994]). In this method, the queried instances are those, about which the active learner is uncertain about their labels. This approach is often straightforward for probabilistic learning models. For example, when using a probabilistic model for binary classification, uncertainty sampling simply queries the samples whose probability of being positive is nearest 0.5 (Lewis & Catlett [1994]; Lewis & Gale [1994]).

Query-By-Committee Another more theoretically-motivated query selection framework, is the query-by-committee (QBC) algorithm (Seung *et al.* [1992]). The QBC approach involves maintaining a committee $C = \theta^1, \ldots, \theta^c$ of models, which are all trained on the current labeled set L, but represent competing hypotheses. Each committee member is then allowed to vote on the labels of query candidates. The most informative queries are considered to be the samples about which the committee members most disagree. The aim of QBC is to find the version space, which is the set of hypotheses that is consistent with the current labeling set L. Thus, by searching the best model within the version space, the goal of using active learning is to constrain the size of this space as much as possible with as few labeled instances as possible.

Relevance Sampling This strategy selects the most probable positive samples and tries to maximize the size of the set of positive instances. This strategy is very useful in multimedia indexing, where positive samples are most often sparse within the whole set, and finding negative samples is inexpensive.

2.7 Evaluation of Video Indexing Effectiveness

Once the shots in a video test corpus are automatically annotated by indexing systems, annotation qualities need to be assessed for performance comparisons between different systems. For this, many evaluation metrics have been used by researchers; in the following, we introduce some of these metrics. TREC has provided an evaluation tool, denoted as *Trec_eval*, which calculates several measures to compute the quality of concept detection and indexing, among others the Recall, Precision and Mean Average Precision.

The most popular evaluation metrics for comparing different information retrieval systems are the precision and recall. These metrics are also widely adopted for evaluating the effectiveness of auto-annotation approaches in the information retrieval community. In this community, the precision of a query is defined as the ratio of the number of relevant documents that are returned by the system to the total number of documents returned, whereas recall is defined as the ratio of the number of relevant documents returned to the total number of relevant documents in the database.

$$precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|}$$

$$recall = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|}$$

These two values do not contain all the information needed to compare all systems. Theoretically, the evaluation must be based on curves showing the precision as a function of the recall. However, it is also necessary to have unique values for automatic comparison, thus, other measures based on recall and precision are proposed:

P(10), P(30), P(N$_r$) : measure the precision achieved in the top $10, 30, N_r$ retrieved documents.

R(10), R(30), R(N$_r$) : measure the recall achieved in the top $10, 30, N_r$ retrieved documents.

Mean Average Precision (MAP) : measures the non-interpolated average precision.

Inferred Average Precision (InfAP) : measures the inferred average precision.

Among these measures, the MAP and InfAP measures have the advantage of summarizing the recall-precision curve in a single value. They have been widely used as the official measure of several image and video retrieval campaigns, such as TRECVid and Pascal-VOC. The MAP is defined by the following formula:

$$MAP = \frac{1}{R} \sum_{j=1}^{S} \frac{R_j}{j} \times I_j$$

where R is the number of relevant video shots in a corpus containing S shots. Let L be the sorted list of retrieved documents, at each index j, R_j is the call after j shots are retrieved, and I_j is equal to 1 if document j is relevant, otherwise it is equal to 0.

InfAP measure was proposed by Yilmaz & Aslam [2006] to be used as the evaluation metric for TRECVid 2006.

2.7.1 TRECVid Evaluation Campaign

Since 2001, the TREC VIDEO evaluation campaign offers participants the means to experiment with different approaches of concept detection in documents videos. Initiated by National Institute of Standards and Technologies (NIST), the TRECVid campaign has aimed to promote scientific progress in the field of content search and indexing of video documents. By making available a corpus of annotated videos and assessment tools, TRECVid also allows participants to compare themselves, which in some way limits the number of exploratory work: as soon as it is part of a logic competition, all participants will tend to approach that produced the best results.

In 2001, the corpus consisted of 11 hours of videos, which were collected from free sources (i.e. The Internet Archive, Open Video Project). For a first task, only the video shot segmentation and search tasks were put into competition. The search task was then based only on low-level descriptors, and the text from the ASR. It was then found that these methods of querying are not effectively respond to the proposed queries. Thus, for 2002, organizers have created the first concept detection task (10 concepts), which could then be used to the researching task.

Collection	Length (hours)	Dataset	Concepts	Shots	Data description
2007	≈ 100	dev	36	21532	cultural, news magazine, historical, science news, news reports,documentary, and education programming supplied by the Netherlands Institute of Sound& Vision
		test	20	22084	
2008	≈ 200	dev	53	43616	
		test	20	42461	
2009	≈ 380	dev	22	43616	
		test	20	115313	
2010	≈ 400	dev	130	119685	high degree of diversity of internet videos from the Internet Archive Creative Commons (IACC)
		test	30	146788	
2011	≈ 600	dev	346	266473	
		test	50	137327	

Table 2.1: The latest TRECVid databases.

The corpus was enriched with 23 hours of videos for the development part and five hours for the test set. In 2003, the sets of development and testing have been enhanced to include 60-hour newscasts each chained from U.S. information (CNN, ABC) and parliamentary debates (CSPAN). The variety of topics is assumed larger than the present year earlier videos. The task of detecting concepts was to identify 17 concepts. For this effort was placed on a collaborative phase of manual annotation by concepts. Thus, NIST has provided manual annotations for 133 local concepts. In 2004, the corpus has doubled in size, the development and test sets of 2003 were grouped together to form the development 2004. In 2005, a new effort annotation was conducted, and a new corpus of 170 hours of international television news programs (United States, China and Lebanon) has been available to researchers. In total, 40 concepts were manually annotated at the global level. They were drawn from all LSCOM[1] (Large-Scale Concept Ontology for Multimedia), which proposes an ontology of hundreds of concepts adapted to the description of multimedia documents (Naphade *et al.* [2006]). The choice of a corpus into different languages has been motivated to encourage teams to focus on methods of concept through the visual modality. Indeed, the text quality was largely diminished because in addition to a phase of automatic transcription of speech, it was necessary to add a step of machine translation into English. Moreover, the extraction task was the identification of 10 concepts.

Finally, similar to the 2004 campaign, in 2006 the corpus has doubled in size while maintaining the corpus as a set of development in 2005. This time 39 concepts were to be identified, of which only 20 were evaluated by NIST. In table 2.1, we summarize the evolution of the TRECVid collections from the year 2007 till 2011. As the table shows, the TRECVid dataset has largely extended, in both the data size and the number of the evaluated concepts. It has reached 600 hours of video data in 2011, with 346 annotated concepts in the development set and 50 concepts for the evaluation. In 2007-2009 TRECVid provided participants with cultural, news magazine, documentary, and education programming supplied by the Netherlands Institute for Sound and Vision. In 2010 and 2011 TRECVid provided a new set of videos, from the Internet Archive Creative Commons (IACC[2]), characterized as is common in much ”web video” by a high degree of diversity in creator, content, style, production qualities, original col-

[1] http://www.lscom.org.

[2] http://www.archive.org/index.php

lection device/encoding, language, etc. Table 2.2 presents the evaluated concepts of the latest collections of the TRECVid campaigns, the '*' indicates the common concepts between 2010 and 2011. In chapter 6, we will give more details on the collection of TRECVid 2011.

Thanks to TRECVid, it is now possible to evaluate methods for processing, analysis, classification and information retrieval in large video collections. Research teams around the world working for various purposes on videos of around hundreds of hours. All experiments that we present in this thesis were conducted on TRECVid corpus and for the concept detection task (called High Level Feature Extraction (HLF) or Semantic Indexing (SIN)).

2.7.2 TRECVid Semantic Indexing Evaluation Task

The semantic indexing task (SIN)(Over *et al.* [2010]) is as follows: Given a standard set of shot boundaries for the semantic indexing test collection and a list of concept definitions, participants were asked to return for each concept in the full set of concepts, at most the top 2000 video shots from the standard set, ranked according to the highest possibility of detecting the presence of the concept. The presence of each concept was assumed to be binary, i.e., it was either present or absent in the given shot. If the concept was true for some frame (sequence) within the shot, then it was true for the shot. This is a simplification adopted for the benefits it afforded in pooling of results and approximating the basis for calculating recall.

Year	Concepts
2007	1.Sports 2.Weather 3.Office 4.Meeting 5.Desert 6.Mountain 7.Waterscape_Waterfront 8.Police_Security 9.Military 10.Animal 11.Computer_TV-screen 12.Flag-US 13.Airplane 14.Car 15.Truck 16.Boat_Ship 17.People-Marching 18.Explosion_Fire 19.Maps 20.Charts
2008	1.Classroom 2.Bridge 3.Emergency_Vehicle 4.Dog 5.Kitchen 6.Airplane_flying 7.Two_people 8.Bus 9.Driver 10.Cityscape 11.Harbor 12.Telephone 13.Street 14.Demonstration_Or_Protest 15.Hand 16.Mountain 17.Nighttime 18. Boat_Ship 19.Flower 20.Singing
2009	1.Classroom 2.Chair 3.Infant 4.Traffic-intersection 5.Doorway 6.Airplane_flying 7.Person-playing-a-musical-instrument 8.Bus 9.Person-playing-soccer 10.Cityscape 11.Person-riding-a-bicycle 12.Telephone 13.Person-eating 14.Demonstration_Or_Protest 15.Hand 16.People-dancing 17.Nighttime 18.Boat_Ship 19.Female-human-face-closeup 20.Singing
2010	1.Airplane_Flying 2.Animal 3.Asian_People 4.Bicycling 5.Boat_Ship 6.Bus 7.Car_Racing 8.Cheering 9.Cityscape 10.Classroom 11.Dancing 12.Dark-skinned_People 13.Demonstration_Or_Protest 14.Doorway 15.Explosion_Fire 16.Female-Human-Face-Closeup 17.Flowers 18.Ground_Vehicles 19.Hand 20.Mountain 21.Nighttime 22.Old_People 23.Running 24.Singing 25.Sitting_Down 26.Swimming 27.Telephones 28.Throwing 29.Vehicle 30.Walking
2011	1.Adult 2.Anchorperson 3.Beach 4.Car 5.Charts 6.Cheering* 7.Dancing* 8.Demonstration_Or_Protest* 9.Doorway* 10.Explosion_Fire* 11.Face 12.Female_Person 13.Female-Human-Face-Closeup* 14.Flowers* 15.Hand* 16.Indoor 17.Male_Person 18.Mountain* 19.News_Studio 20.Nighttime* 21.Old_People* 22.Overlaid_Text 23.People_Marching 24.Reporters 25.Running* 26.Scene_Text 27.Singing* 28.Sitting_down* 29.Sky 30.Sports 31.Streets 32.Two_People 33.Walking* 34.Walking_Running 35.Door_Opening 36.Event 37.Female_Human_Face 38.Flags 39.Head_And_Shoulder 40.Male_Human_Face 41.News 42.Quadruped 43.Skating 44.Speaking 45.Speaking_To_Camera 46.Studio_With_Anchorperson 47.Table 48.Text 49.Traffic 50.Urban_Scenes

Table 2.2: The evaluated concepts of the latest TRECVid collections.

2.8 Summary

In this chapter, we have presented the state-of-the-art of approaches, which are used in multimedia indexing systems, and the active learning approaches that are used to build development sets of multimedia databases. The related work is introduced throughout the thesis as it is appropriate. Active learning algorithms attempt to overcome the labeling bottleneck by selecting the samples to be labeled by an oracle. They aim to increase the system performance using as few labeled samples as possible, thereby minimizing the cost of labeling data (e.g. money and time). Furthermore, active learners are classification systems that learn interactively from the oracles' feedback. The efforts toward building multimedia indexing systems based on active learning, consist of several phases: multimedia document description, classification, fusion techniques and the active learning scenarios and their selection strategies.

We have started by presenting the CBMIR systems, in which we have discussed the major components for multimedia indexing, including the video document, structuring, segmentation, description, classification, etc. The characterization of video documents can be achieved in several levels of abstraction. The low-level descriptors are not sufficient to achieve a semantic description of multimedia documents. Nevertheless, they support the supervised classification algorithms to achieve a semantic description. There are several low-level descriptors, which can be extracted from video documents, such as visual and audio descriptors. The choice of low-level descriptors has an important role in the indexing phase. Therefore, it is recommended to use descriptors that discriminate the different modalities of a video. In this thesis, the indexing system was evaluated using various video characterization techniques, which have been produced by several partners of the IRIM project of GDR-ISIS (Quénot *et al.* [2009]). Moreover, these descriptors were tested with different normalization techniques.

We have shown that, bridging the semantic gap (i.e. passing from low-level to semantic descriptions), is necessarily achieved through a classification phase. We have presented different kinds of classification approaches, including the generative and discriminative algorithms. In information retrieval, the generative algorithms were the first used methods for document classification. However, many works in multimedia indexing have shown that the best performance is obtained when using discriminative approaches rather than generative methods. Specifically, the support vector machines (SVM) have been widely used in many research areas that require supervised learning methods, and they often lead to better performances. Mainly by representing the data using a kernel space, the non-linearly separable problems can be easily solved. Hence, the SVM has became very popular, and many implementations are available on the internet. We have chosen to use the libSVM (Chang & Lin [2001]), which is an open source library. Besides the SVM, we have also reviewed some other used methods, such as KNN and the ensemble learning. We have shown that KNN is a good approach to be used as a baseline approach, because it calculates fast and it gives coherent performance results in multimedia indexing. The ensemble learning methods are also good methods to index large datasets, especially in the case of sparse concepts. The idea in ensemble systems is to create many classifiers; the final decision in such systems is obtained by combining the outputs of the classifiers in the ensemble. However, in this thesis we have developed an ensemble approach, which provides diversity in a different way than the popular methods of creating ensembles, such as Bagging, Boosting, etc. Moreover, our ensemble approach can work on any kind of classifiers. This approach is detailed in the next chapter, section 3.1.2.

Finally, We have presented three different scenarios of the use of active learning approaches, namely, (i) membership query synthesis, (ii) stream-based selective sampling, and (iii) pool-based sampling. We have also introduced three well-known strategies, which are used with the active learning approaches to select the new samples to be annotated. For the following of the thesis, we will focus on the third approach (i.e. the pool-based sampling), more over we will evaluate the active learning method using the different strategies of selecting the samples.

The methods, used in the thesis will be evaluated and validated on different collections of the well-known international benchmark, called TRECVid.

Chapter 3

Contributions to Multimedia Indexing and Retrieval

This chapter presents a variety of techniques and challenges regarding the Multimedia Semantic Indexing and Retrieval. It proposes a novel technique based on multi-learner (ML) approach for image and video classification. Moreover, it proposes new techniques for improving the classification performance, including an approach for re-ranking samples in video retrieval system and the descriptor normalisation. Evaluation experiments on the proposed methods are also presented. This chapter ends with some conclusions.

3.1 Multi-Learner Approach for Class-Imbalance Problem

In many particular domains of machine learning, the class imbalance problem became a hot topic, especially in those related to multimedia databases. As mentioned before, a dataset is called imbalanced if at least one of the classes is represented by a significantly smaller number of samples than the others. Imbalance is well-known to be a serious issue in designing classifiers (Chawla *et al.* [2004]; Japkowicz & Stephen [2002]). However, supervised learning algorithms that do not consider the class imbalance problem tend to be more influenced by the majority class and will probably ignore the minority class samples (Chawla *et al.* [2004]).

The class imbalance issue is closely related to the cost-sensitive learning, in which misclassifying minority class samples is usually more serious than misclassifying majority class samples. Breiman *et al.* [1984] pointed out four closely connected components in machine learning problems, which are the size of the training set, the class priority, the error cost in different classes, and the placement of the decision boundaries. Many existing methods, which deal with the class imbalance issue, rely on connections among these four components. For instance, two important classes of methods to handle the class imbalance issue are: the sampling methods, which handle class imbalance by varying the minority and majority class sizes in the training set; and the cost-sensitive learning methods, which deal with class-imbalance by giving different weights to the two classes (Weiss [2004]).

3.1.1 Related Work

Many existing methods for class-imbalance learning manipulate the four components which are pointed by Breiman *et al.* [1984]: the size of the training set, the class priority, the error cost, and the placement of the decision boundaries. In the following, we will review the sampling methods, which are the most widely used approaches for class-imbalance classification. For more complete and detailed review on the other methods, we refer to Weiss [2004].

Both under- and over-sampling methods change the training sets by, respectively, sampling a smaller majority training set and repeating samples in the minority training set. The imbalance level will be reduced in both methods, in the hope that a more balanced training set will lead to the building of more accurate classifiers. However, random under-sampling and over-sampling are the most popular non-heuristic methods that balance the data set respectively, through random replication of minority class sample and random elimination of majority class samples. Both methods are easy to implement but suffer from some limitations. For example, under-sampling can discard potentially useful data that could be important for the classification process, while over-sampling increases the training time and it may lead to over-fitting, since it makes exact copies of the minority class samples (Batista *et al.* [2004]).

Several heuristic methods are proposed in the literature to overcome these limitations. Tomek links (Tomek [1976]), Condensed Nearest Neighbour Rule (CNN) (Hart [1968]), One-sided selection (Kubat & Matwin [1997]) and Neighborhood Cleaning rule (NCL) (Laurikkala [2001]) are several well-known methods for under-sampling, while the Synthetic Minority Over-Sampling Technique (SMOTE) is a well-known method for over-sampling (Chawla *et al.* [2002]). The basic idea of SMOTE is to add new synthetic minority class samples by randomly interpolating pairs of closest neighbors of the minority class samples. Some other methods combine different sampling strategies to achieve further improvement (Batista *et al.* [2004]).

Liu *et al.* [2009] examine the class imbalance problem by combining classifiers built from multiple under-sampled training sets. Several subsets from the majority class are created, each subset having approximately the same number of samples as the minority class. One classifier is trained from each of these subsets and the minority class, after which the classifiers are combined. However, the authors used only weak classifiers, such as bagging, boosting and the random forest approaches.

An alternative approach is to randomly select a subset from the negative samples with a size comparable to that of all positive samples (Bishop [2007]). It is even possible to balance the loss of information related to sub-sampling of the negative class by making several selections on this class set and merging the outputs of different classifiers built from these subsets.

Tahir *et al.* [2009a,b] went deeper in the sub-sampling by taking subsets from the majority class, which are smaller in size than the minority class, thus reversing the balance between classes for each random selection. The method is called Inverse Random Under-Sampling (IRUS). A large number of random selections are made to compensate for this inverse sub-sampling, so that all elements of the majority class are finally used. The objective is to compensate for the natural tendency of the classification algorithms to favour the majority class. Due to the inverse proportion, it is the minority class which will be favoured again; this is what is required for the evaluation metric that gives more weight to the well-placed positive samples. This additional importance to the minority class is preserved when merging the results of elementary classifiers. However, the authors validated their approach with a single type of classifier, which is based on logistic regression.

3.1.2 Our Proposed Multi-Learner Approach based Classification

In this thesis, we consider only the binary classification problems by ensemble classifiers built using different under-sampling training sets. Our proposed method is based on under-sampling and the ensemble learning techniques to handle the class-imbalance problem in a large imbalanced dataset. We propose the following as our model for multimedia document classification (Safadi & Quénot [2010]):

$$ML = \langle X, Ann, Desc, Mono, f_{maj}, f_{min}, App, App_{param}, Fu, Eval \rangle$$

where:

- X indicates the collection of all the samples that will be processed by the model, i.e. the all samples of the considered collection. This collection is partitioned into two sets: L and U, which indicate respectively the training and the evaluating sets. The partition is done considering the following two conditions: $L \cup U = X$ and $L \cap U = \phi$.

- Ann is the set of the ground truths (i.e. the annotations) associated to the collection X. It contains one annotation for each video shot $x \in X$, thus $\{Ann = ann_x : x \in X\}$. The set of ground truths are generated manually (i.e. by oracles) and it is considered as the ideal annotation set of the collection X. These ground truths are done for each concept separately. Notice that Ann_L is the set of the annotation of L, and Ann_U is the set of the annotation of U.

- $Desc$ defines the model, which will be used to extract the feature vectors of the shots within the collection X. In other words, $Desc$ is the document description model, which extracts the low-level features (e.g. visual, audio) of the video shots.

- $Mono$ is a global parameter, which defines whether the system uses a mono- or a multi-learner approach. When $Mono = 1$ the system uses the mono-learner approach, otherwise the multi-learner approach will be used.

- f_{maj} is a factor that indicates the probability (e.g. number of times) of each sample in the majority class to be selected. Since, the system applies a random sub-sampling over the samples of the majority class, therefore, it is obvious that some samples will be selected in different subsets, and others will be never selected. By increasing the value of the f_{maj}, the system is more able to integrate more samples from the majority class. Furthermore, the combination of the majority class samples is increased.

- f_{min} is a factor, which defines the desired ratio between the majority and the minority class and will be used in sub-sampling a balanced set from the set L. Classifiers tolerate a moderate imbalance ratio between the classes, thus the optimal value of this parameter is obtained by the cross-validation on the classifiers, taking into account the complexity of the problem. When using the multi-learner approach, the f_{maj} and f_{min} parameters will be used to determine the number of subsets that will be generated from L, as given in section 3.1.2.1.

- App defines the supervised learning approach (e.g. logistic regression or SVM), which will learn the relations between low-level features and the annotations of a sub-sampled set.
- App_{param} defines the settings of the optional parameters of the classifier App (e.g. the parameter C in SVM). These parameters are passed to the designated classifier App.
- Fu defines the fusion function in case of the multi-learner approach. For each sample, this is done by merging its prediction scores from all the classifiers. Many options can be used as the fusion function, like arithmetic mean, minimum, maximum, etc. It can also be applied by adding an additional classifier, which learns the fusion rules and gives one score for each sample.
- $Eval$ indicates the evaluation metric, which measures the performance of the model. For instance, the common metric that is used in multimedia indexing is the mean average precision (MAP).

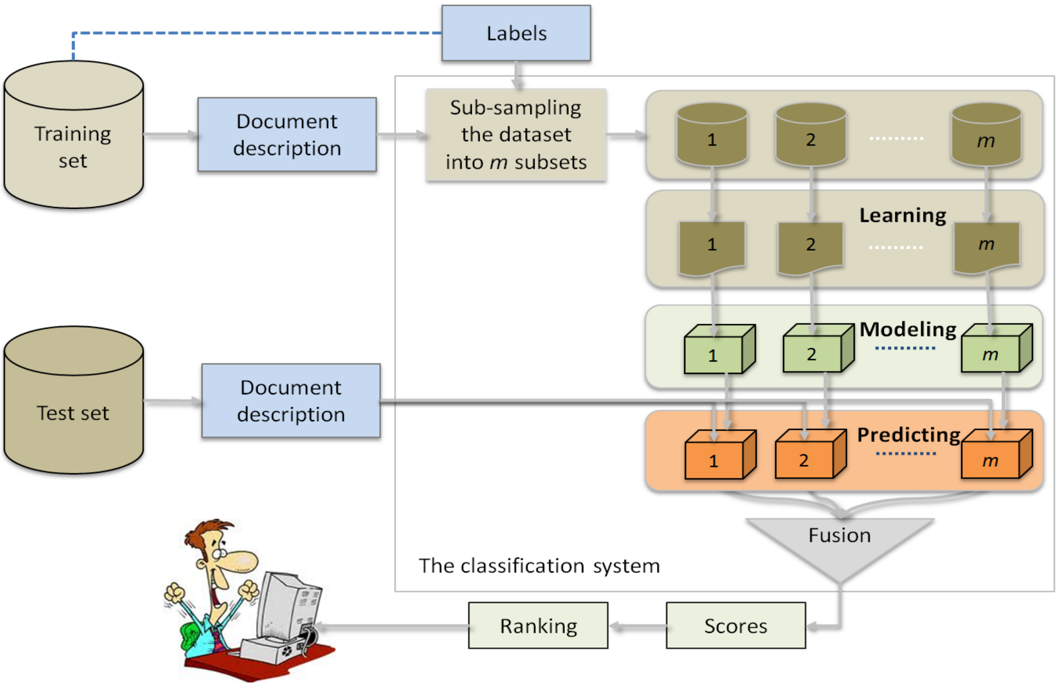


Figure 3.1: The framework of the multi-learner approach.

Figure 3.1 presents the framework of our multimedia indexing system with the multi-learner approach. The system is similar to the standard multimedia indexing system that we showed in the state of the art. The difference is in the classification part. For each concept c, the f_{min} and f_{maj} parameters are used to calculate the number of sub-sampled sets (m), Then the training set

(L) is sub-sampled into m subsets. Each subset contains all the minority class samples (nb_{min}) and $f_{min} \times nb_{min}$ randomly chosen samples (from L) of the majority class. One classifier is trained on each subset and generates a model, thus m different models will be generated. For each sample in the test set, these models are used to predict its likelihood to contain the concept c. A fusion function Fu is applied on the predicted scores and generates a final score for each sample. Lastly, the samples are ranked according to their scores, and the ranked list are then available to a user.

3.1.2.1 The Number of Subsets

According to our model, the number of the classifiers is equal to the number of the sub-sampled datasets; one classifier will be trained on each subset. Each subset, of which are provided to a classifier, contains all the minority class samples and a sub-sampled set of the majority class samples (randomly chosen). For the binary classification of a concept c, the system first calculates the number of samples of each class (i.e. positives and negatives) from Ann_L. Regarding to the number of samples in each class, the system defines the importance of the imbalance problem. If the the data is balanced then the system applies the ML model with $Mono = 1$, otherwise, it calculates the number of sub-samples (m) as follows:

$$m = \frac{f_{maj} \times nb_{maj}}{f_{min} \times nb_{min}} \tag{3.1}$$

where nb_{maj} and nb_{min} represent respectively the number of samples of the majority and minority class, and f_{min} and f_{maj} parameters are those in our ML model. For most concepts, the majority class refers to the negative samples, whereas the minority class refers to the positive samples.

In both the mono- and multi-learner methods, these two parameters (f_{maj} and f_{min}) are added to the hyper-parameters of the classifiers. For the f_{maj} parameter, higher values may produce slightly higher performance. However, higher values lead to higher computational costs as well, because more subsets are generated. Therefore, a robust choice can be considered by setting $f_{maj} = 1$, which means that, ideally, each sample in the majority class will be selected at least once. If we set the $f_{maj} = 1$, the only free parameter apart from the classifier of the two methods (i.e. the mono- and multi-learner), is the f_{min}. This parameter must be optimized together with the internal parameters of the classifier. This is done by cross-validation on the development set.

3.1.3 Discussions

The results presented in Tahir *et al.* [2009a,b] showed the benefits for the multi-learner approach with the IRUS method for image and video classification. However, the authors validated their approach, in the context of two commonly used video and image collection benchmarks, namely Mediamill (Snoek *et al.* [2005a]) and Scene (Boutell *et al.* [2004]) benchmarks, using a single type of classifier based on logistic regression. In practice, the decision function is a combination of a sigmoid function and a linear form on the descriptors' space. It is exactly the same type of functions that is used when transforming the output of a linear SVM classifier into a probability using the Platt method (Platt [1999]). It is also the function that corresponds to a single layer perceptron. The general formula of this function is:

$$f(x) = \frac{1}{1 + e^{w.x+b}}$$

Logistic regression and linear SVM in its probabilistic version use the same type of decision function, but they do the learning differently. The first one seeks an optimal function in one step to output directly the probabilities, while the second seeks first a linear form that maximizes the margin and then applies a sigmoid function optimized by cross-validation to produce the final probabilities. Each of these methods has advantages and disadvantages. The linear SVM theoretically minimizes the empirical risk by maximizing the margin. The logistic regression optimizes everything directly without cross-validation. Furthermore, it is much faster. In the case of SVM, it is possible to use a kernel function such as the radial basis function (RBF) to define a scalar product between two descriptor vectors after transforming them in another space by a non-linear function (we use a Gaussian kernel here):

$$K(x, y) =< \Phi(x), \Phi(y) >= e^{-\gamma \cdot \|x-y\|^2}$$

An SVM based on this type of scalar product, performs better than one based on a simple scalar product between descriptor vectors. The transformation of their outputs into probabilities is also possible using the Platt method. This type of generalization does not seem possible for the method based on logistic regression.

The transformation into probabilities is not mandatory in the multi-learner method, however, it performs a normalization of the outputs, which allows the use of simple and effective fusion functions. The multi-learner approach is then possible using logistic regression and SVM, with linear or non-linear kernels such as the Gaussian kernel.

3.1.4 Experiments

We have implemented the mono- and multi-learner methods on three classification approaches: logistic regression, SVM with linear kernel and SVM with Gaussian kernel. We evaluated two versions of these three methods in the concept detection task of TRECVid 2008 and 2009.For comparison, we have also implemented and evaluated a method of classification based on the K nearest neighbors (KNN) with mono-learner, but with adaptive balance based on the ratio between positive and negative samples.

One major difficulty in evaluating and comparing the methods, is that each method depends on a number of rather critical hyper-parameters, such as the coefficient C in the SVM, the γ in the SVM with Gaussian kernel or the λ in the logistic regression approach. These hyper-parameters are usually optimized by cross-validation on the development collection. The classification method is then applied with the corresponding values on the test collection for evaluation. We have used libSVM (Chang & Lin [2001]) for SVM and RS-TRIRLS (Komarek [2005]) for logistic regression. We have interfaced these two libraries so that they can work with our multi-learner approach.

The experiments were conducted in the context of the high level features detection tasks of the TRECVid 2008 and 2009 collections. Each collection has 20 concepts (i.e. features)

Descriptor	SRBF	MRBF	SLIN	MLIN	SLR	MLR
global_tlep	8	4	2	0.5	2	0.2
bow_sift_1000	8	4	4	1	2	0.2
global_qwm1x3	4	3	4	2	2	0.05
hg104	4	2	2	0.05	2	0.05
opp_sift_har	3	3	3	0.2	2	0.02

Table 3.1: Optimal values of the ratio between the number of negative and positive samples for different methods and on different descriptors; optimization was done on the TRECVid 2008 development set.

for evaluation, of which ten concepts are the same and ten are different (see table 2.1). The classification methods (i.e. App in our model) that are evaluated in these experiments, are denoted as follows:

- SRBF and MRBF: single- and multi-learner versions with SVM Gaussian kernel,
- SLIN and MLIN: single- and multi-learner versions of SVM with linear kernel,
- SLR and MLR: single- and multi-learner versions with logistic regression,
- SKNN: K nearest neighbors in mono-learning, taking into account all the negative samples.

3.1.4.1 Descriptors

We have evaluated the aforementioned classification methods with descriptors of different types and sizes, which have been produced by various partners of the IRIM project of GDR-ISIS (Quénot *et al.* [2009]). We have used five descriptors (i.e. $Desc$ in our model), namely, CEAL-IST/global_tlep, LEAR/bow_sift_1000, ETIS/global_qwm1x3, LIG/opp_sift_har and LIG/hg104. For more details on the descriptors please refer to Appendix B.

3.1.4.2 Hyper-parameters Optimization

The hyper parameters of the classification methods and the ratio (f_{min}) between the size of positive and negative samples, were optimized. This was done on the development collection of TRECVid 2008 with its associated annotations.

Table 3.1 shows the optimal values of the ratio between the size of the negative and positive sample subsets for the different methods × descriptor combinations. We observe that:

- For linear SVM (i.e. SLIN and MLIN) and LR methods (i.e. SLR and MLR), the optimal ratios are greater than 1 (with values ranging between 2 and 4) with the single-learner method, while it is generally less than 1 (with values between 0.05 and 0.2) for the logistic regression and between 0.05 and 2 for linear SVM in the multi learner approach. The values found in the case of logistic regression are consistent with those used in the IRUS approach of (Tahir *et al.* [2009a,b]).

Descriptor	SRBF	MRBF	SLIN	MLIN	SLR	MLR	SKNN
global_tlep	0.0667	0.0751	0.0319	0.0405	0.0368	0.0598	0.0678
bow_sift_1000	0.0489	0.0561	0.0237	0.0345	0.0231	0.0469	0.0467
global_qwm1x3	0.0561	0.0566	0.0348	0.0465	0.0369	0.0469	0.0608
hg104	0.0541	0.0596	0.0223	0.0310	0.0240	0.0481	0.0580
opp_sift_har	0.0651	0.0747	0.0485	0.0652	0.0486	0.0644	0.0621
Fusion	0.1022	0.1099	0.0786	0.0968	0.0784	0.0969	0.1160

Table 3.2: Performance of different methods on different descriptors and their simple late fusion score on the test set of TRECVid 2008

- For the SVM method with Gaussian kernel (i.e. SRBF and MRBF), the optimal ratios are significantly higher than LIN and LR approaches for both the single- and multi-learner approaches. This is probably due to the fact that the Gaussian kernel allows for a better definition of the boundary between classes and due to the fact that a learner based linear classification (e.g. LIN or LR) is more likely to find a good boundary with only a few samples. This suggests that the finding of a value less than 1 for the optimal ratios in LR and LIN approaches, may be related more to the chance of finding a good separation with fewer samples than offsetting a bias favoring the majority class.

These optimal values are fixed and used for the evaluation on the test collections.

3.1.4.3 Evaluation on the Test Sets

Table 3.2 shows the performance obtained for various method × descriptor combinations on the test collection of TRECVid 2008. We note that:

- The performance measurements are consistent with those measured on the development collection.
- The relative performance is generally quite stable against the different descriptors, even if their individual performance is variable.
- The multi-learner approach is consistently better than the single-learner approach.
- Logistic regression is often, but not always, better than the linear SVM, perhaps because the optimization in a single step is generally better than the optimization in two steps. It is also significantly faster in practice.
- The SVM with Gaussian kernel is better than the two other methods. The multi-learner approach is better than the single-learner approach, but not as significantly as the other two methods. This is probably because the single SVM learner with a RBF kernel already has the ability to make a non-linear and therefore more general separation between the classes.
- The method based on KNN performs well and is even better for one of the tested descriptors. It also leads to better fusion.

Descriptor	SRBF	MRBF	SLIN	MLIN	SLR	MLR	SKNN
global_tlep	0.0572	0.0608	0.0198	0.0274	0.0240	0.0490	0.0551
bow_sift_1000	0.0378	0.0427	0.0172	0.0203	0.0117	0.0331	0.0334
global_qwm1x3	0.0501	0.0559	0.0319	0.0439	0.0250	0.0416	0.0538
hg104	0.0512	0.0505	0.0143	0.0269	0.0151	0.0445	0.0479
opp_sift_har	0.0564	0.0664	0.0419	0.0575	0.0391	0.0628	0.0580
Fusion	0.0971	0.1105	0.0796	0.0967	0.0718	0.0864	0.0999

Table 3.3: Performance of different methods on different descriptors and their simple late fusion score on the TRECVid 2009 test set.

The same experiments were conducted on the TRECVid 2009 datasets. The results, on the test set, are reported in table 3.3. They are consistent with those obtained on the collection of TRECVid 2008, even if half of the concepts are different. The best fusion is obtained by SVM with Gaussian kernel using the multi-learner approach. From the reported results, one can say that the system has poor performances (i.e. very low MAP values), however, these results are acceptable for the video concept indexing tasks, especially in TRECVid. Hence, it reflects the challenge and difficulties of this task. We believe that having low MAP values is due to several reasons. The evaluated concepts are semantically difficult; the indexing is applied on large-scale and diverse video databases. Also, the MAP is calculated on the top 2000 shots out of the total number of shots of the test set. Therefore, low values can be obtained even when there are coherence numbers of relevant samples, which are retrieved at the top of the ranked lists. For instance, there are positive samples in the top 10 retrieved samples.

3.1.4.4 Time Processing

All the experiments were done on a machine with two quad-core processors running at 2.66 GHz and with 32 Gbytes of RAM. The execution time depends upon the descriptor and the learning method used. It also depends upon the sub-sampling ratio between the numbers of negative and positive samples given in table 3.1. Here we give some ideas about the execution time of the training process of our experiments:

- Obviously, the mono-learner approach is faster than multi-learner.
- The LR is faster than all the others; for the longest descriptor, the execution time is less than half an hour for the mono-learner method, while it is around two hours for the multi-learner method.
- The LIN and RBF have almost the same executing time, both are slower than the LR method. For the longest descriptor, the execution time is around 7 hours for the mono-learner method, while it is around 30 hours for the multi-learner method.
- In multi-learner experiments, the ratio between negative and positive samples has an effect on the execution time: when this ratio is less than 1, more learners will be generated, thus the execution time is increased.

- The KNN takes in total about 8 hours with the longest descriptor, but it processes all concepts at once (the computation of the nearest neighbors for a test sample is done only once for all concepts), while the time given for the other methods corresponds to the processing of a single concept.

3.2 Re-ranking Video Shots

In general, semantic indexing is achieved by supervised learning approaches, based on training classifiers on positive and negative samples of a target concept (the development set). The classifier will generate a model, which will be used to predict the likeliness of new samples (the test set) to contain the target concept. The likeliness is often computed homogeneously as a probability for each data sample to contain the concept. Retrieval can then be done by ranking the samples according to their probability scores. Such ranking is initially done using a score for each sample, using only information from the development set. It is often possible to improve the indexing and retrieval performance by re-ranking the samples, using the results of the initial ranking on the whole test collection. Thus, re-ranking may lead to the retrieval of more relevant samples at the top of the ranked lists.

3.2.1 Related Work

Recently, several methods have been proposed and developed for re-ranking. We review below some of these methods.

Context fusion (Jiang *et al.* [2007]; Liu *et al.* [2007a]): the results of different searching models (e.g. concept-based search model, text-based search model or query by example) are used to re-rank the ranked lists. In fact, the focus here is on the fusion of output scores of different models. This method trains new classifiers on new descriptors. Since in our work we also use the fusion of output scores obtained by multiple models, we consider this as a baseline approach.

Classification-based re-ranking (Kennedy & Chang [2007]): the initial results of a baseline system are used to discover the co-occurrence patterns between the target semantics and extracted features. This is very similar to the *"learning to rank"* method (Herbrich *et al.* [1999]), which is based on training a ranking model which can precisely predict the ranking lists in the dataset. Kennedy & Chang [2007] used the top-ranked and low-ranked samples as respectively pseudo-positive and pseudo-negative examples to train a new classification model for ranking. The classification margin for a target concept is regarded as its (new) re-ranked score. The use of SVM as the classification model, leads to the method called *RankSVM* (Herbrich *et al.* [1999]).

Ordinal re-ranking (Yang & Hsu [2008]): re-rank initial results are re-ranked by using the co-occurrence patterns via the ranking functions. For each sample, the final score is a weighted combination of its original score and its re-ranked score. The authors adopted a training method to train the re-ranking algorithm on some concepts; the re-ranking algorithm was applied to re-rank the remaining concepts.

In video collections, the retrieval units are often video shots, rather than the whole video. Our aim is to efficiently re-rank the video shots according to their initial scores, which were obtained from initial classifiers, and according to the video knowledge and nature. Our proposed method is similar to the one proposed by Wang & Merialdo [2009]. Their method re-ranks the initial results of shots using the video knowledge score, which was estimated by calculating the arithmetic mean on the initial scores of all shots within the same video.

This research goes further: the generalized mean rule was adopted to calculate a global score for each shot, depending on the knowledge obtained from the scores of its neighbors within the video. This has been proved to be more efficient. Moreover, we studied the effectiveness of the re-ranking when applied on homogeneous and non-homogeneous databases. Furthermore, two windowing functions, the Rectangular and the Gaussian, were used on the neighbors of each shot to calculate its global score.

3.2.2 The Proposed Re-ranking Method

In multimedia systems based on video retrieval, we need to rank the video shots according to an estimation of their relevance to what the user wants to see. This estimation can be the prediction score obtained by the trained model, which refers to the likeliness of a shot to contain a target concept. In general, in multimedia retrieval systems, the retrieved lists are ranked, but they may contain some irrelevant samples. Therefore, we apply a re-scoring method in order to minimize the error within these ranked lists.

The method we propose here considers three hypothesis: i) videos have rather homogeneous contents; ii) the presence of a given concept in a video depends on the nature of the video itself; iii) the estimated scores are computed independently for all video shots in the corpus. The proposed re-ranking method is done by re-scoring the video shots, which is done in two steps (Safadi & Quénot [2011a,b]). First, for each shot, we compute a global score z, which is calculated through the initial scores of its predefined neighbors within the same video. Then this global score will be used to re-evaluate the initial score of each shot.

Let the test collection consist of a set of videos $V = (v_1, v_2, \ldots, v_m)$, m being the number of videos in the collection. Each video v_i is composed of a sequence of shots $v_i = (s_{i1}, s_{i2}, \ldots, s_{in_i})$, n_i being the number of shots of v_i. For each shot s_{ij}, an initial classification score x_{ij} is computed from supervised learning on the development set.

Many ways, including arithmetic mean, minimum, maximum, geometric mean, harmonic mean and root mean square, are possible to compute a global score z_{ij} for the shot x_{ik} in video v_i from its neighboring shots. We consider the formula of a generalization mean rule (equation 3.2), to be the method to calculate the global scores of each shot in the video, since all the above mentioned methods can be inherited from this rule by evaluating different parameters of α.

$$z_{ij} = \left(\frac{\sum_k f_\theta(j,k)(x_{ik})^\alpha}{\sum_k f_\theta(j,k)} \right)^{1/\alpha} \tag{3.2}$$

where x_{ik} indicates the score of shot k in video i, α defines the used function, which has to be tuned by cross-validation. Hence, different values of α lead to different functions, such as: Min: $\alpha = -\infty$; Max: $\alpha = \infty$; Geometric Mean: $\alpha = 0$; Arithmetic Mean: $\alpha = 1$; Harmonic Mean: $\alpha = -1$ and Root Mean Square: $\alpha = 2$. $f_\theta(j,k)$ works as a window around the current shot j, and defines its neighboring shots in the $video_i$. Two kinds of windowing functions are considered: the rectangular ('hard') and the Gaussian ('soft'). In both cases, the size of the window is defined by a parameter θ. For the rectangular window, the number of neighbors of each shot in video i is given by $2\theta + 1$. For the Gaussian window, we have applied $\sigma = \sqrt{\theta(\theta+1)/3}$. In this way, both windowing functions have the same variance for the same value of θ. This θ parameter has also to be tuned within the training set. Moreover, there are two critical values of θ: $\theta = 0$ and $\theta = \infty$. $\theta = 0$ gives the baseline, which corresponds to the initial ranking. $\theta = \infty$ uses a global score of the video itself, which is calculated from all the shots belonging to the same video, in other words ($z_{ij} = z_i$).

After these global scores z_{ij} are calculated, the score of each shot is updated according to its previous score and its global score obtained from the video knowledge. Again, many options are possible for the score fusion. We choose a weighted multiplicative fusion:

$$x'_{ij} = x_{ij}^{1-\gamma} \times z_{ij}^{\gamma}, \tag{3.3}$$

where γ is a parameter that controls the ”strength” of the re-ranking method. However, this parameter has to be tuned by cross-validation within the development collection as well.

3.2.3 Experiments

In this section, we present our experiments in which we have evaluated the proposed re-ranking method on the semantic indexing task. The experiments were conducted on the TRECVid 2008 and 2010 databases. Each database consists of two large sets: the development and the test set. Table 3.4 shows general information about these two databases. The TRECVid 2010 development set (2010d) consists of 119685 shots of 3173 videos with an average of 37 shots per video, and the test set (2010t) consists of 146788 shots of 8467 videos with an average of 17 shots per video; it seems that videos in this database are homogeneous. The TRECVid 2008 development set (2008d) consists of 43616 shots of 219 videos with an average of 199 shots per video, and the test set (2008t) consists of 42461 shots of 219 videos with an average of 193 shots per video; these videos are not homogeneous.

Collection		Concepts	Shots /Videos	Min /Mean /Max
2008	dev	20	43616 / 219	19 /199 /1003
	test	20	42461 /219	14 /193 /1029
2010	dev	130	119685 /3173	1 /37 /1381
	test	30	146788 /8467	1 /17 /1423

Table 3.4: size of the TRECVid 2008 and 2010 databases.

3.2.3.1 Re-ranking on Semantic Indexing Task TRECVid 2010

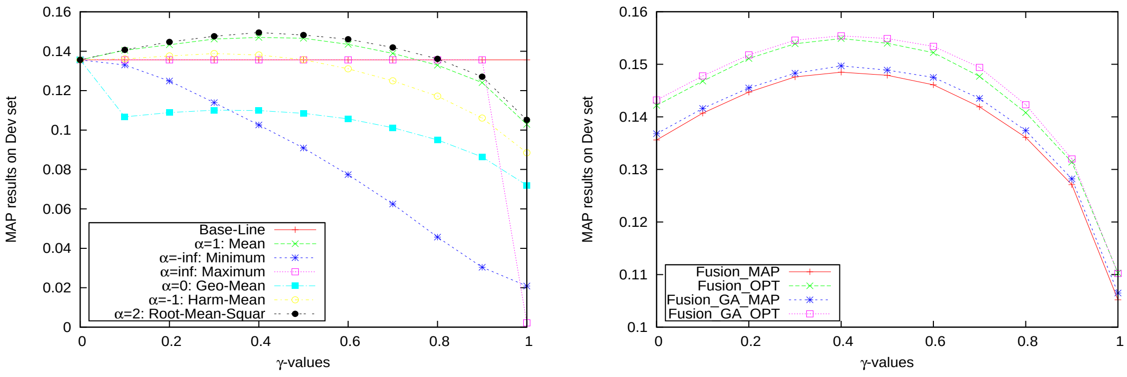
At first, we have applied the re-ranking method on a homogeneous dataset. This experiment was conducted on the TRECVid 2010, which provided 130 concepts with ground truth labels in a training set. The evaluation was done by calculating the Mean Average Precision (MAP) on only 30 concepts that were chosen by NIST.

We have evaluated the re-ranking method on four different initial classification results, which have been submitted to TRECVid 2010 (Gorisse *et al.* [2010]). We included different fusion strategies, such as weighted and direct optimized weighted fusion (denoted as *Fusion_MAP and Fusion_OPT*), as well as the combination of these two fusion types with the genetic fusion (denoted as *Fusion_GA_MAP and Fusion_GA_OPT*). These fusion strategies were applied on score vectors obtained by training different systems on 45 different descriptors, including audio and visual descriptors, which have been produced by the partners of the IRIM project of the GDR-ISIS (Gorisse *et al.* [2010]). Each of these fusion methods can be considered as the context fusion method, which we took as the baseline method for our re-ranking algorithm.

3.2.3.2 Parameters' Optimization

The tuning of α, θ and γ parameters (equation 3.2 and 3.3 in section 3.2.2), was conducted using the aforementioned initial classification results, which were calculated on the TRECVid 2010 development set. The aim of the tuning is to get the best values of α, θ and γ , in order to obtain the best performance of our system.

Figure 3.2 gives the results of tuning α and γ, in which we show the performance of the system in function with γ. The evaluation metric is taken by the MAP on the 130 concepts of the TRECVid 2010 development set. Each plot in figure 3.2(a), is related to a different value of α. It also shows the MAP with different values of γ (including $\gamma = 0$ and $\gamma = \inf$). For each video, we have set θ to be the number of all shots related to it, and we have used the initial scores of *Fusion_MAP* for evaluation. As we can see from the plots, $\alpha = 1$ and $\alpha = 2$ perform better that the others. Furthermore, the best result can be obtained with $\alpha = 2$ (Root Mean Square) and $\gamma = 0.4$. In addition, in figure 3.2(b), we show the performance of the system on the same collection, using our four initial scores with $\alpha = 2$. As we can see, the highest performance on each of the initial scores was achieved when the re-ranking method with $\gamma = 0.4$ was applied.



(a) α optimization

(b) γ optimization

Figure 3.2: Tuning α and γ parameters on TRECVid 2010 development set.

We consider now the θ parameter in equation 3.2. As mentioned before, this parameter controls the range, in which we expect the video to have a homogeneous content. The optimal value for this range is likely to depend on the collection contents. We re-run the previous evaluations with different values of θ, including the baseline $\theta = 0$ and $\theta = \infty$. This means that the global score of each video is assigned to all the shots belonging to it ($z_{ij} = z_i$). Figure 3.3 shows the MAP calculated on the 130 concepts on the *Fusion_GA_OPT* run, which we consider as the best run (as shown in figure 3.2(b)). The evaluations were done using the Rectangular and Gaussian windows with different θ-equivalent parameters for the re-ranking method. We have applied a sliding window of size $2\theta + 1$ as the neighbors of shot j using a rectangular function, and $\sigma = \sqrt{\theta(\theta+1)/3}$ using the Gaussian window. Thus, the two windowing functions have the same variance for the same value of θ. As we can see, for the two window functions the best result was obtained when $\theta = \infty$. This is probably due to the fact that the videos in the TRECVid 2010 collection are quite short (a few minutes in average), and they have homogeneous content. Thus, we can conclude that local re-scoring (using local neighbors) does not perform better than global re-scoring (using all the shots within the video).

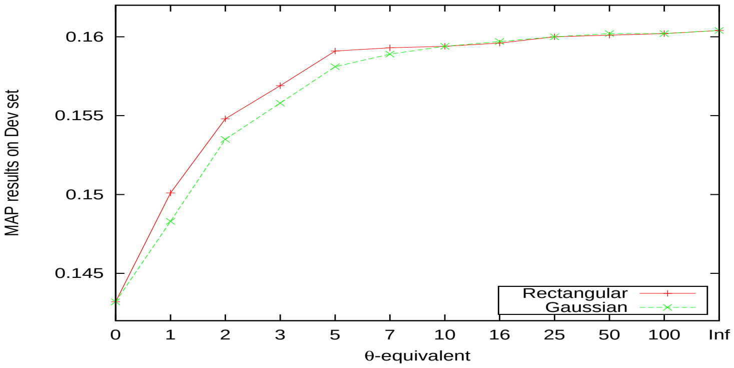


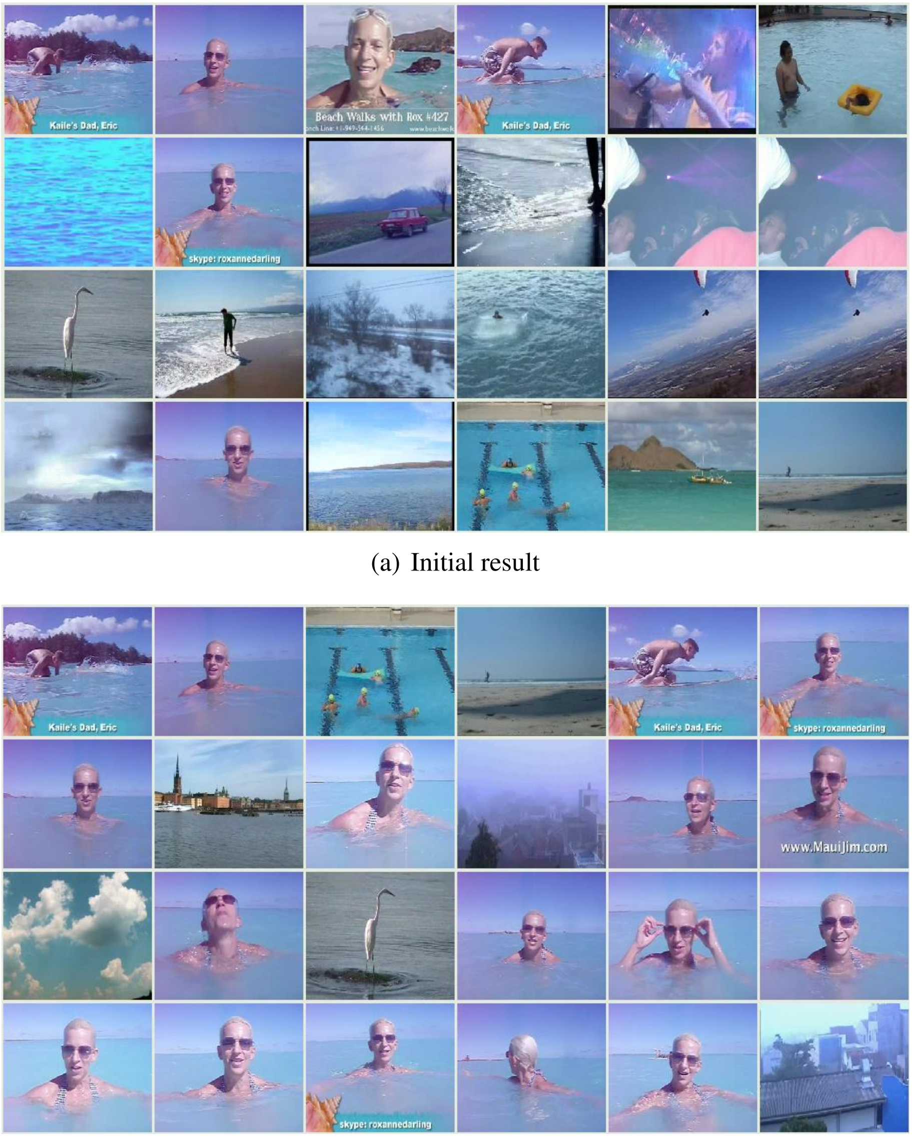
Figure 3.3: Tuning θ-equivalent parameters on the TRECVid 2010 development set, using *Fusion_GA_OPT* run.

3.2.3.3 Evaluation on the 2010 test set

We have applied the proposed method on the TRECVid 2010 test set, with the best parameters ($\alpha = 2$, $\gamma = 0.4$ and $\theta = \infty$) obtained by the cross-validation, with the two windowing functions (rectangular and Gaussian). We have compared the new results, which were obtained after re-ranking, with the results of the initial scoring methods from the best run (the *Fusion_GA_OPT* run). According to the official results of the TRECVid 2010, we obtained the 33^{th} place over 101 (Safadi *et al.* [2010b]).

Figure 3.4 shows the shots that are retrieved as the top relevant to the query for the concept *Swimming*. As we can see, the re-ranking (figure 3.4(b)) enhances the retrieved results in the top of ranked list better, by correcting the order of the initial results (figure 3.4(a)).

In table 3.5, we give the evaluation results of the re-ranking method on the 30 concepts of the TRECVid 2010 test set, using the MAP. As we can see, our proposed method has significantly improved the performance of the initial scoring methods. On this collection, the proposed re-ranking method - with the fully homogeneity $\theta = \infty$ - was able to improve the system performance with about 18% in average. The absolute MAP values are significantly different than in cross-validation (on the development set, see figure 3.3). This is probably due to the fact that the set of concepts is different than in the development set (only 30 concepts out of 130).



(a) Initial result

(b) Result after re-ranking

Figure 3.4: The top 24 retrieved samples for the concept '*Swimming*'; (a) shows the baseline system; the result with our re-ranking method is shown in (b).

	θ/σ	MAP
Baseline	0	0.0480
ALL	∞	0.0568 (+18%)
Rectangular	$\theta = \infty$	0.0568 (+18%)
Gaussian	$\sigma = \infty$	0.0568 (+18%)

Table 3.5: Results of the re-ranking method on the test set of TRECVid 2010.

3.2.3.4 Re-ranking on HLF Task of the TRECVid 2008

The second experiment was conducted on the TRECVid 2008 High-Level Feature (HLF) extraction task. The MAP on the 20 concepts of TRECVid 2008 is used as the evaluation metric, which measures the system's performance. The evaluation of the re-ranking method has been conducted using the simple late fusion of four types of image descriptors taken from IRIM GDR-ISIS partners (Gorisse *et al.* [2010]), which are: a combination of color histogram and Gabor transform, texture patterns, quaternionic wavelets and bag of SIFTs. The multi-learner approach based SVM-RBF kernel was used as the classification system. It was implemented as in section 3.1.2. Since the TRECVid 2008 sets are not as homogeneous as the TRECVid 2010 sets (see table. 3.4), we have fixed only the two optimal parameters $\alpha = 2$ and $\gamma = 0.4$ (taken from section 3.2.3.2). The goal was to find the best value of θ for the re-ranking method, when dealing with non-homogeneous videos .

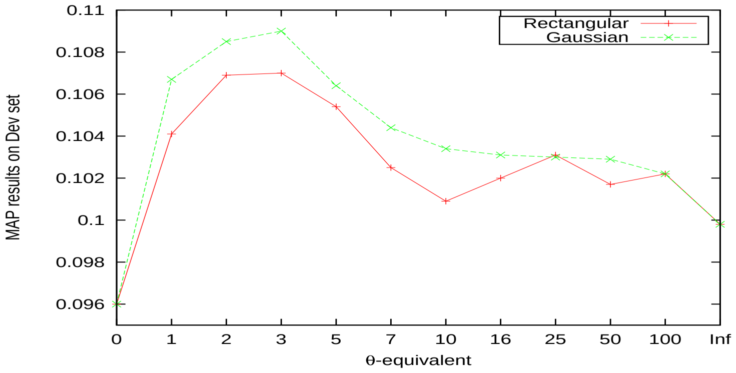


Figure 3.5: Tuning θ-equivalent parameter on TRECVid 2008, using the fusion of four descriptors.

We have evaluated our method on the TRECVid 2008 development set, using the late fusion of the four aforementioned descriptors, with different values of θ-equivalent parameter, and within the same conditions as in section 3.2.3.2. We present the performance of the systems in figure 3.5, which shows the MAP (calculated on the 20 concepts) with different values of θ-equivalent in both functions, the rectangular and Gaussian. As we can see, the Gaussian function performs better than the rectangular function. The performance, using the two windowing functions, enhances significantly when θ-equivalent is small, where the best result is given when $\theta = 3$. In the Gaussian function the best results is obtained when $\theta = 3 \rightarrow \sigma = \sqrt{3(3+1)/3} = 2$.

We have also evaluated the re-ranking method with the optimal values ($\alpha = 2, \gamma = 0.4$ and $\theta = 3$) on the TRECVid 2008 test set. It was evaluated using the two windowing functions. The final results are reported in table 3.6, in which we show the performance using different values of θ: $\theta = 0$ is the baseline, $\theta = \infty$ corresponds to the application of the re-ranking on the whole videos, and the optimal θ-equivalent values (i.e. θ / σ), which define respectively the rectangular and Gaussian functions. As we can see, the re-ranking with the optimal θ can significantly enhance the performance of the retrieval system. As expected, this collection is not homogeneous and there is not much enhancement when the re-ranking is done by a global

	θ/σ	MAP
Baseline	0	0.099
ALL	∞	0.101 (+2%)
Rectangular	$\theta = 3$	0.112 (+13%)
Gaussian	$\sigma = 2$	0.109 (+11%)

Table 3.6: Results of the re-ranking method on the test set of TRECVid 2008.

score on the whole video (i.e. $\theta = \infty$). When applying the re-ranking with $\alpha = 2, \gamma = 0.4$ and $\theta = 3$, the performance of the system is enhanced in average by about 11-13% on the late fusion of the used descriptors with both the Gaussian and the rectangular windows.

3.3 Normalization Techniques for Video Descriptions

In multimedia indexing, a considerable research effort is directed towards the development of efficient, fast and robust indexing and retrieval systems. There are still some major challenges that need to be tackled to increase the retrieval performance of the indexing system, especially when the datasets are of large-scale. One possibility of increasing the system's performance is to carefully examine the feature normalization techniques, which have the potential to greatly decrease the error rate of the classification, and thus increase the indexing performance. However, it has been so far neglected in most research papers on multimedia indexing. In general, only a few words are devoted to the used normalization technique, even though feature normalization is a crucial step for the multimedia indexing systems.

In general, for video indexing, the Chi square distance is considered to be more suitable than the Euclidean distance to compare histogram-based visual descriptors. SVM with RBF kernels can be used with both types of distance, the Euclidean and Chi square. However, the Chi square distance has two inconvenience: it is significantly more costly to compute because of the divisions in its formula and it is not compatible with PCA-based dimensionality reduction. While the Euclidean distance is conserved during the application of the PCA rotation matrix, the Chi square distance is completely transformed, almost randomly and might even become undefined, since it is normally computable only between vectors with positive or null components, a property which is not conserved during the application of the PCA rotation matrix.

In this section, we investigate a simple descriptor component transformation whose goal is to make the Euclidean distance closer to the Chi square distance. After this transformation, the Euclidean distance is expected to be as suited as the Chi square distance for comparing histogram-based image descriptors and an SVM with a Chi-square distance based RBF kernel is expected to be as suited as an SVM with an Euclidean distance based RBF kernel for image classification using histogram-based image descriptors. This transformation permits a reduction of the classification time both from using a distance simpler to compute and from being able to perform a dimensionality reduction using PCA. We compare the classification performance on TRECVid 2010 using the multi-SVM with RBF kernels with either the Chi-square or the Euclidean distance. The comparison is complicated because other and complementary normalisations can be performed either at the level of the descriptor vector (e.g. L1 or L2 normalization to a unit length) or at the level of the vector component (e.g. min-max or standard deviation normalization) or a combination of several of them. Furthermore, We present an empirical evaluation of several feature normalization techniques, namely: unit-length normalization (L_1 and L_2), min-max normalization, zero-mean and unit-variance normalization (σ-norm) and the power-law normalization. These normalization techniques are applied to several feature vector description methods and applied on the semantic indexing task of the TRECVid 2010 collection.

Another objective is to show that after an appropriate power-law transformation, the Euclidean distance becomes as effective as the Chi-square distance for image classification using SVMs with RBF kernels. Moreover, PCA-based dimensionality reduction permits a further computation time reduction while still being effective as well.

3.3.1 Feature Normalization Techniques

The main goal of feature normalization, is to independently normalize the feature components, in such a way that their values lie within a similar range (e.g. $[0, 1]$ range). The normalization is often done by either using the component values of each vector independently (e.g. the L_1 and L_2 normalization), or by normalizing the vectors using their bin values (e.g. min-max normalization). However, other normalization techniques work directly on the values independently, and it does not consider any of the other related values (e.g. power-law normalization).

Let X be the set of N feature vectors of the dataset to be normalized, each vector x_i consists of d-dimensions (components) $x_i = (1, 2, \ldots, d)$. In the following, we give five techniques of feature normalization, which are often used for image and video representation.

L_1 and L_2 normalization L1 and L2 normalization are often considered as the unit length normalizations. The two normalization methods scale independently their components. The normalized feature vectors (X') are produced through applying the following formula:

$$x'_{ij} = \frac{x_{ij}}{\|x_i\|}, i = 1, 2, , n, \quad j = 1, 2, , d$$

where c_{ij} stands for the j^{th} component of vector x_i, and $\|.\|$ denotes the norm operator, which in $L1$ is $\sum_j x_{ij}$ and in L_2 is $\sum_j x_{ij}^2$. The L_1 and L_2 normalization methods are widely used to normalize the feature vectors based histograms and back of words (BoW).

Min-Max normalization This function aims to scale the values for each feature bin (in a low-level description), so that they all fall in the range of *Lower to Upper* bounds (l, u). Thus, the descriptor values are normalized by subtracting the minimum and maximum value for each feature bin and then applying the following equation on each bin value:

$$x'_{ij} = l + \frac{(u - l) \times (x_{ij} - min_j)}{max_j - min_j}$$

where x_{ij} is the j^{th} feature component of the feature vector x_i, min_j and max_j are respectively the minimum and maximum value of the j^{th} component among X, and u and l are the new dimension space. Results in x' are often normalized to the $[0, 1]$ range.

Zero-mean and unit-variance normalization (σ-norm) The feature values are normalized by subtracting the variance value for each feature bin and scaling all the values, so that they will be normalized by the variance σ of its feature bin. This can be done by applying the following equation on each bin value:

$$\sigma_j = \frac{\sum_{i=1}^{N}(x_{ij} - \bar{x_{ij}})^2}{N}; \quad j = 1, \ldots, d$$

$$x'_{ij} = \frac{x_{ij} - \bar{x_j}}{\sigma_j} \quad i = 1, \ldots, n \; and \quad j = 1, \ldots, d$$

where d indicates the length of the vector x_i (i.e. number of components), n is the number of samples in the collection; $\bar{x_i}$ and σ_i are the mean and variance values of the i^{th} feature bin respectively.

Power-law normalization The goal of the power transformation is to normalize the distributions of the values, especially in the case of histogram components. It simply consists of applying an $x \leftarrow x^{\alpha}$,$x \leftarrow -(-x)^{\alpha}$ if $x < 0$) transformation on all components individually (Safadi *et al.* [2011a]).

The power-law transformation was applied by Jégou *et al.* [2011], in which the authors applied the power-law only on the *Fisher kernel* descriptor. They empirically observed that this step consistently improves the quality of the representation. They gave several complementary interpretations that justify this transform. First, it reduces the influence of bursty visual elements, which were shown to corrupt the image similarity in (Jégou *et al.* [2009]). Second, assuming the compound Poisson distribution as a good generative model of Fisher Vectors, the power normalization can be interpreted as a variance stabilizing transform, which corrects the dependence between the variance and the mean. The authors have applied the power-law with $\alpha = 0.5$. However, the authors used only one descriptor to justify their conclusions and they did not show the impact of the power-law with different values of α.

In the following, we will study the impact of the α parameter with the power-law on different descriptors.

3.3.2 Experiments

The experiments on the normalization method of video description, were conducted on the TRECVid 2010 collection. This data collection consists of two large sets: the development and the test set, as given in table 3.4. We have used several descriptors of different types and sizes, which have been produced by various partners of the IRIM project of GDR-ISIS (Gorisse *et al.* [2010]). Most of the selected descriptors are based on the color histograms and the bag of words approaches. However, we choose to compare the methods also with different types of descriptors, such as those based on Gabor filter and audio. In practice, we have used 12 descriptors that indicates in table 3.7, more details on the descriptors are given in appendix B.

3.3.2.1 Optimizing parameters

For the evaluation of each normalization method, we use the multi-learner approach (ML) based SVM with RBF kernel (MSVM) as a classifier. For simplicity reasons and due to the results of other experiments, we have fixed the f_{min} parameter of the MSVM to be equal to 4. However, there are other parameters that need to be optimized, such as the γ parameter for the classifier. In addition, the power-law normalization has the α parameter, which also needs to be tuned. The optimization is done by the cross-validation on the development set of TRECVid 2010, in which we split the dataset into two sets: the training and validation sets. In the following, we present the optimization process of these two parameters:

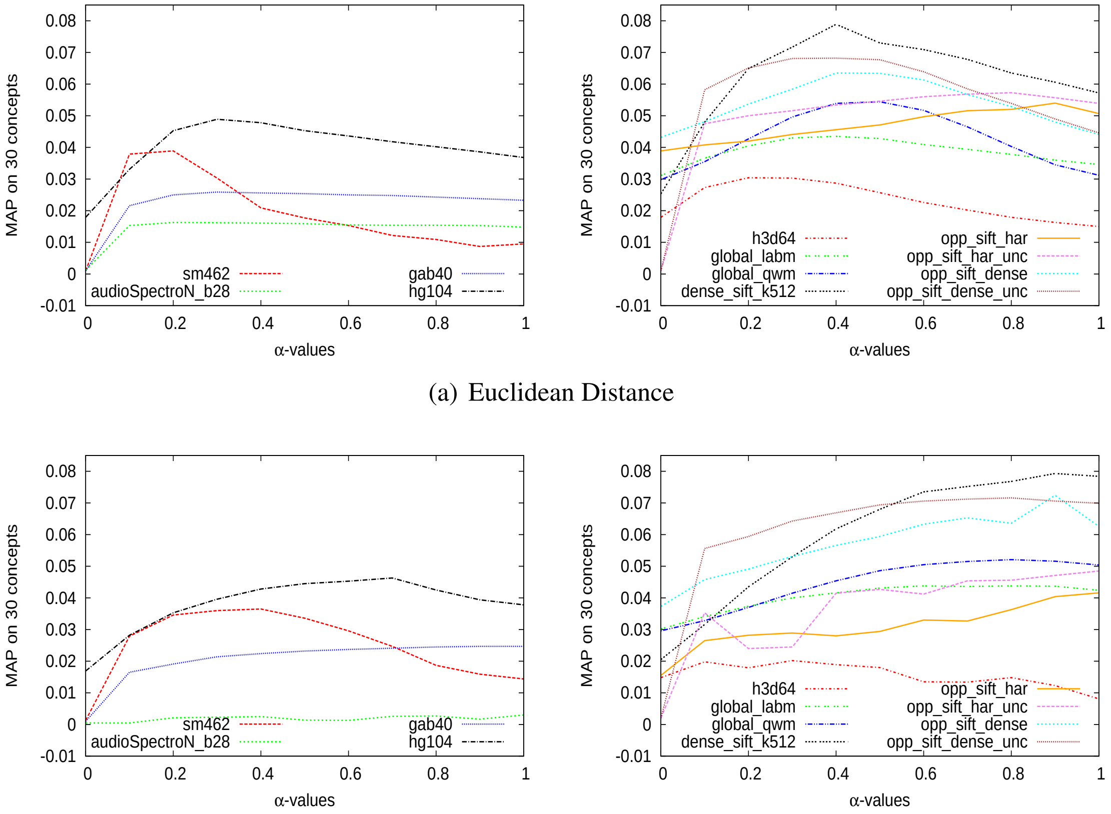
The parameter α for the power-law normalization The normalization methods, except the power-law, are of free parameter optimization. Only the power-law has a parameter α, which needs to be tuned. Before we compare the normalization methods, we need to find the optimal value of α in the power-law for each of the tested descriptors. Thus, first

	Raw	L1	L2	σ-norm	Min-max	Power-law
EUR/sm462	0.0095	0.0120	0.0121	0.0189	0.0115	0.0389
GIPSA/AudioSpectroN_b28	0.0155	0.0155	0.0156	0.0138	0.0157	0.0163
LIG/gab40	0.0265	0.0264	0.0257	0.0240	0.0182	0.0271
LIG/hg104	0.0368	0.0327	0.0366	0.0407	0.0278	0.0489
LIG/h3d64	0.0158	0.0158	0.0159	0.0255	0.0161	0.0304
ETIS/global_labm1x3x192	0.0346	0.0347	0.0342	0.0316	0.0355	0.0432
ETIS/global_qwm1x3x192	0.0312	0.0337	0.0351	0.0356	0.0373	0.0544
INRIA/dense_sift_k512	0.0572	0.0588	0.0610	0.0695	0.0636	0.0790
LIG/opp_sift_har_1000	0.0507	0.0507	0.0529	0.0485	0.0455	0.0540
LIG/opp_sift_har_unc_1000	0.0539	0.0539	0.0540	0.0510	0.0516	0.0573
LIG/opp_sift_dense_1000	0.0441	0.0441	0.0449	0.0545	0.0494	0.0635
LIG/opp_sift_dense_unc_1000	0.0446	0.0446	0.0472	0.0617	0.0591	0.0682

Table 3.7: The MAP values on the TRECVid 2010 development set, using the different normalization methods with the Euclidean distance.

we have evaluated the power-law normalization on the different descriptors. This was done using 10 different values of $\alpha \in [0, 1]$. The evaluation was done on the TRECVid 2010 development set with the following settings: i) the MSVM was used as a classifier; ii) the γ parameter in the MSVM was set as function of the h parameter and the mean average distance between the vectors, calculated as follows: $\gamma = h/d_m$, where d_m is the mean distance (e.g. the Euclidean or the χ^2) and $h = log_2(i)$, in which five values of i are used ($i = -1, 0, 1, 2, 3$); and iii) the performance measure is the MAP, which is calculated on the 30 concepts of the TRECVid 2010 test set. The results of the α optimization are given in figure 3.6, for the two considered distances: the Euclidean is presented in sub-figure 3.6(a) and 3.6(b) presents the Chi-square. Each plot refers to the system performance using one descriptor. As the plots show, the α parameter has different values for each of the descriptors, with both distances. Since we believe that the only difference in the results for each descriptor and the used distance is α value, this shows the important of choosing the best value of α. For instance, with the Euclidean distance the hg104 descriptor has the best performance with $\alpha = 0.3$, the dense_sift_k512 descriptor has the highest performance with $\alpha = 0.4$.

Tuning the h parameter For all the normalization methods, the γ parameter of the MSVM with the RBF approach was optimized as in the previous paragraph. It was optimized independently for each descriptor. In tables 3.7 and 3.8, we give the results of the system performance on the development set of TRECVid 2010, respectively with the Euclidean and Chi-square distance, using the five normalization methods. Furthermore, the results obtained after normalization are compared with the result when the baseline method is used with the both distances, which means no normalization is applied (i.e raw normalization). As the mentioned tables shows, the system performance varies significantly with the different normalizations. For the L_1, L_2, σ-norm and min-max normalization, the performance is in most cases closer to the baseline method, and the best normalization



(a) Euclidean Distance

(b) Chi-square Distance

Figure 3.6: Tuning α parameter of the power-law normalization on TRECVid 2010.

	raw	L1	L2	σ-norm	Min-max	Power-law
EUR/sm462	0.0144	0.0155	0.0155	0.0243	0.0149	0.0365
GIPSA/AudioSpectroN_b28	0.0030	0.0030	0.0019	0.0017	0.0096	0.0026
LIG/gab40	0.0247	0.0215	0.0215	0.0240	0.0186	0.0247
LIG/hg104	0.0378	0.0456	0.0387	0.0447	0.0350	0.0463
LIG/h3d64	0.0081	0.0081	0.0124	0.0137	0.0112	0.0202
ETIS/global_labm1x3x192	0.0424	0.0435	0.0399	0.0379	0.0435	0.0438
ETIS/global_qwm1x3x192	0.0504	0.0520	0.0455	0.0417	0.0430	0.0521
INRIA/dense_sift_k512	0.0784	0.0775	0.0760	0.0762	0.0841	0.0793
LIG/opp_sift_har_1000	0.0416	0.0416	0.0370	0.0367	0.0334	0.0404
LIG/opp_sift_har_unc$_1$000	0.0485	0.0485	0.0453	0.0425	0.0432	0.0471
LIG/opp_sift_dense_1000	0.0626	0.0626	0.0623	0.0586	0.0572	0.0724
LIG/opp_sift_dense_unc_1000	0.0699	0.0699	0.0746	0.0688	0.0676	0.0716

Table 3.8: The MAP values on the TRECVid 2010 development set, using the different normalization methods with the Chi-square distance.

among them is not stable for all the descriptors. However, the power-law normalization has always the highest performance with the Euclidean distance, compared to the other normalization methods.

In table 3.8, the results are not stable, for instance there is no method of transformation, which is considered as the best for all the descriptors, however, in most cases the power-law gives the highest results. The power-law transformation with Euclidean distance, in most cases, has higher scores than the Chi-square. the Chi-square distance is significantly better with the opp_sift_dense* descriptors.

Figure 3.7 shows the choosing process of the h values. It compares the system performance (MAP) on the development set, with different values of h using the dense_sift_k512 descriptor, which gave the highest performance. As we can see, the h has an important impact on the system performance and it varies for each descriptor. However, for all values of h, the power-law normalization performs always best, and the highest performance was obtained with $h = 1$.

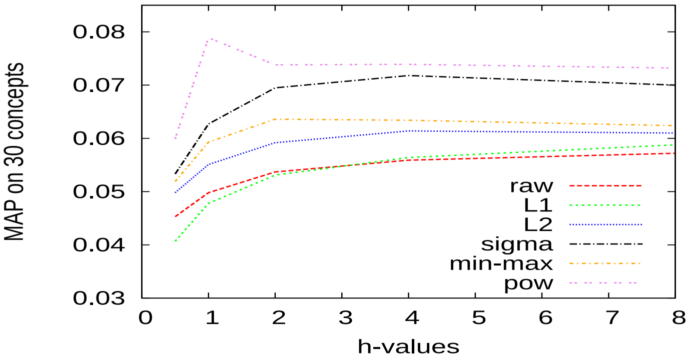


Figure 3.7: Tuning γ parameter on TRECVid 2010, using the sift_k512 descriptor with the five normalization methods.

3.3.2.2 PCA with Euclidean Distance

Figure 3.8 shows the system performance (MAP) obtained by applying the power-law normalization followed by PCA dimension reduction, with the all considered descriptors. For those with a small dimension the use of the PCA is not that important, since their dimension is already small. The main objective is to show the performance when using the PCA on descriptors of high dimensions. We have tuned the k number of PCA (i.e. number of important components) on each of the considered descriptors, using fractions from 0.1 to 1 of the original dimension. As shown in the mentioned figure, the number of the important components, varies for each of the descriptors, for long descriptors, we have fixed the k after PCA to be the value of the first fraction that has a higher performance or closer to the performance of the original dimension. For instance, the chosen k-components for the best descriptor (i.e dense_sift_k512) is $0.4 \times 512 = 204$.

	Raw	L1	L2	σ-norm	Min-max	Power-law	PCA (Power-law)
EUR/sm462	0.0006	0.0035	0.0035	0.0104	0.0062	0.0246	0.233
GIPSA/AudioSpectroN_b28	0.0005	0.0005	0.0007	0.0006	0.0006	0.0011	0.0035
LIG/gab40	0.0097	0.0106	0.0106	0.0100	0.0065	0.0115	0.0114
LIG/hg104	0.0153	0.0132	0.0177	0.0107	0.0167	0.0246	0.0240
LIG/h3d64	0.0040	0.0004	0.0036	0.0053	0.0034	0.0145	0.0126
ETIS/global_labm1x3x192	0.0126	0.0126	0.0122	0.0126	0.0115	0.0270	0.265
ETIS/global_qwm1x3x192	0.0142	0.0137	0.0133	0.0128	0.0134	0.0217	0.0214
INRIA/dense_sift_k512	0.0313	0.0305	0.0338	0.0389	0.0340	0.0418	0.0405
LIG/opp_sift_har_1000	0.0177	0.0177	0.0228	0.0156	0.0142	0.0223	0.0249
LIG/opp_sift_har_unc_1000	0.0210	0.0210	0.0268	0.0255	0.0209	0.0293	0.0313
LIG/opp_sift_dense_1000	0.0214	0.0214	0.0244	0.0332	0.0282	0.0375	0.0381
LIG/opp_sift_dense_unc_1000	0.0220	0.0220	0.0251	0.0381	0.0315	0.0433	0.0426
Fusion	0.0437	0.0447	0.0470	0.0523	0.0468	0.0632	0.0646
Re-rank	0.0524	0.0529	0.0555	0.0624	0.0533	0.0723	0.0731

Table 3.9: The MAP values on the TRECVid 2010 test set, using the different normalization methods with the Euclidean distance.

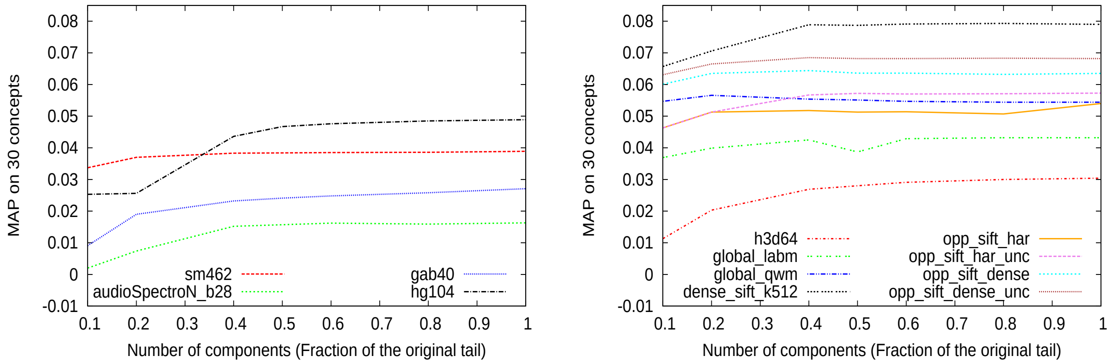


Figure 3.8: Evaluating the PCA with Euclidean distance on TRECVid 2010

3.3.2.3 Results on the test set of TRECVid 2010

We have evaluated the five different method for normalization on the TRECVid 2010 test set, with the best h parameters, the best (α) and PCA dimension reduction with the Euclidean distance for the power-law, which were obtained through the cross-validation. The results on the test set are shown in tables 3.9 and 3.10, they were calculated using both the Euclidean and Chi-square distances. As we can see, the results are consistent with those on the development set. The power-law performs better that all the other evaluated method for normalization, with the all used descriptors, however, it is also better with the Euclidean distance than the Chi-square in most cases. The use of PCA dimension reduction makes the system faster and keeps the performance quality of the system with the Euclidean distance. The power-law performs better with the fusion and re-ranking as well and it reaches the highest score of 0.0731 with PCA and the Euclidean distance, which is close to the best result on TRECVid 2010 (SIN) that was 0.0900.

3.3.2.4 Processing times

All the experiments were done on a machine which has two quad-core processors running at 2.66 GHz and 32 Gbytes of Ram. The execution time depends upon the size of the descriptor. In table 3.11, we report the processing time over the all learning and indexing process with the power-law transformation. The numbers in this table refers to the processing time in hours of the 30 concepts, however, it shows the processing times with the two distances (the Euclidean and the Chi-square) and the PCa with the Euclidean. As we can see, the MSVM-RBF with the euclidean distance is significantly faster than the Chi-square. After applying the PCA the system is much more faster, while as we have shown before the performance is not significantly affected.

	Raw	L1	L2	σ-norm	Min-max	Power-law
EUR/sm462	0.0023	0.0042	0.0041	0.0045	0.0057	0.0178
GIPSA/AudioSpectroN_b28	0.0006	0.0006	0.0003	0.0001	0.0007	0.0006
LIG/gab40	0.0103	0.0098	0.0098	0.0100	0.0074	0.0104
LIG/hg104	0.0182	0.0214	0.0136	0.0193	0.0164	0.0207
LIG/h3d64	0.0033	0.0033	0.0029	0.0054	0.0035	0.0046
ETIS/global_labm1x3x192	0.0238	0.0228	0.0234	0.0174	0.0228	0.0288
ETIS/global_qwm1x3x192	0.0213	0.0211	0.0175	0.0160	0.0192	0.0227
INRIA/dense_sift_k512	0.0349	0.0380	0.0420	0.0388	0.0341	0.0377
LIG/opp_sift_har_1000	0.0187	0.0187	0.0078	0.0102	0.0106	0.0154
LIG/opp_sift_har_unc_1000	0.0284	0.0284	0.0255	0.0146	0.0205	0.0260
LIG/opp_sift_dense_1000	0.0340	0.0340	0.0322	0.0332	0.0309	0.0346
LIG/opp_sift_dense_unc_1000	0.0433	0.0433	0.0371	0.0422	0.0389	0.0451
FUSION-All	0.0592	0.0597	0.0598	0.0596	0.0604	0.0618
RE-RANK-All	0.0652	0.0659	0.0665	0.0675	0.0674	0.0683

Table 3.10: The MAP values on the TRECVid 2010 test set, using the different normalization methods with the Chi-square distance.

	PCA-Euclidean	Euclidean	Chi-square
EUR/sm462	44.54	78.1	137.2
GIPSA/AudioSpectroN_b	05.43	10.9	11.9
LIG/gab40	08.58	9.5	17.3
LIG/hg104	15.72	19.3	28.9
LIG/h3d64	06.96	7.6	11.9
ETIS/global_labm	48.96	148.5	140.8
ETIS/global_qwm	29.00	127.1	126.7
INRIA/dense_sift_k512	26.56	104.2	171.9
LIG/opp_sift_har	80.83	81.8	123.5
LIG/opp_sift_har_unc	80.12	101.2	281.6
LIG/opp_sift_dense	72.72	92.3	199.4
LIG/opp_sift_dense_unc	61.96	95.9	355.4

Table 3.11: Processing time (in hours) on the 30 concepts of the test set, using only the power-law transformation.

3.4 Summary

In this chapter, the multimedia indexing and retrieval systems and two problematic challenges of developing these systems, have been presented. We have also introduced our contributions in three main phases of the indexing systems: a new classification, the enhancement of the ranking order of the retrieved samples and the feature normalization techniques.

We have introduced our multi-learner approach, and we showed its ability to increase the classification performance of the indexing systems. This was evaluated and compared using several single- and multi-learner approaches (e.g. logistic regression and SVM) for the concept indexing task in video documents. We evaluated and compared several single- and multi-learner approaches for the concept indexing task in video documents. The multi-learner approach is designed to address best the problem of sparse concepts, which lead to a strong numerical imbalance between positive and negative classes. The evaluation and comparison have been done in the context of the concept detection task of the TRECVid 2008 and 2009 campaigns.

The multi-learner approach is tested with three types of classifiers: linear SVM, SVM with Gaussian kernels and logistic regression. The methods were evaluated using several types of descriptors. The obtained results are fairly stable against the type of the used descriptor. The multi-learner approach works better than the single-learner approach. In this context, the classifier based on logistic regression works better than the linear SVM classifier, but less well than the SVM classifier with Gaussian kernel.

Video retrieval can be done by ranking the samples according to their probability scores that were predicted by classifiers. It is often possible to improve the retrieval performance by re-ranking the samples. We have proposed a re-ranking method that improves the performance of semantic video indexing and retrieval. The re-ranking is done by re-evaluating the scores of the shots using the homogeneity and the nature of the video they belong to.

The experimental results showed that the proposed re-ranking method was able to improve the performance of the system by about 18% in average on the TRECVid 2010 semantic indexing task, which is considered to have videos with homogeneous contents. For TRECVid 2008, in the case of collections of videos with non-homogeneous contents, the system's performance was improved by about 11-13%.

Moreover, we have evaluated and compared five normalization techniques. The evaluation was conducted using the TRECVid 2010 collection, with different types of video descriptors and the MAP as the evaluation metric. A multi-SVM with a Gaussian RBF kernel approach was used as a classifier with either the Euclidean or Chi square distance. Moreover, the normalization techniques were also evaluated in the same conditions, after applying a PCA for dimensionality reduction.

Experimental results show that the normalization based power-law is the best to consider among the others, especially with the Euclidean distance. However, the α parameter for the power-law has an important impact on the classification result, thus it needs to be optimized through the cross-validation. The power-law transformation makes the SVM with Euclidean distance as effective as the Chi square one. Moreover, the PCA-based dimensionality reduction permits a further computation time reduction without system performance loss.

Chapter 4

Evaluation of Active Learning Methods for Multimedia Indexing

This chapter proposes the integration of the multi-learner approach as a classifier for the active learner, called ALML. It also proposes an incremental method to speed up the performance of the ALML approach, denoted as Inc-ALML. Furthermore, it shows some evaluation experiments on the proposed methods and it ends with some conclusions and remarks.

4.1 Active Learning with Multiple Classifiers for Multimedia Annotation (ALML)

Several strategies or heuristics can be considered to predict samples' usefulness. Most of them operate by *selective sampling*, which means progressively adding to the training set the samples that are expected to be the most informative. The most popular ones include the uncertainty and the relevance sampling as described in section 2.6.2. The relevance sampling is especially effective for highly class imbalanced datasets, which in multimedia indexing is very frequent. Finding negative samples is easy. Whatever the sampling strategy is, these generally come numerous enough. Active learning with relevance sampling can significantly increase the ratio of positive to negative samples in the set of annotated samples. The imbalance ratio can be reduced (in the early iterations) by selecting samples from the majority class by factors of up to 5 times the minority class, according to Ayache & Quénot [2007a]. However, the class imbalance problem becomes more critical and bigger when the number of annotated samples of the dataset are increased.

Active learning and the multi-learner approach are two different and complementary ways of dealing with the class-imbalanced problem; the former attempting to build more balanced training sets and the latter trying to get the best from an imbalanced training set. In this section, we combine the video annotation system based on active learning with the multi-learner approach. Furthermore, we show that active learning is more effective when it is combined with the multiple classifiers (or multi-learner) approach. The combination is done by replacing the active learner by the multi-learner approach, which is represented in section 3.1.2. Figure 4.1 shows an instance of this system.

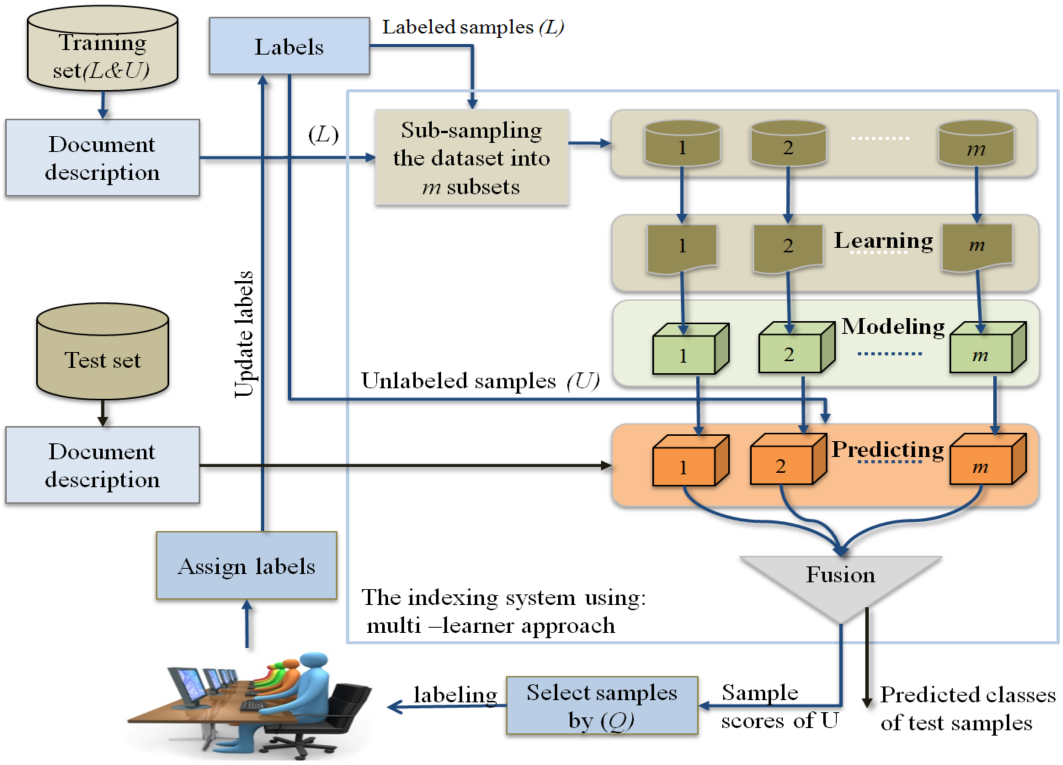


Figure 4.1: Active learning with multi-learner approach for multimedia annotation.

4.1.1 Our Proposed Approach

The active learning with multi-learner Approach is detailed in algorithm 2. The algorithm description includes an evaluation part, which is not actually part of the active learning procedure but is included for convenience. Apart from that, this algorithm is a classical active learning algorithm, in which we have replaced the single classifier by a set of elementary classifiers. Thus, the proposed model ($ALML$) is defined as follows (Safadi & Quénot [2010]; Safadi & Quénot [2010]):

$$ALML = \langle ML, Q \rangle$$

where ML implements the model proposed in section 3.1.2, and Q is the selection strategy of the active learner. For implementation purposes, the elementary learning algorithm A is split into two parts: train and predict. A global parameter, the mono-learner, can force the classical active learning mode with a single classifier.

At each iteration i, the development set Dev (or S) is split into two parts: L_i, labeled samples and U_i, unlabeled samples. A global parameter f_{min} defines the ratio between the negative and positive samples in all learners and for all iterations. This defines the number of negative samples for each learner at iteration i. In the multi-learner approach, the number of learners is computed so that each negative sample appears, in average, a given number of times (usually once) in the different subsets T_j. The T_j contains all positive samples and a randomly chosen subset of negative samples. Classifiers C_j are then trained on the T_j with associated labels and applied to the test set for evaluation and on the U_i set for the selection of the next

Algorithm 2 Active Learning with Multi-Learner Approach

```
    $S$: all data samples.
    $L_i, U_i$: labeled and unlabeled subsets of $S$.
    $A$=(Train, Predict): the elementary learning algorithm.
    $Q$: the selection (or querying) function.
    Initialize $L_i$ (e.g. $10$ positives & $20$ negatives).
    while $S \setminus L_i \neq \emptyset$ do
        if mono-learner then
            $m = 1$
        else
            $m =$ Calculate the number of Learners
        end if
        for all $j \in [1..m]$ do
            Select subset $T_j$ from $L_i$ for training
            $C_j \leftarrow$ Train($T_j$)
            $P_{un}^j \leftarrow$ Predict($U_i$, $C_j$)
            Evaluate the model on a test set
        end for
        $P_{un} \leftarrow$Fuse($P_{un}^j$)
        Apply $Q$ on $P_{un}$ and select $\tilde{X} \in U_i$ samples.
        $\tilde{Y}$ = Label $\tilde{X}$
        $L_{i+1} \leftarrow L_i \cup (\tilde{X}; \tilde{Y})$
        $U_{i+1} \leftarrow U_i \setminus \tilde{X}$
    end while
```

samples to be annotated. Predictions from the elementary classifiers are then merged in both cases to produce a single prediction score per sample. The predictions on the U_i set are used by the selection (or querying) function Q to produce a sorted list of the next samples to annotate. From the top of this list, an $\tilde{X}$ set is selected for annotation. The $\tilde{X}$ set is then added with the associated set of labels $\tilde{Y}$ to the L_i set to produce the L_{i+1} set. The $\tilde{X}$ set is also removed from the U_i set to produce the U_{i+1} set.

The global algorithm is determined by the A=(train, predict) elementary learning algorithm (e.g. logistic regression or support vector machine) and by the selection (or querying) function Q, which implements the active learning strategy (e.g. relevance or uncertainty sampling). It is also determined by some global parameters, such as the ratio between the number of negative and positive samples (depending in practice upon the learning algorithm and the descriptor type), the way of choosing the initial positive and negative samples (cold start), the fusion between classifier outputs (the Fuse function) and the way of choosing the number of new samples to be integrated at each iteration.

4.1.2 Experiments

We have evaluated the active learning with multi-learner approach in a variety of contexts. It has been applied using four types of image descriptors, two types of classifiers that have been evaluated in their mono- and multi-learner versions, and with two different active learning strategies (the relevance and the uncertainty sampling), which was completed by the random and linear scan sampling strategies for comparison. Five variants of the fusion function were considered, all applied to probability values: arithmetic mean, geometric mean, harmonic mean, minimum, and maximum. The cold start problem was not really explored: a random set of 10 positive and 20 negative samples was used to bootstrap the active learner. The global parameters of the single- and multi-learner approaches, such as the f_{min} ratio and the internal parameters of the classifiers, were determined by cross-validation within the development set only. We have chosen to select, at each iteration, samples to be added in a variable step size, since we have observed in previous experiments that having small steps in the beginning of the active learning process is better to speed the performance improvement. In practice, we have used 40 steps in total for the active learning algorithm, considering the geometric scale function with the following formula:

$$S_k = S_0 \times \left(\frac{N}{S_0}\right)^{k/K}$$

where N is the total size of the development set, S_0 is the size of the training set at the cold-start, K is the total number of steps and k is the current step. At each iteration (i.e. step), the algorithm calculates the S_k to be the size of the new training set and it chooses new samples to be labeled with the size equal to $S_k - S_{k-1}$. The evaluations were conducted using the TRECVid 2008 test collection and protocol (table 2.1).

It has been applied with a classification system using four types of image descriptors, which are taken from IRIM GDR-ISIS partners (Quénot *et al.* [2009]), including the LIG/hg104, CEALIST/Global_Tlep, ETIS/global_qwm1x3x256 and LEAR/bow_sift_1000 (see appedx. B). Two types of classifiers for our model (ML) were used: support vector machines (SVM) with RBF kernel and logistic regression (LR).

4.1.2.1 The TRECVid 2008 Collection

The TRECVid 2008 provided participants with $\approx$ 200 hours of cultural, news magazine, documentary, and education programming supplied by the Netherlands Institute for Sound and Vision. The collection contains 43616 video shots in the training set and 42461 shots in the test set. The training set is fully annotated for the evaluated concepts (nothing remains to be annotated), which makes the use of active learning irrelevant. However, such large, fully annotated sets constitute opportunities to simulate, evaluate and compare strategies and methods in active learning without the need of actually involving an oracle (Ayache & Quénot [2007a]). In our experiments, the active learning method starts with very few annotations available from the training set. Then, each time a human annotation is required, the corresponding subset of the full annotation set is made available to the active learner.

4.1.2.2 Execution Times

The experiments were done using several machines and nodes with 2.66 GHz Intel processors. Table 4.1 gives the total execution time for the whole active learning process (the 40 iterations), per method and per descriptor. They are calculated on the evaluated concepts of TRECVid 2008 (20 concepts). The processing times are reported in this table, related to one selection strategy (the relevance sampling); they are similar for the different sampling strategies.

Descriptor	dim	SSVM	MSVM	SLR	MLR
LIG/hg104	104	4.80	59.54	0.30	14.15
CEALIST/global_tlep	576	96.56	395.45	0.77	13.96
ETIS/global_qwm1x3x256	768	45.67	460.60	0.78	66.47
LEAR/bow_sift_1000	1000	181.00	592.10	0.62	27.50

Table 4.1: Execution times with relevance sampling strategy (in hours)

As expected, the single-learner versions are faster than the multi-learner ones. The ratio between both is much higher for LR than for SVM. This is due to the much lower f_{min} ratio for LR that induces a much greater number of learners. This almost compensates for the fact that the elementary LR classifier is much faster than the SVM classifier. The computation time generally increases with the descriptor dimensionality, however, not in a simple way and there are some exceptions.

4.1.2.3 Comparison of Active Learning Strategies

Figure 4.2 compares the effectiveness of the relevance and uncertainty sampling strategies for the four classifier types (i.e. SLR, MLR, SSVM and MSVM). The performance of the linear scan and random sampling strategies are shown as baselines. The results presented here are for the LIG_hg104 descriptor only; however, a similar behaviour is observed with the other descriptors. For the multi-learner experiments, fusion by harmonic mean has been used. These plots show the evolution of the indexing performance measured by the mean average precision (MAP) measure with the number of annotated samples. The faster it grows and the higher it goes, especially in the beginning, the better.

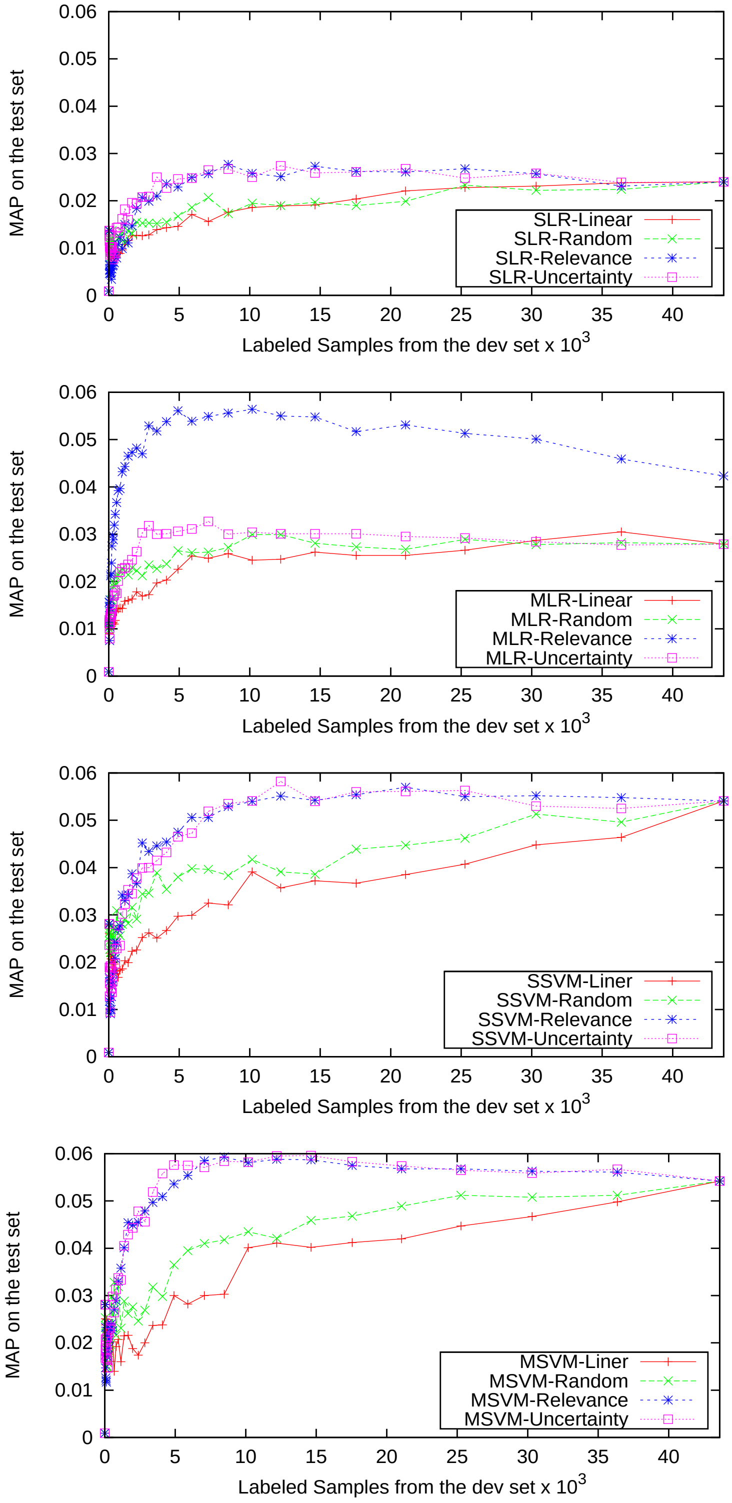


Figure 4.2: Linear, random, relevance and uncertainty sampling strategies with the LIG_hg104 descriptor. Classification methods: top: LR mono-learner, second: LR multi-learner, third: SVM-RBF mono-learner and bottom: SVM-RBF multi-learner, fusion method for multi-learners: harmonic mean.

Unsurprisingly, the SLR method leads to a much lower performance than the MLR or SVM methods, indicating that a single linear boundary is not appropriate for the considered type of data. The MSVM is the best method, and the uncertainty is the best sampling strategy. The SSVM method is almost as good for both strategies: it goes almost as high but it; grows slower than MSVM. However, it is significantly faster than MSVM, as we can see in table 4.1. The MLR method is almost as good as well, but only with the relevance sampling strategy: it grows as fast as the MSVM approach, but it has a bit lower performance than MSVM. However, in average it is significantly faster than the MSVM, considering all aforementioned descriptors. For all classifier types and querying strategies (excluding the baselines), the maximum performance value is reached when a small fraction (typically between 10% and 25%) of the training set is annotated. The (small) performance drop can be attributed to the fact that the imbalance between the positive and negative sample sets increases significantly: few new positive samples are discovered, while no new useful information is found in the next negative samples.

The overall system performance, in the absolute MAP, is low, about 0.075 for the best descriptor. However, for individual descriptors this performance is good. A classifier with a significantly higher performance can be built by fusing the outputs of several classifiers, which are learned on such descriptors. Furthermore, this performance can be improved by using multiple frames in the candidate shots. Note, that the performance of the best classification system at TRECVid 2008 was of 0.167 (type A run), see (**?**).

4.1.2.4 The Effectiveness of Active Learning with Multi-Learner Approach

Figure 4.3 compares the effectiveness of the four classifier types for the four considered descriptors. The results are presented for the relevance sampling strategy. For the multi-learner experiments, fusion by harmonic mean has been used. These plots show significant variability according to the descriptor type. SLR is always the worst method. MLR is competitive with SVM only for two types of descriptors. MLR increases the class imbalance problem between positive and negative samples. This can be seen from the figures after annotating 25% of the dataset, which decreases the system performance (by MAP). The MSVM approach is consistently the best method. The SSVM is often almost as good as the MSVM, except for one descriptor. Despite the variability according to the descriptor type, in general the slowest method leads to the best performance, with often a small difference in performance and a large difference in execution time. This allows us to tune the speed versus quality compromise over a wider useful range.

Some combinations of methods and strategies can also be used, for example, using the MLR with relevance sampling in the early iterations followed by MSVM with uncertainty sampling in the advanced iterations. The total execution time of the worst case (592.10 hours) is comparable to the total annotation time with a single annotator assigning one label to one video shot in an average of 2.5 seconds (605 hours for the 20 concepts). The experiments were conducted until the whole set was annotated for evaluation purposes. In practice, the annotations could be stopped after annotating only a fraction (e.g. 20%) of the training set and both the processing and annotation time would be reduced accordingly.

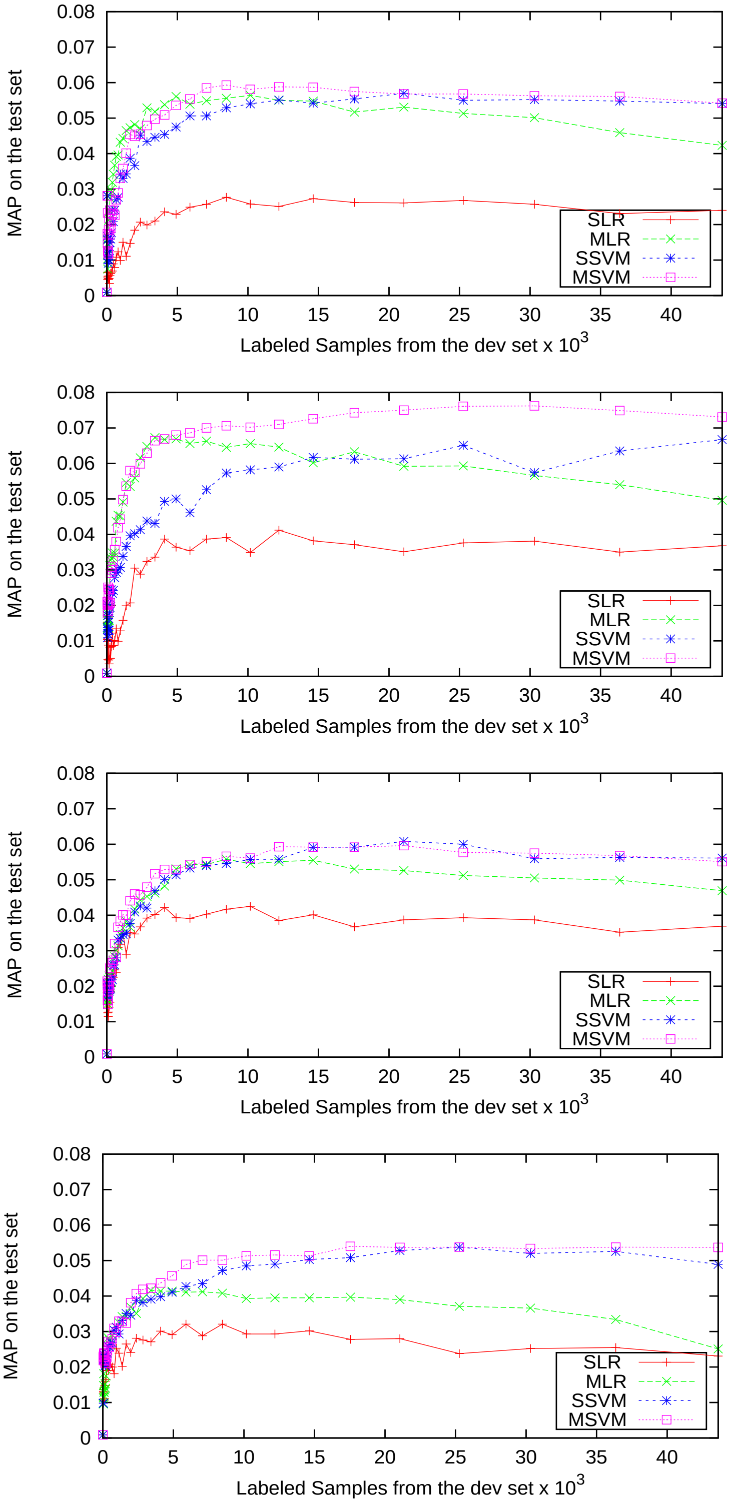


Figure 4.3: The four classifiers using relevance sampling strategy. Descriptors: top: LIG/hg104, second: CEALIST/global_tlep, third: ETIS/global_qwm1x3x256, and bottom: LEAR/bow_sift_1000.

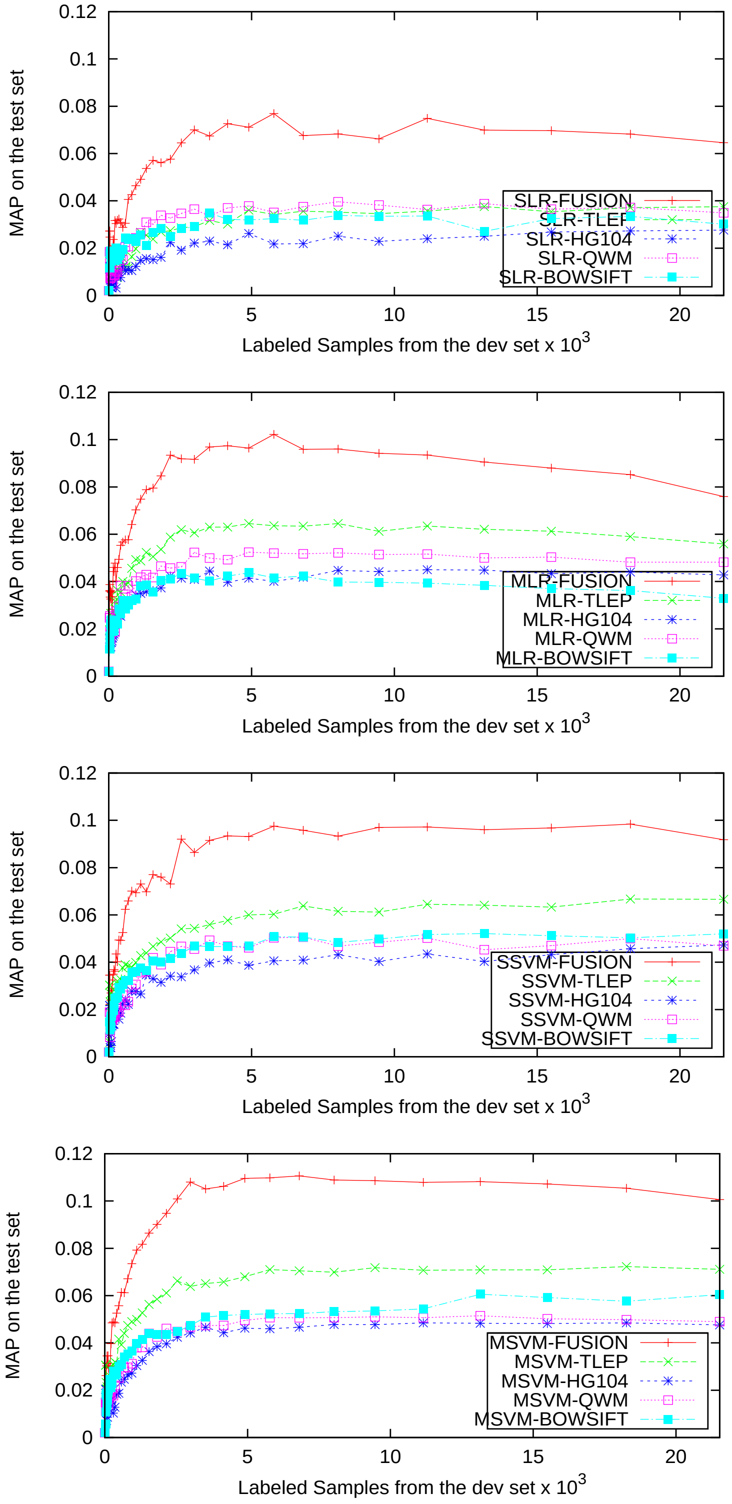


Figure 4.4: Combination of fusion and active learning with mono- and multi-learner approaches: top: LR mono-learner, second: LR multi-learner, third: SVM-RBF mono-learner and bottom: SVM-RBF multi-learner.

4.1.2.5 Descriptor Fusion

Until now, we have studied the combination of multi-learner and active learning approaches using only individual descriptors. The most efficient methods for concept classification actually use a number of descriptors. This can be done with a number of fusion strategies, among them early fusion, late fusion (Snoek *et al.* [2005b]) or kernel fusion (Ayache *et al.* [2007]). The performance of a system that combines several individual descriptors, is in general significantly higher than the performance of a system, which uses a single descriptor. The gain is more important when individual descriptors are of different types, such as color, texture and SIFT. In this case, the higher performance after using fusion is related to the best individual descriptor, even if there is a large disparity among the performance of the individual descriptors.

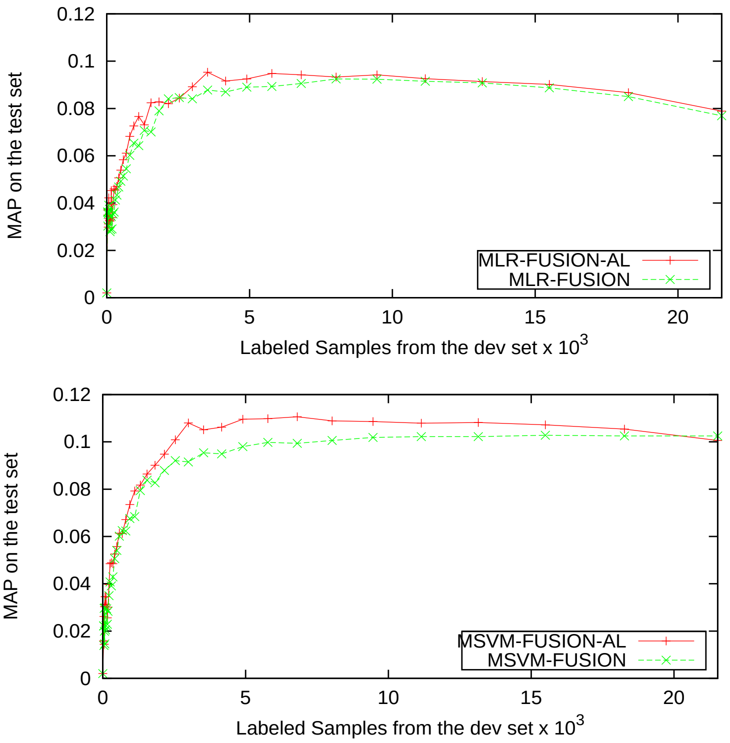


Figure 4.5: Performance of active learning when the fusion is done within the active learning process or separately: top: LR multi-learner and bottom: SVM-RBF multi-learner.

Fusion has been successfully combined with the multi-learner approach, as we showed in chapter 3. We have also evaluated it in combination with the multi-learner and the active learning approaches simultaneously. Figure 4.4 shows the behavior of active learning, using the late fusion of the four studied descriptors with mono and multi-learner approaches and with the two considered classifiers. We observe that:

- in all cases, the fusion significantly improves the performance of the active learning, like in classical learning;

- as for individual descriptors, the SVM-RBF classifier is better than the LR and the multi-learner approach is better than the mono-learner;
- the maximum performance is obtained when 10 to 15% of the dataset is annotated, which is less than for individual descriptors; this absolute value probably depends upon the size of the dataset, as observed by Ayache & Quénot [2007a].

Figure 4.5 shows the behavior of the active learning when fusion is applied directly within the active learning or separately. In the second case, active learning is performed separately for each descriptor and the fusion is applied on the resulting classifiers. As we can see from the plots, the inclusion of the fusion within the active learning improves both the speed at which the maximum performance is reached and the system performance. The effect is more significant in the case of the SVM-RBF classifier.

4.2 Incremental Method for Active Learning with Multi-Learner Approach (Inc-ALML)

As we have showed in the previous section, combining active learning with multi-learner approaches, increases significantly the effectiveness of the active learner. However, it makes the system also slow comparing to the single-learner approach. This makes a big challenge in the automatic annotation task, which mostly is directed by learning from users' feedback. Thus, we need to enhance our approach in such a way that it becomes faster and keeps a significant and efficient performance.

During each iteration of the active learning algorithm with a multi-learner approach (presented in algorithm 2), new labeled samples will always be added to the training set to be used in the next iteration. Multi-SVM are used here. Each iteration involves previous training information and new untrained samples. If we can re-use the previous information and learn the incremental information derived from new samples, the calculation time will be saved. Hence, it is useful to adapt an incremental learning algorithm in this setting.

4.2.1 Related Work

Some incremental learning methods focus on how to deal with the newly added training samples, and how to use the previous training results effectively to get better and faster classification results (Shilton *et al.* [2005]; Wu *et al.* [2009]; Zhang *et al.* [2009]). Shilton *et al.* [2005] proposed to re-train a model of SVM after adding a small number of additional training samples to the training set of an existing SVM model. The method of Wu *et al.* [2009] chooses the most important samples for the incremental learning to reduce the computational cost of the SVM incremental training. The method proposed by Zhang *et al.* [2009] first removes samples in the training process, which are independent of the support vectors, and then decomposes the remaining points into blocks to accelerate the next training process. Those methods check the Karush Kuhn Tucker conditions (Kuhn & Tucker [1950]) of SVM quadratic optimization problems for every sample. This involves more calculation time. An early stopping method is proposed by Bordes *et al.* [2005] to achieve faster convergence of active learning, by counting the number of support vectors derived from the previous training. If the number of support vectors stabilizes, it means that all possible SV's have been selected by the active learner. This method may loose useful information, because the number of SV's can change after several stable values, and the stability of SV's is not clearly defined. Rüping [2001] proposed an SVM based incremental learning method, in which the support vectors (SV's) from previous steps are involved with different weights in the learning process, together with the new labeled samples at each iteration. This method can work for balanced data sets. However, for highly imbalanced data sets, the weights are rough. Furthermore, because this method needs to train the previous support vectors, no calculation time will be saved. Wen & Lu [2007] proposed an incremental learning for SVM, which works by merging classifiers. Multi-SVM's are trained. The training set is divided into several learning batches, where each batch is used to train one classifier. The test samples are predicted by those classifiers, and their prediction scores are combined to give the final score for each test sample. In this method, the batches work independently. Moreover, they do not use any information from previous training steps.

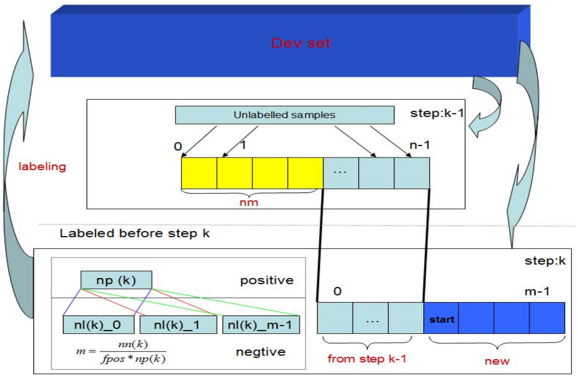


Figure 4.6: The framework of the proposed incremental method

In our method (ALML) presented in the previous section, the active learning with multi-learner approach uses, at each iteration, only part of the labeled samples for training each classifier. In this way, many classifiers are needed to be trained during each iteration. Although some samples have been well trained in previous steps, they may not be selected in the following steps. In the following, we will propose an incremental method for the active learning algorithm with multi-learner. This method attempts to reduce the calculation time by minimizing the number of classifiers needed at each iteration.

4.2.2 The Proposed Incremental Method

As described in section 2.6, algorithm 2 can be used to handle the class imbalance problem for multimedia indexing. The algorithm gave significant results for multimedia indexing. However, it generates many subsets and runs many classifiers at each iteration, which makes the system performance slow. In this section, we propose an incremental method to make algorithm 2 faster. The aim is to reduce the number of learners, which are needed (according to equation 3.1 in section 3.1.2.1) to be trained at each iteration of the active learner (Safadi *et al.* [2010a, 2011b]).

Let nl_k be the number of learners that are required at iteration k, we define nm_k to be the minimum number of learners to be learned at step k, given that $nm_k < nl_k$. The conditions to remove existing learners and to add new learners are presented in table 4.2. The method starts by calculating the nl_k and nm_k, then we need to define which learners from the previous iteration will be kept and which ones will be removed. Thus, the method calculates the parameter rm, which indicates the number of learners that will be removed from iterations $k-1$. This parameter is equal to $rm = nl_k - (nl_{k-1} - nm_k)$, if the number of learners at the current iteration is higher than the learners from the previous iteration. Otherwise, rm is equal to nm_k. The other important parameter is add, which defines the number of learners (that will be learned) at the current iteration. This number is equal to $add = nl_k - nl_{k-1} + nm_k$, and takes a minimum value equal to mn_k.

Figure 4.6 shows one step of the proposed algorithm. The method works as follows: first, at each iteration the algorithm removes the learners with the minimum number of positive samples

At step k:
nl_k : number of learners at step k
nl_{k-1}: learners trained from the previous steps
nm_k : minimum number of learners to be learned
The number of learners to be removed from the previous step:
$if(nl_k >= nl_{k-1}) \;\; rm = nm_k$
$if(nl_k < nl_{k-1}) \;\; rm = nl_{k-1} - nl_k + nm_k$
The number of learners to be learned:
$if(nl_k <= nl_{k-1}) \;\; add = nm_k$
$if(nl_k > nl_{k-1}) \;\; add = nl_k - nl_{k-1} + nm_k$

Table 4.2: The conditions of removing and adding learners.

(i.e. the learners taken from the oldest iterations). Then, the algorithm trains new learners, where each learner should be trained on a subset that consists of all the positive samples and of a comparable number of negative samples, which are randomly selected from the labeled set. Lastly, it applies a fusion function on the results obtained from the considered learners and it gives the final scores for the unlabeled samples. These scores will be used as in algorithm 2, and a new iteration will start.

In our experiments, we have fixed the minimum values of nm to be 1 and $nm_k = 20\% nl_k$.

4.2.3 Experiments

We have evaluated the active learning with multi-learner approach and the proposed incremental method in a variety of contexts. It has been applied using the four types of image descriptors used in the previous experiments (section 4.1.2). The SVM with RBF kernel was applied as a classifier, and the relevance sampling was used as a querying function for active learning. The harmonic mean function was used to fuse the results of the multiple classifiers. The cold start problem, the the number of samples to be added at each iteration, and the global parameters, like the f_{min} ratio and Fu in the ML model, were taken from our previous experiments. The evaluation was conducted on the TRECVid 2008 concepts, annotated on the TRECVid 2007 and 2008 collections (see table table 2.1).

As mentioned in the previous experiments, the training sets of the two collections are fully annotated; this means that nothing remains to be annotated. Thus, we will use the same conditions as in the previous experiments.

4.2.3.1 Incremental Active Learning Effectiveness

Figures 4.7 and 4.8 compare the effectiveness of the three methods (the single-learner, the multi-learner and the incremental method) using the four descriptors and the relevance sampling strategy. The performance of the single-learner is shown as a baseline method. These plots show the evaluation of the indexing performance of the test sets measured by the MAP metric, with the number of annotated samples at each step (in total 40 steps are used). As the plots in the mentioned figures show, the proposed incremental algorithm has achieved almost the same performance as that of the multi-learner. Both of them are significantly higher and reach the

	TRECVid 2007		TRECVid 2008	
Descriptor	$G_{I-S}(\%)$	$G_{I-M}(\%)$	$G_{I-S}(\%)$	$G_{I-M}(\%)$
LIG/hg104	14.77	2.34	6.50	1.83
CEALIST/global_tlep	12.84	0.62	22.42	-1.70
ETIS/global_qwm	4.76	0.73	1.20	0.02
LEAR/bow_sift	8.04	-3.16	5.22	0.75

Table 4.3: The gain of the system performance between the proposed incremental and the single- and multi-learners, with the four descriptors, evaluated on TRECVid 2007 and 2008.

highest value faster than the single-learner. With our incremental learning method, the highest performance can be reached when annotating only 15-30% instead of all the samples.

In further analysis, we consider the index of G_{a-b} to be the performance measure between two active learning curves (a and b). This measure was calculated as follows: $G_{a-b} = (A_a - A_b)/A_b$, where A_a and A_b indicate the area under curves (AUC) a and b respectively. This gain is calculated in two steps: first, the curves are normalized in each plot; then, the AUC is calculated for each curve using the following formula:

$$A = \frac{1}{2}\left|\sum_{i=0}^{n+1} x_i \times y_{i+1} - y_i \times x_{i+1}\right|$$

where n is the total number of iterations, and (x_i, y_i) indicates the number of annotated samples and the MAP value at iteration i. Table 4.3 shows the gain when using the incremental method compared to both the single and multi-learner methods with the two considered collections. As we can see, the gain is much higher and significant when using our incremental method compared to the single-learner, denoted as G_{I-S}, while the gain is very small compared to the multi-learner, denoted as G_{I-M}.

4.2.3.2 Execution Times

Table 4.4 gives the total execution times for the whole active learning process (40 iterations) on all 20 concepts on each experiment collection, per method and per descriptor, using the relevance strategy. As we can see, the single-learner is faster than the multi-learner and the incremental method. However, considering the performance of the single-learner described in the above section, its performance is much lower than that of the multi-learner. Compared with the multi-learner, the new proposed incremental method has saved nearly 48-66% of the processing time without loosing any performance.

Descriptor	TRECVid 2007				TRECVid 2008			
	Single	Multi	Inc	G	Single	Multi	Inc	G
LIG/hg104	1.40	20.63	7.64	66%	4.80	59.54	23.34	60%
CEALIST/global_tlep	23.90	115.02	64.17	52%	96.56	395.45	204.9	48%
ETIS/global_qwm	13.40	142.97	64.10	55%	45.67	460.60	212.3	54%
LEAR/bow_sift	43.42	162.18	79.16	52%	181.00	592.10	300.6	49%

Table 4.4: The execution times for the two evaluated collections, TRECVid 2007 and 2008, with G that indicates the gain of time using our incremental method compared to the multi-learners.

4.3 Summary

In this chapter, we have presented the active learning techniques for multimedia annotations. A new method for multimedia indexing and annotating, based on the combination of standard active learning and the multi-learner approach, is proposed. This method was evaluated on the corpus annotation and concept indexing task using the TRECVid 2008 dataset, which is a highly imbalanced datasets. The experiments were conducted using four different types of video shot descriptors, with two types of classifiers (logistic regression and SVM with RBF kernel) and with two different active learning strategies (relevance and uncertainty sampling). Results show that the multi-learner approach significantly increases the effectiveness of the active learner. On the considered dataset, the best performance for individual descriptors, is reached when 15 to 30% of the corpus is annotated, and for their fusion when 10 to 15% of the corpus is annotated.

Furthermore, the active learning with multi-learner approach has showed good performance for concept indexing in multimedia documents, especially in the case of highly imbalanced data. It involves however a large number of computations. We have proposed a new incremental active learning algorithm, based on multi-SVM for the annotations of multimedia document. Experimental results show that the best performance (MAP) is reached when 15 to 30% of the corpus is annotated; the new method can achieve almost the same precision while saving 50 to 63% of the computation time.

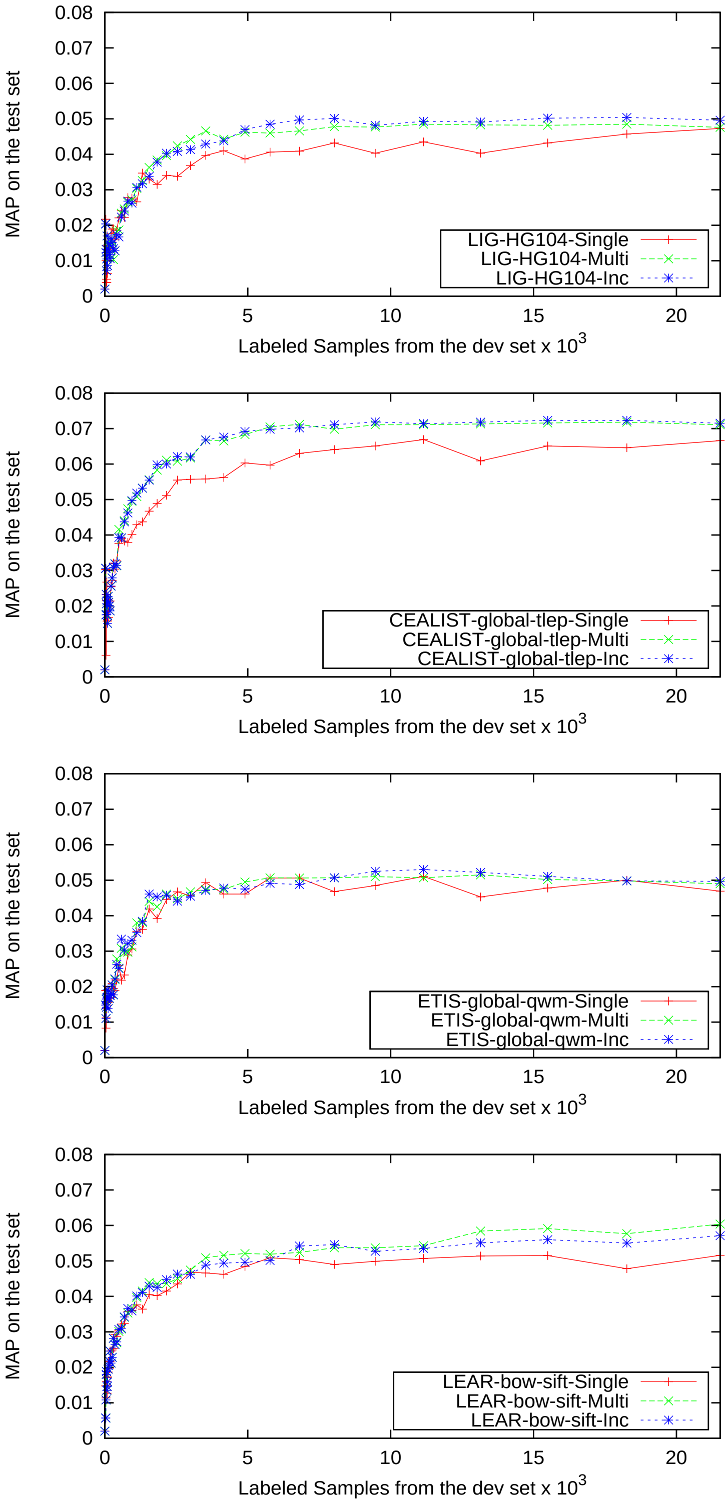


Figure 4.7: The MAP results on the TRECVid 2007 test collection evaluated on the four descriptors, each one of the plots shows the results using the single-learner (in red), the multi-learner (in green) and the incremental (in blue).

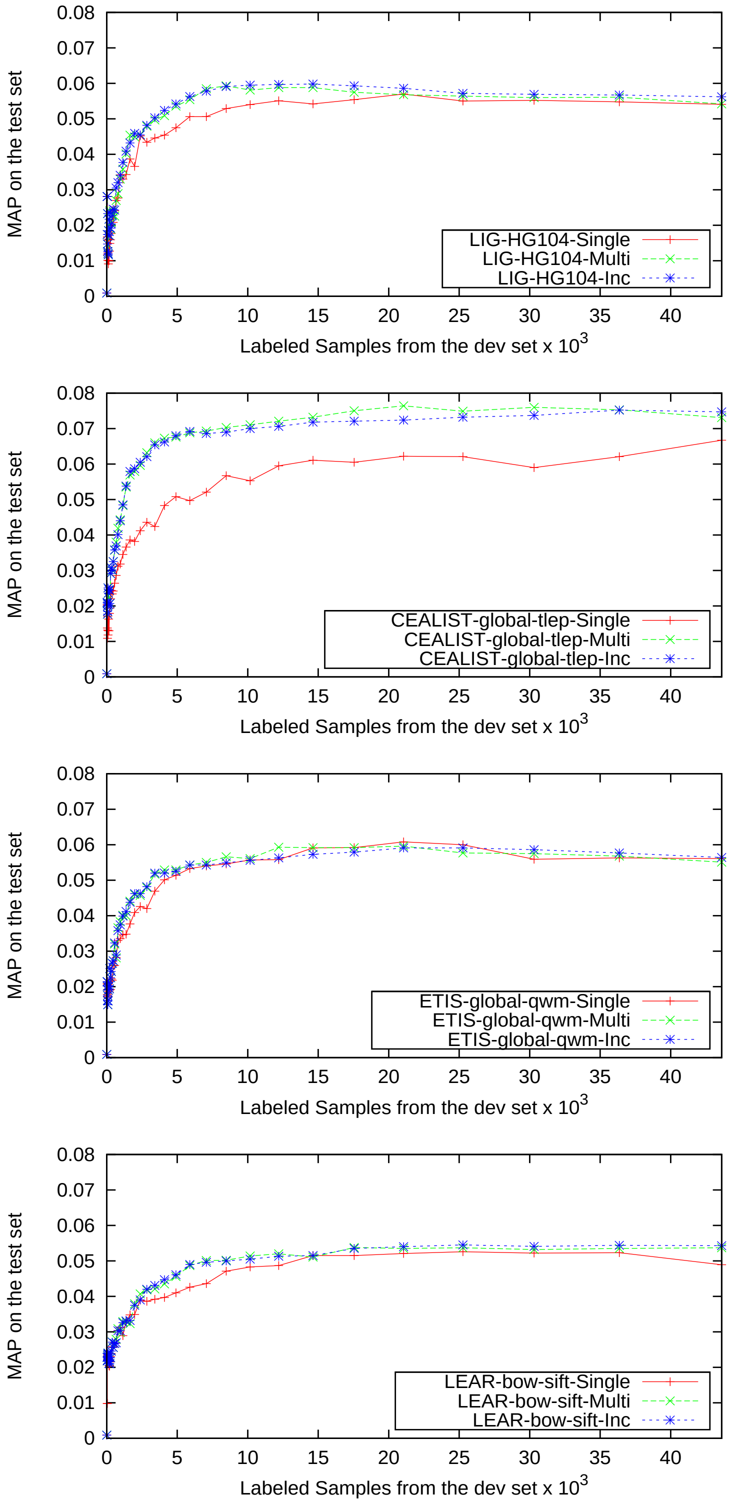


Figure 4.8: The MAP results on the TRECVid 2008 test collection evaluated on the four descriptors, each one of the plots shows the results using the single-learner (in red), multi-learner (in green) and the incremental (in blue).

Chapter 5

Active Cleaning For Multimedia Indexing

This chapter gives an overview on problem of noisy annotations and presents the collaborative annotation. It proposes an approach, denoted active cleaning, to reduce the noise in the annotations. It ends with results and conclusions derived from some experiments on the method.

5.1 Noisy Annotations

The quantity of the annotated samples is important for the performance of the system. Their quality is however very important as well, since most advanced classification methods are sensitive to mislabeled training samples. The use of crowd-sourcing methods (Howe [2006]; Snow *et al.* [2008]) leads to a quick change of the landscape for the quantity and quality of labeled data available for the supervised learner. While such data can now be obtained quicker and cheaper than ever before, the generated labels however tend to be more noisy, due to limitations of quality control mechanisms. The quality of the labels obtained from annotators varies. Some annotators provide random or bad quality labels in the hope that they will go unnoticed but will still be paid, and yet others may have good intentions but completely misunderstand the task at hand or they become distracted or tired over time.

The standard solution of the problem of noisy labels is to assign the same labeling task to more annotators, in the hope that at least a few of them will provide high quality labels or that a consensus emerges from a great number of labels. In either case, a large number of labels is necessary, and even though a single label is cheap, the costs can accumulate quickly. We can notice that if one is aiming to produce quality labels within minimum time and with minimum costs, it makes more sense to dynamically decide on the number of annotators needed. For instance, if an expert annotator provides a label, we can probably rely on it being of high quality, and we may not need more labels for that particular task. On the other hand, if an unreliable annotator provides a label, we should probably ask for more labels until we find an expert or until we have enough labels on which we can apply the majority vote to decide the final label.

Given the substantial human effort required to gather good training sets, as well as the expectation that more data is almost always advantageous, researchers have begun to explore new ways to collect labeled data. Both active learning and crowd-sourcing methods are promising ways to efficiently build up training sets for concept indexing and retrieval. The active learning techniques aim to minimize human efforts by focusing the labeling requests on those are the most informative samples for the classifier (Ayache & Quénot [2007a]; Joshi *et al.* [2009]; Qi *et al.* [2008]; Vijayanarasimhan & Grauman [2008]). On the contrary, the crowd-sourcing methods explore how to package annotation tasks in such a way that they can be dispersed effectively (Kumar & Lease [2011]; Sheng *et al.* [2008]; Vijayanarasimhan & Grauman [2008]). The interesting questions raised in these areas, such as how to deal with noisy labels, how to measure reliability or how to mix strong and weak annotations, make it clear that data collection is no longer an ordinary necessity, but a thriving research area in itself.

5.2 Related Work

Recent years have seen significant growth in label aggregation researches. Especially, by using the Amazon Mechanical Turk (mTurk) tool, which is a marketplace for recruiting online annotators. The idea behind using the mTurk lies in its ability to produce more annotations in a short time. The use of annotations obtained using the mTurk has been studied (Hsueh *et al.* [2009]; Kittur *et al.* [2008]; Kumar & Lease [2011]; Sheng *et al.* [2008]; Vijayanarasimhan & Grauman [2008]). For example, Vijayanarasimhan & Grauman [2008] presented an approach for live learning of object detectors, in which the system autonomously refines its models by actively requesting crowd-sourced annotations on images crawled from the worldwide web. Kumar & Lease [2011] showed that generating additional labels for labeled samples reduces the potential label noise. Moreover, faster learning can be achieved by incorporating knowledge of worker accuracies into consensus labeling. Sheng *et al.* [2008] presented repeated-labeling strategies of increasing complexity, and their results show clearly that when labeling is not perfect, a good strategy will be based on the selective acquisition of multiple labels. Moreover, for certain label-quality/cost systems, the benefit is substantial.

The use of multiple annotations to reduce labeling noise, has also been used in the context of crowd-sourcing. Producing a full double or triple annotation set is more costly than a simple full set. However, this is not in the spirit of data annotation based active learning approaches, in which we do not need to annotate all the samples in the dataset.

In the following, we propose the use of an active learning approach to select the samples for second or third annotations. We call this approach *active cleaning*.

5.3 Annotation Type

We consider the binary annotations, which are often used for image and video classification, such as ”Does the video shot contain an instance of the given visual concept C or not?”.

Let t_x be the target value for the sample x and y_{xk} the k^{th} label for the sample x given by an annotator. The set of target values T and the set of labels Y are binary scalars, thus $y_{xk}, t_x \in \{-1, 1\}$, which means that a label is positive when $t_x = 1$ and is negative when $tx = -1$. T values are decided by applying the majority vote on Y values. In the collaborative annotation we have a third case, which we call *skipped*: the user already saw the shot but he/she was not sure of its label. Three possible annotations are thus considered: *positive, skipped* and *negative*, we name them *pos, skip* and *neg* respectively. More details on the collaborative annotation system are given in section 6.1.

5.4 Our Active Cleaning Approach

Active cleaning (Safadi *et al.* [2012]) is the method of using an existing classification system to select samples for re-annotation, in order to improve the quality of an annotated corpus. It may be implemented in an incremental way, i.e. in conjunction with an active learning based annotation algorithm. In this case, the annotations may be cleaner and more correct, which makes the active learning more effective and efficient. Active cleaning may also be used for cleaning an already existing annotated data set, which can be either complete or partial. In this case, the benefits of the cleaning are only at the level of the resulting annotations, and the cleaning is applied only once.

Cleaning during active learning is the approach that was used in the collaborative annotation system. The active cleaning algorithm based on concept annotations, is detailed in algorithm 3. It implements the classical active learning algorithm, in which we added the cleaning process.

Let *D* be the data set, which needs to be labeled as containing a target concept (e.g. Airplane, Person, etc.); L and U are respectively the labeled and unlabeled subsets, thus $L \cup U = D$ and $L \cap U = \phi$. N indicates the set of possible choices of the oracle to label sample x as containing a given concept or not. Three possible choices are allowed by the annotation system: positive, skipped and negative, (see section 5.3). We denote Q_{al} and Q_{cl} to be the selection strategies of respectively the active learning and cleaning (see section 5.5). Before explaining the algorithm, we define four components of the algorithm in order to facilitate the understanding of it:

1. The set of available annotations: $Y = \{y_{xk} \in N : x \in L; k \in \{1, 2, \ldots, t\}\}$, where y_{xk} defines the k^{th} label of sample x given to an annotator. We ask, orderly, for up to three annotations for each sample, thus we set $t = 3$.

2. The subset of conflicting samples: $Conf\,ANN = \{x \in L : y_{x1}, y_{x2} \in Y \wedge y_{x1} \neq y_{x2}\}$, a subset of L, in which each sample has two different annotations.

3. The subset of second-annotations: $SANN_{Q_{cl}} = \{x \in L : y_{x1} \in Y \wedge y_{x2} \notin Y\}$, a subset of L, in which each sample has only one annotation, selected according to the cleaning strategy Q_{cl}.

4. The subset of primary-annotations: $PANN_{Q_{al}} = \{x \in U\}$, samples which are not yet annotated, selected according to the active learning strategy Q_{al}.

The algorithm is iterative. For implementation purposes, the elementary learning algorithm A is split into two parts: train and predict. The algorithm starts by initializing the L_0 set, which can be done by collecting initial labels Y_0 for some samples of D, through the annotators. Iteratively, the development set D is split into two parts: labeled samples L_i, and unlabeled samples U_i. Classifier A is trained using L_i with its associated labels Y_i. It obtains the model m_i, which is then used to predict the scores P_l and P_u of the samples in L_i and U_i sets respectively (each sample is scored as its likeliness to contain the target concept). These predicted scores are used to select the samples to be labeled in the next iteration. The selection is done in three steps: first, the algorithm chooses the samples with conflicting labels *ConfANN* (*). Then it applies the cleaning strategy Q_{cl} on the predicted scores P_l of the samples in L_i, and selects the samples of the *SANN* set to be re-annotated by different users (**). Finally, the predicted scores P_u of

Algorithm 3 Active Cleaning Algorithm Based Concept Annotations

```
$D$: all data samples.
$L_i, U_i$: labeled and unlabeled subsets of $S$, ($L_i \cup U_i = D$).
$A$=(train, predict): the elementary learning algorithm.
$Q_{al}, Q_{cl}$: the selection strategies, respectively, for the active learning and cleaning.
$Y_i$: available annotations for $L_i$.
Initialize $L_0$ and $Y_0$.
while $D \setminus L_i \neq \emptyset$ do
    $m_i \leftarrow$ Train($A$, $L_i, Y_i$)
    $P_u \leftarrow$ Predict($U_i$,$m_i$)
    $P_l \leftarrow$ Predict($L_i$,$m_i$)
    (*) Select the subset $ConfANN \subset L_i$
    (**) Apply $Q_{cl}$ on $P_l$ in order to select the subset $SANN \subset L_i$.
    (***) Apply $Q_{al}$ on $P_u$ in order to select subset $PANN \subset U_i$.
    $\tilde{Y}$ = (Label ($ConfANN$)) $\cup$ (Label ($SANN$)) $\cup$ (Label ($PANN$))
    $Y_{i+1} \leftarrow Y_i \cup \tilde{Y}$
    $L_{i+1} \leftarrow L_i \cup PANN$
    $U_{i+1} \leftarrow U_i \setminus PANN$
end while
```

the unlabeled samples in U_i are passed to the Q_{al} strategy, which selects the *PANN* set (***). The annotators are asked to annotate all the samples in these three sets, taking into account that a data sample x can be examined maximum once by the same annotator, and annotators cannot access the judgements of other annotators. When the new set of annotations $\tilde{Y}$ is completed, it will be added to the global annotations set Y. The set *PANN* is lastly added to the L_i set to produce the set L_{i+1}, while it is also removed from the U_i set to produce the U_{i+1} set. Then a new iteration will start.

5.5 Active Learning and Cleaning Strategies, Q_{al} and Q_{cl}

In the literature, we find several sampling strategies for active learning. The most famous strategies are the relevance and uncertainty sampling. From our previous experimental results (as described in section 4.1.2), we have observed that relevance sampling is a good strategy for sparse concepts, in which the objective is to find as many positive samples as possible from the unlabeled set U to be annotated. In general, the difference between the relevance and uncertainty sampling strategies is not that significant. In this chapter, our focus lies on the cleaning strategy, thus we did not evaluate our cleaning method using the different strategies for active sampling that are given in the literature. Therefore, we choose to implement only the relevance sampling for Q_{al}, which selects the most probable positive samples regarding their classification scores (i.e. samples with high prediction scores).

For active cleaning, several strategies Q_{cl} can be used for the selection of samples to be re-annotated. The choice among these strategies may relate to the type of annotations (e.g. the number of possible judgements) or the problem of highly imbalanced datasets, which is a very frequent case in video indexing. Furthermore, these strategies can depend on whether the first annotations were done incrementally or not. We propose here a cleaning strategy, denoted as *Cross-Val*. It is based on re-annotating the wrongly labeled samples due to an error of the annotator (for example, if the annotator missed the change of the concept to annotate). Detecting the wrongly labeled samples is done by training classifiers on these labeled samples and using the trained models to predict the correctness of these labeled samples. Thus, through the predicted score of each sample, we can expect if the sample has a correct label or not. The wrongly labeled samples are then those having positive labels with low scores, or negative labels with high scores. Basically, this strategy selects fractions of the labeled samples. These fractions are denoted as P%, N% and S% and refer to annotated samples as positive, negative and skipped respectively (see section 5.3). In addition, the selected samples are proposed to annotators for a second annotation round.

In the Cross-Val strategy, N%, P% and S% correspond to the percentage of the labeled samples as negative, positive and skipped. Three essential cases are derived from the Cross-Val strategy: the baseline is given when N=P=S=0 (i.e. no second annotations are proposed); *Skip-Pos* means re-annotating all skipped and positive samples, and is given when P=S=100 and N=0; *Full* is given when N=P=S=100, which means fully cleaning of all the annotated samples.

In this research, we evaluated the Cross-Val strategy with different fractions and several ways of re-annotations, as seen in table 5.1. By studying the system's performance with the Cross-Val strategy for cleaning annotations, we aim to find the best fraction values for this process.

5.6 A Posteriori Cleaning

In the case of a posteriori cleaning, we assume that the first annotations have been produced, thus we have one annotation for each sample. These annotations will be cleaned globally with a single iteration, as described in algorithm 4.

Algorithm 4 The Posteriori Cleaning Algorithm

```
    $D$: all the data samples.
    $A$=(train, predict): the elementary learning algorithm.
    $Q_{cl}$: the cleaning strategies.
    $Y$: the available annotations for $D$.
    $m \leftarrow$ Train($A$, $D, Y$)
    $P \leftarrow$ Predict($D$,$m$)
    Apply $Q_{cl}$ on $P$ in order to select the subset to be re-annotated $X \subset D$.
    $\tilde{Y}$ = Label ($X$)
    $Y \leftarrow Y \cup \tilde{Y}$
```

In this algorithm, a system is trained using the fully annotated set D and generated a model m. Each sample in D will be predicted using the model m, which will assign a predicted score to each sample. The samples are ranked according to their probability scores as being positive

Q_{cl}	E1	E2	E3	E4	E5	E6	E7	E8
pos %	0	10	0	0	5	10	20	100
neg %	0	0	0	10	5	10	20	0
skip %	0	0	10	0	5	10	20	100

Table 5.1: The P%,N% and S% fraction values that were used in our experiments with the active cleaning strategy.

samples. The Q_{cl} strategy (given fractions P%, S% and N% of samples) will be used to select the samples for a second annotation round. For the positive samples, the system chooses the P% of positive samples with a false prediction, which are the samples annotated as positive and have the lowest predicted scores. For the negative samples, it chooses the first N% of the samples annotated as negative and having the highest predicted scores. For the skipped samples, the system chooses the S% of the skipped samples that have uncertainty scores (predicted score is close to the classifier boundaries). In all cases, when conflicting is detected between the first and second annotations, a third annotation is required from the annotators. Finally, the algorithm updates the set of annotations to contain the new annotations.

5.7 Experiments

We have evaluated the active cleaning approach based on the Cross-Val (Q_{cl}) strategy in a variety of contexts. It has been applied with a classification system using four types of image descriptors, which are taken from IRIM GDR-ISIS partners (Quénot *et al.* [2009]), including the LIG/hg104, CEALIST/Global-Tlep, ETIS/Global-Qwm and LEAR/BoW-SIFT (see appedx. B). The multi-SVM classifier with RBF kernel (implemented ML as in section 3.1.2) was applied as the classification algorithm. The harmonic mean was used as a fusion function (Fu) in our model (ML). The evaluations were conducted using the TRECVid 2007 collection metrics and protocol (see section 2.7.1). In order to carry out the experiments on the simulated active cleaning, three annotations are needed for each concept in this dataset. The annotation of a data sample x as containing a concept c, is denoted as $c \times x$. We have collected and completed all the annotations, which were produced by the collaborative annotation on the considered database, so that we get at least two labels for each $c \times x$. In addition, we used a complete set of annotations: one label for each video shot, produced independently by a group from the Multimedia Content Group, Institute of Computing Technology, Chinese Academy of Sciences (MCG-ICT-CAS).

Since, our goal is to study the system performance with the Cross-Val (Q_{cl}) strategy for cleaning annotations, we present the different fractions that were used in our experiments in table 5.1, in which $E1$ is the baseline, $E8$ refers to the cleaning of all skipped and positive samples, and $(E2, E3, \ldots, E7)$ indicate the cross-validation strategy with different N%, P% and S%)fractions.

The cold start problem was solved by using another TRECVid collection, the 2005 one. We trained SVM classifiers on the TRECVid 2005 collection and predicted the usefulness on the development set of TRECVid 2007; we have started with annotating the first 100 samples at the top of the ranked list (i.e. samples with high predicted scores), after which the active learning and cleaning system was run to label all the shots within the development set. For the active learning steps, we have used the same conditions as in our previous experiments, which is presented in section 4.1.2.

5.7.1 Available Annotations

In the following, we present the two resources of the considered annotations:

1- Collaborative Annotations (CA): Annotations were done in collaboration with 32 groups of participants at TRECVid, each group contributed with several annotators. The annotation system used is based on the active learning approach. For each concept×annotation (i.e. $c \times x$) in the data set, the annotators have left the choice to label x as containing an instance of concept c or not, *pos* and *neg* respectively; they also can skip annotating it, in the case they are confused about its label. This can be considered as crowd-sourcing, since each shot could be proposed to several annotators to judge whether it contains c or not. Since we were limited in time for the annotating phase of TRECVid, this data set was not fully annotated. Furthermore, there are multiple annotations for the annotated samples L for each concept c. They are still available and can be used as multiple judgements for the experiments on the simulated active cleaning approach. For our experiments, these judgements have been completed to have at least two annotations for each sample.

2- MCG-ICT-CAS Annotations (MCG): The MCG-ICT-CAS team has produced, on its own, complete and independent annotations of all the concepts (i.e. $c \times x$). The annotations were made by a pool of students. Each student annotated shots to contain only a specific concept, and the annotations were done on the whole data set. However, active learning was not considered. Each $c \times x$ has only one label, since only one annotator (student) could examine and label it, which means that it does not contain multiple annotations. This set of annotations has the advantage of being complete, and since it was made using a smaller number of annotators, one can say it is probably more consistent.

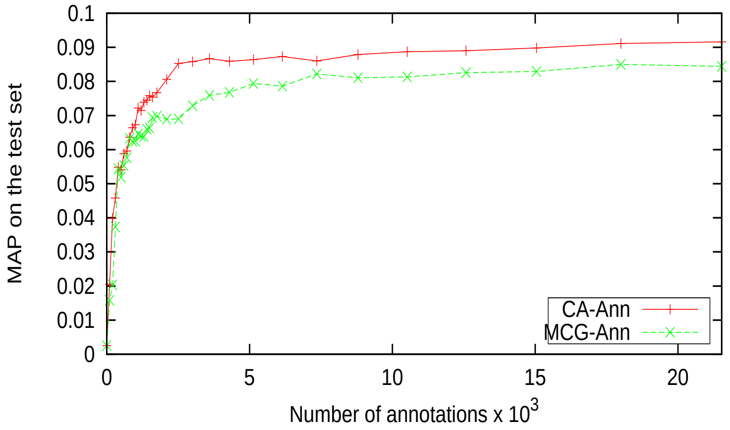


Figure 5.1: The MAP calculated on 20 concepts of the TRECVid 2007 test set, with two different annotation sources.

The annotations were produced by different annotators and within two different systems, and there is some noise in the annotations. This noise is the result of the used annotation systems and the annotators themselves. For instance, given a concept *Sports*, we got 482 positive samples from the CA annotations, while from the MCG annotations we got only 226 positives. In addition, the two sources agreed on only 168 positive samples.

The performance of our baseline system, by using only single annotations from the two annotation resources, is shown in figure 5.1. This figure shows the effectiveness in performance of the classification system with the number of the annotated sub-shots from the development set. It presents the MAP of the 20 concepts, calculated on the TRECVid 2007 test set. We consider a better curve to be the one which grows the fastest and which reaches, especially in the beginning, the highest MAP value. As we can see, the system performance that uses the annotations produced by the CA, is much higher than the one that uses the MCG annotations. This can be because of the annotation strategy, which is different in the two cases, or it may be related to the annotators themselves.

From this result, we assume that for each concept×annotation (i.e. $c \times x$), the annotations taken from CA are cleaner than the MCG. We planned two main experiments to study the effectiveness of the active cleaning strategies:

1. (MCG-CA): the first annotation, for each $c \times x$, is taken from low-quality annotators (i.e. MCG), and the second annotation was taken from better-quality annotators (i.e. CA).

2. (CA-MCG): the first annotation, for each $c \times x$, is taken from good-quality annotators (i.e. CA), and the second annotation was taken from lower-quality annotators (i.e. MCG).

In both experiments, we have used the second annotation produced by CA as the third annotation. It was also used in case of a conflict between the two annotations (CA and MCG).

5.7.2 Active Cleaning Effectiveness

We have studied the performance of the annotation system using the cleaning strategy, Cross-Val with different P%, N% and S% fractions as set in table 5.1. We report the obtained results from our two main experiments MCG-CA and CA-MCG.

For simplicity reasons, we report in table 5.2 the results of the last iteration of the active cleaning. Furthermore, in figure 5.2 we present the full iterative results of the cleaning performance for some experiments. Table 5.2 presents the evaluation results of the two main combinations MCG-CA and CA-MCG, using the cleaning strategy Cross-Val with different P%, N% and S% fractions as set in table 5.1. Moreover, it presents the number of cleaning annotations required for each experiment in the two considered combinations. As we can see from this table, some experiments do not have a real effect on the system performance, especially when the cleaning system does not include the negative samples, as in E2, E3 and E8. This is due to the fact that the number of re-annotated samples is very small, since there are few positive and skipped samples in the data set. However, the performance is higher when the negative samples were included in the cleaning system; it goes up to 15% in the case of MCG-CA and 5% in CA-MCG. This was expected, since, as shown in figure 5.1, we consider that annotations from MCG are of lower-quality than annotations from CA.

	MCG-CA	#Annotations	CA-MCG	#Annotations
E1=N0P0S0	0.084	21532	0.091	21532
E2=N0P10S0	0.084 +0%	+65	0.091 +0%	+46
E3=N0P0S10	0.086 +2%	+50	0.092 +1%	+11
E4=N10P0S0	0.095 +14%	+2100	0.096 +5%	+2150
E5=N5P5S5	0.096 +14%	+1100	0.095 +4%	+1100
E6=N10P10S10	0.097 +15%	+2200	0.090 -1%	+2215
E7=N20P20S20	0.097 +15%	+4400	0.095 +4%	+4420
E8=N0P100S100	0.086 +2%	+1150	0.093 +2%	+580

Table 5.2: The result of the cleaning strategies with the eight experiments described in table 5.1.

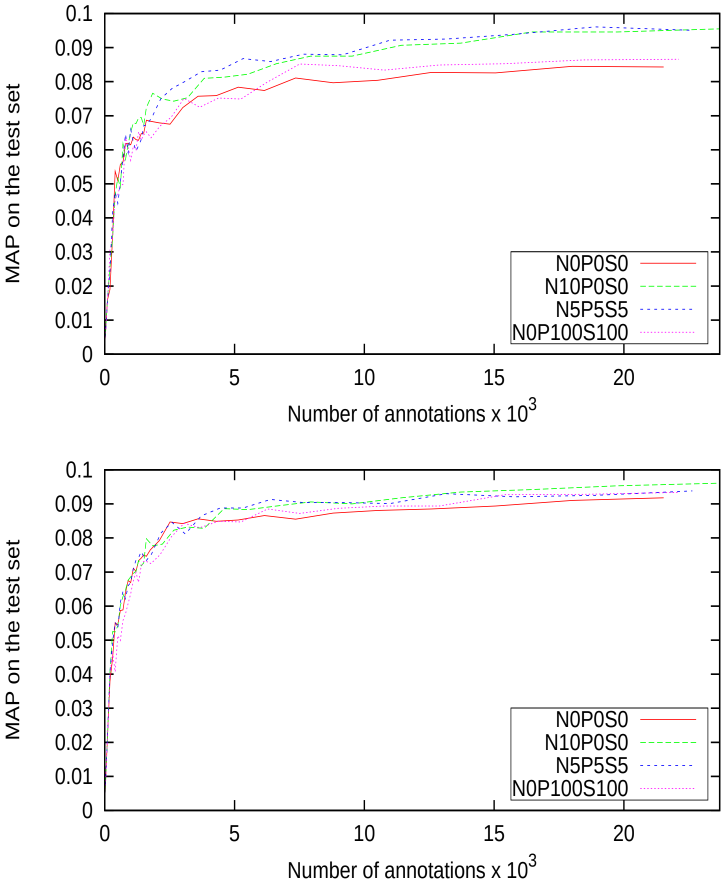


Figure 5.2: Active cleaning strategies: cleaning MCG annotations by CA in top, and in bottom cleaning CA by MCG annotations.

	MCG-CA	CA-MCG
E1=N0P0S0	0.0840	0.0910
E2=N0P10S0	0.0833	0.0917
E3=N0P0S10	0.0847	0.0927
E4=N10P0S0	0.0858	0.0917
E5=N5P5S5	0.0841	0.0921
E6=N10P10S10	0.0852	0.0910
E7=N20P20S20	0.0877	0.0921
E8=N0P100S100	0.0866	0.0931
Full=N100P100S100	0.0962	0.0962

Table 5.3: The result of the posteriori cleaning with the eight experiments described in table 5.1.

Figure 5.2 shows the effectiveness of the active cleaning strategies E4 and E5 compared to the baseline (E1) and the Skip-Pos (E8) strategy, with the two considered experiments, the MCG-CA (top) and CA-MCG (bottom). As we can see in this figure, in both experiments the system performance (using the MAP) was increased when the cleaning system considered the re-annotations of negative samples, as in E4 and E5. The Cross-Val strategy E4 works with re-annotating only 10% of the negative samples, and E5 with re-annotating 5% of each type of the annotations (negative, positive and skipped). Moreover, the active cleaning maintains the purpose of using the active learning approaches to annotate large scale image and video databases. Thus, the best performance could be obtained when annotating only 15-30% of the development set. The enhancement in the performance is more important when cleaning the lower-quality annotations than the better-quality annotations. Furthermore, the active cleaning can better enhance the performance when the number of annotations is the same.

5.7.3 A Posteriori Cleaning Effectiveness

The system performance results of the posteriori cleaning method are shown in table 5.3. The results are similar to the results obtained by the active cleaning approach. However, active cleaning is more effective and efficient. As the table shows, the use of the full annotations (i.e. the case of N100P100S100) leads to a better performance of the system than the use of other different fractions, as in table 5.1. However, this requires three times more annotations than the baseline, while each of the other combinations requires only a few more annotations than the baseline.

By comparing the results of the best performances of the system in the posteriori and active cleaning methods, we can conclude that the active cleaning method is more effective and robust than the posteriori cleaning method.

5.8 Summary

We have described the active cleaning approach that was used to complement the active learning approach in the TRECVid collaborative annotation. The actual impact of the active cleaning approach was evaluated on the TRECVid 2007 collection. The evaluations were conducted using complete annotations, which were collected from different resources, namely, the TRECVid collaborative annotations and the MCG-ICT-CAS annotations.

From our experiments, a significant improvement of the annotation quality was observed when applying the cleaning by cross-validation strategy, which selects the samples that need to be re-annotated. Experiments show that a higher performance can be reached with minimum double annotations of 10% of negative samples or 5% of all the annotated samples, which are selected by the proposed cleaning strategy, using the cross-validation. We have showed that, with an appropriate strategy, the use of a small fraction of the annotations for cleaning improves the system's performance much more than using the same fractions to add more annotations, as in the case of posteriori cleaning.

Chapter 6

Application to TRECVID

This chapter presents our system for video annotations applied to TRECVid (i.e. TRECVid 2010 and 2011). It also gives our results at the TRECVid 2011 and ImageCLEF 2011 evaluation campaigns.

6.1 TRECVid 2011 Corpus Annotation

In this section, we describe our application system for multimedia annotations. The system uses the active learning technique, which leads the users to annotate the unlabeled video shots of the corpus. The system was used in the collaborative annotations of the TRECVid 2010 and 2011 development sets. The goal was to produce as many coherent annotations as possible of the development sets, which the participants of TRECVid will use for the classification task. This should be done with the cheapest cost and within a short time (i.e. less than two months).

6.1.1 Semantic Indexing Task

A potentially important asset to help video search is the ability to automatically detect the occurrence of various semantic concepts (i.e. features), such as *Indoor, Outdoor, People, Speech, etc.*, which occur frequently in videos. The ability to detect concepts is an interesting challenge by itself but takes on added importance to the extent it can serve as a reusable, extensible basis for query formation and search. For the second year, our laboratory (UJF-LIG)[1] with the support of the Quaero[2] program, have co-organized (with NIST) the semantic indexing task at TRECVid. The semantic indexing task was a follow-on to the feature extraction task, and it had the following additional new objectives (Over *et al.* [2011]):i) to increase the number of semantic concepts most systems can extract and the number evaluated; ii) to support experiments using relations in a simple ontology among the concepts to be detected; iii) to offer a ”light” version of the task to encourage new participation.

A list of 500 target concepts has been produced, of which 346 have been collaboratively annotated by the participants and 50 have been officially evaluated at TRECVid. Section A.1 shows the 346 annotated concepts of TRECVid 2011.

[1] UJF: Université Joseph Fourier; LIG: Laboratoire d’Informatique de Grenoble.

[2] http://www.quaero.org

The 500 concepts are structured according to the LSCOM[1] hierarchy (Naphade *et al.* [2006]). They include all the TRECVid ”high level features” from 2005 to 2010, the CU-VIREO374 set (Jiang *et al.* [2008]) plus a selection of LSCOM concepts, so that we end up with a number of generic-specific relations between them. We enriched the structure with two relations, namely *implies* and *excludes*. For example, Single_Person implies Person and excludes Crowds. The goal was to promote research on methods for indexing many concepts and using ontology relations between them.

TRECVid provides participants with the following material:

- a development set that contains roughly 400 hours of videos, which forms the collection sets of TRECVid 2010;
- a test set that contains roughly 200 hours of videos;
- shot boundaries (for both sets);
- a set of 500 concepts with a set of associated relations;
- elements of ground truth: some shots were collaboratively annotated. For each shot and each concept ($x \times c$), four possibilities are available: the shot has been annotated as positive (it contains the concept), the shot has been annotated as negative (it does not contain the concept), the shot has been skipped (the annotator could not decide), or the shot has not been annotated (no annotator has seen the shot).

The goal of the semantic indexing task is then to provide, for each of the 346 annotated concepts, a ranked list of 2000 shots that are the most likely to contain the concept. The test collection contains 137,327 shots. A light version of the task has also been proposed in order to facilitate the access to small and/or new groups. More information about the organization of this task can be found in the TRECVid 2011 overview paper (Over *et al.* [2011]).

In the following, we first present the datasets of the TRECVid 2011, then we describe the active learning system that was used for the collaborative annotations. Lastly, we present the results of the collaborative annotations (the annotations of the development set and the assessments for the evaluation), which was given to the TRECVid participants to make the success of the TRECVid 2011 campaign.

6.1.2 Development and Test Sets

Data used in the TRECVid campaign are free of right for research purposes as it comes from the Internet Archive[2]. In total about 200 hours of TRECVid 2010 IACC[3] test data (i.e. IACC.1.A) and 200 hours of 2010 IACC training data (i.e. IACC.1.training) were available for system development (i.e. the development set of 2011). Approximately 200 additional hours of Internet Archive videos with Creative Commons licenses in MPEG-4/H.264 and with durations between 10 seconds and 3.5 minutes were used as new test data (i.e. IACC.1.B).

Table 6.1 provides the main characteristics of the collection set.

[1] Large Scale Concept Ontology for Multimedia: http://www.lscom.org.

[2] http://www.archive.org/index.php

[3] Internet Archive Creative Commons

Characteristics	**TRECVid 2011**
#videos	19, 856
Duration (total)	∼600 hours
min;max;avg ± sd	11s;1h;132s±93s
# shots	403, 800
# shots (dev)	266, 473
# shots (test)	137, 327

Table 6.1: Collection feature

The whole set of videos has been split into two parts, the development and the test set. Both sets were automatically split into shots using the LIG shot segmentation tool (Quénot *et al.* [2003]).

6.1.3 The Evaluation Measure

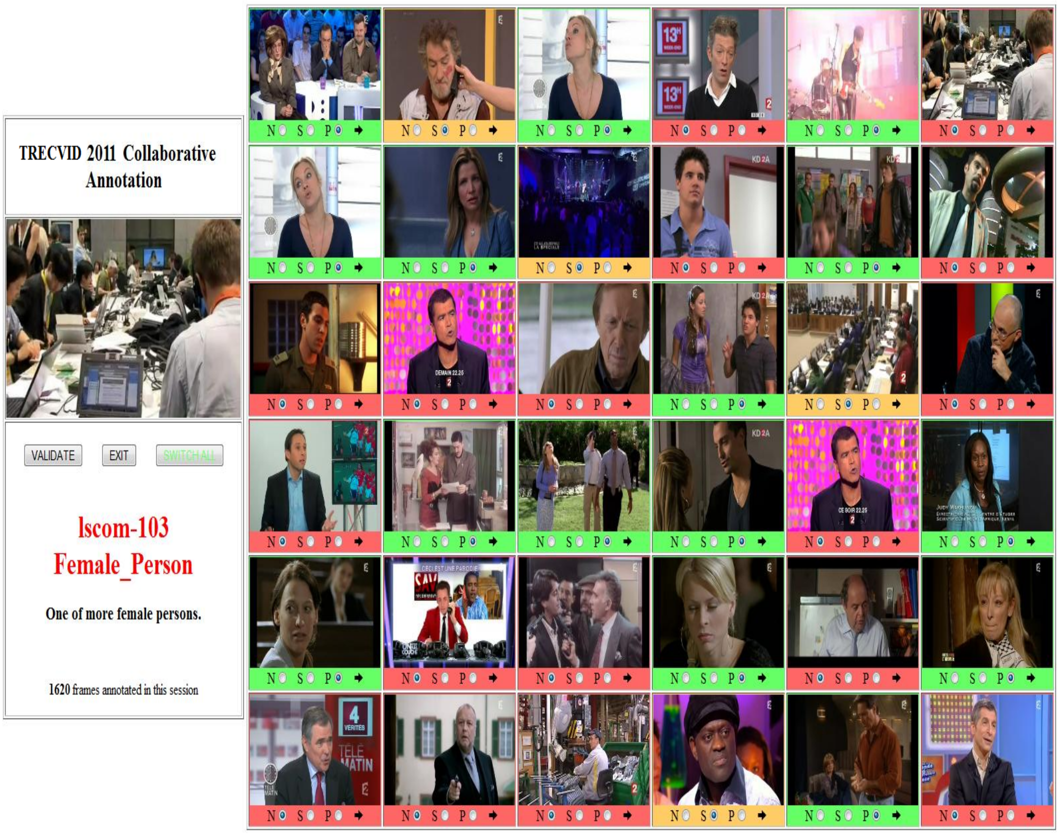
The evaluation measure used by TRECVid is the MAP (Mean Average Precision). Given the size of the corpus, the inferred MAP is used, as it saves human efforts and has shown to provide a good estimation of the MAP (Yilmaz & Aslam [2006]).

6.1.4 The Active Learning System

We (laboratory UJF-LIG) have organized a web-based collaborative annotation tool based on the work of (Ayache & Quénot [2007b]) in the spirit of what has been done at TRECVid since the year of 2007. Active learning has been used in order to simultaneously get the most useful information from the partial annotation and to significantly reduce the annotation effort per participant, like in the previous collaborative annotations. The active learner in the mentioned tool was improved, by using our multi-SVM with RBF kernel. In the latest iterations, the incremental method was used to make the learning phase faster. Moreover, active cleaning was applied to provide annotations of higher quality. The harmonic mean function was applied as the fusion function of the multi-SVM. Furthermore, we have used only two descriptors, which were normalized with the power-law. Their dimensions were reduced by the PCA in order to make the classification process faster (see section 3.3). Moreover, the simple fusion was used on the two results of the descriptors and finally our re-ranking method (see section 3.2.2) was applied after each classification step.

The TRECVid 2011 collaborative annotation system has been designed to be efficient and easy to use. Basically, we have used the same system, as in the TRECVid 2007 collaborative annotation. The system is a web based application, thus, no local software installation is required. Participation is restricted to groups that are registered as TRECVid participants and that have signed a license agreement to access the video data.

Figure 6.1 illustrates the web interface of our system. The system has two modes of operation: the sequential mode, in which the shots to annotate are displayed one by one, and the parallel mode, in which the shots are displayed in a two-dimensional grid. In the parallel mode, users can define the dimensions of the array in order to adapt visualization to the screen size.



TRECVID 2011 Collaborative Annotation

lscom-103
Female_Person

One of more female persons.

1620 frames annotated in this session

Figure 6.1: The TRECVid collaborative annotation system for the concept *Female_Person* (parallel mode).

Users are asked to annotate only one concept at a time. The system gives priority to the concept, which has the less annotated samples. For the current concept to annotate, images are displayed, either one by one or by group, depending upon the chosen mode. For each shot, the user has three choices for the annotation: i) positive (the concept is clearly there); ii) negative (the concept is clearly not there); iii) skipped (the user is not sure if the concept is there, whatever the cause of the uncertainty).

In the parallel mode, users see by default a shot at a smaller resolution than the video itself (160×120 instead of 352×288). By moving the mouse over one of the small shots, they can get an enlarged view of it in a corner of the screen. In both modes, users also have the possibility to play the whole video shot, if they feel that this will help them make a better decision. This is often the case for dynamic concepts (e.g. Walking, Running, etc.).

6.1.5 Cold-Start and Querying Strategy

Since most of the concepts to annotate in 2011 are new for TRECVid, it was a challenge to bootstrap the active learner for the new concepts. However, for the repeated concepts this issue was not the case. The 130 concepts of TRECVid 2010 are all included in 2011, thus, we used their annotations to continue the annotations of the same concepts in 2011. For the new concepts, the challenge how to start the annotations was considerable. One can consider several ways to start annotating new concepts. We present here three ideas for the cold-start: i) for each new concept, create a new sub-set from the web with positive and negative samples; ii) start the annotation randomly or linearly until some positive samples are found; iii) use concept ontologies. Considering the cold start problem, the first two strategies are costly and time consuming. In the first one, we need to search and build a relevant subset for each concept, which, however, may not be relevant to the samples of the development set that need to be annotated. The second one is a very time consuming strategy, due to the fact that we may annotate many negative samples before we get a positive sample. Taking this into account, we have decided to use the third strategy, in which we have bootstrapped a new concept from its (semantically) most similar annotated concept. For example, concept *Airplane* was used to bootstrap concept *Airplane_Landing*. When we have positive annotations for the new concept, the active learner will then use these annotations for learning.

We have applied the relevance strategy as the querying function for the active learner, since the goal is to find as many positive samples as possible, and in our previous experiments, the relevance sampling strategy proved its effectiveness .

6.1.6 Annotations of the Development Set

Shots in the development set have been collaboratively annotated by the TRECVid 2010 participants. As concept's density is most often very low, an active learning strategy has been set up in order to enhance the probability of providing relevant shots to annotators: the active learning algorithm takes advantage of previously done annotations in order to provide shots that will more likely be relevant. Although this strategy introduces a bias, it raises the number of annotated samples that are available to the classification systems. Moreover, it exhibits some trends in the concept's difficulty. For example, for the concept *Person*, the number of positive samples is larger than the number of negative samples. This means that the active learning algorithm was able to provide the annotators with more positive samples than negative ones, which means that the concept *Person* is probably ”too easy” to detect.

A total of about 4.2 M single concept × shots annotations were made, of which about 0.9 M by Quaero, about 2.2 M by the TRECVid 2010 participants and about 1.1 M by the TRECVid 2011 participants. Among these, about 88% were done at least once, about 9% were done at least twice and about 3% were done three or more times. The multiple annotations were selected by the active learning tool as those being the more likely to correspond to errors or ambiguities and they were made for cleaning as much as possible the annotations made. The resulting 4.2 M annotations were amplified to about 18 M usable annotations, by the use of propagation based relations between concepts. The used relations includes the ”implies” and ”excludes” relations, and the propagation using these two relations are done in the following four steps:

1. propagate positive samples from specific concepts to generic concepts using the ”implies” relations (e.g. positive for Cat implies positive for Animal)

2. propagate positive samples to negative samples using the ”excludes” relations (e.g. positive for Indoor implies negative for Outdoor);

3. propagate negative samples from generic concepts to specific concepts using the ”implies” relations (e.g. negative for Person implies negative for Male_Person);

4. take a majority decision for each annotated shot.

These resulted in more than one annotation per concept × shot in the original (direct) annotation; there may be even more annotations per concept × shot after the propagation takes place; all raw annotations are propagated and the multiple are also propagated; the majority decision is taken after all possible propagations were made. A transitive closure is implemented for the ”implies” relations.

These ∼18 M annotations represent about 13% of all the possible annotations on the development set. These have been selected by our active learning approach (as presented in chapter 4), which makes them almost as efficient as if the whole set was completely annotated.

6.1.7 Assessments

50 concepts were selected for evaluation out of the 346 concepts, for which participants were asked to provide results for the full SIN task. Assessments were done partly by NIST (20 concepts) and by Quaero (30 concepts). Assessments were done by visualizing the whole shot for judging whether the target concept was visible or not at any time within the shot. This makes the assessment task much slower than annotating the development set, in which the annotations were applied only on keyframes (about 2 to 3 seconds for each annotation). From experiments, it was shown that assessment annotation is about 10 to 20 times slower than the development shots.

A total of 268156 concept × shots assessments were made by NIST and Quaero. The frequencies of the shots with each concept (i.e. the evaluated concepts) is shown in figure 6.2, in which we can see that most of the concepts are still sparse (the number of positive samples is higher than the previous years). There are 13 concepts with frequency hits more than 5%. In addition, the figure shows the common concepts that were also evaluated in the context of TRECVid 2010.

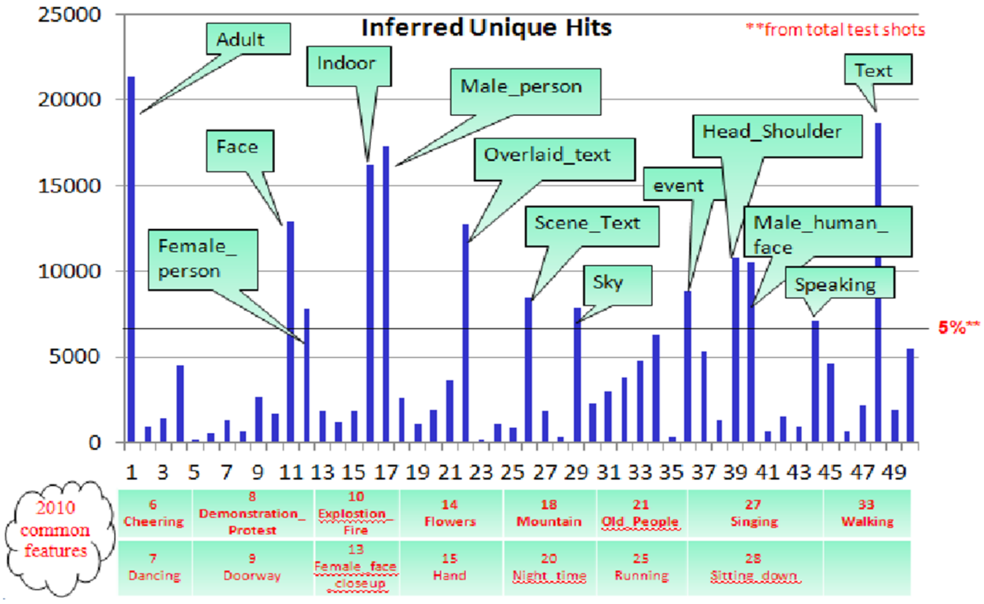


Figure 6.2: Frequencies of shots with each evaluated concept.

Coherent numbers of annotations were produced for 346 of the 500 concepts. The TRECVid 2011 campaign was then based on the annotations of these 346 concepts, produced using our system.

6.2 Participation in the Semantic Indexing Task

The TRECVid 2011 semantic indexing task is described in the TRECVid 2011 overview paper (Over *et al.* [2011]). Automatic assignment of semantic tags representing high-level features or concepts to video segments can be fundamental technology for filtering, categorization, browsing, search, and other video exploitation. New technical issues to be addressed include methods needed/possible as collection size and diversity increase, when the number of features increases, and when features are related by an ontology.

The task is defined as follows: ”Given a standard set of shot boundaries for the semantic indexing test collection and a list of concept definitions, participants were asked to return for each concept in the full set of concepts, at most the top 2000 video shots from the standard set, ranked according to the highest possibility of detecting the presence of the concept”. 346 concepts have been selected for the TRECVid 2011 semantic indexing task. Annotations on the development set were provided in the context of the collaborative annotation.

The classical approach for concept classification in images or video shots is based on a three-stage pipeline: descriptors extraction, classification and fusion. In the first stage, descriptors are extracted from the raw data (video, image or audio signal). Descriptors can be extracted in different ways and from different modalities. In the second stage, a classification score is generated from each descriptor, for each image or shot, and for each concept. In the third stage, a fusion of the classification scores obtained from the different descriptors is performed in order to produce a global score for each image or shot and for each concept. This score is generally used to produce a ranked list of images or shots that are the most likely to contain a target concept. We have tried to improve the performance of a generic classification system with the use of features, obtained from a face detection and categorization system. The original system uses a combination of low-level features, including color, texture, SIFTs (van de Sande *et al.* [2008]) and audio, and intermediate level features (Ayache *et al.* [2006]).

Our system uses a six-stages processing pipeline for computing scores for the likelihood of a video shot to contain a target concept. These scores are then used to produce a ranked list of shots that are the most likely to contain the target concept. The pipeline is composed of the following steps:

1. Descriptor extraction: a total of 47 audio, image and motion descriptors have been considered (section 6.2.1).

2. Descriptor optimization: a post-processing of the descriptors allows to simultaneously improve their performance and to reduce their size (section 6.2.2).

3. Classification: two types of classifiers are used as well as their fusion (section 6.2.3).

4. Fusion of descriptor variants: we fuse here variations of the same descriptor, e.g. bag of word histograms with different sizes or associated to different image decompositions (section 6.2.4).

5. Higher-level fusion: we fuse here descriptors of different types, e.g. color, texture, interest points and motion (section 6.2.5).

6. Re-ranking: we post-process here the scores using our re-ranking method, which based on the fact that videos statistically have a homogeneous content, at least locally (section 3.2.2).

In addition, our system includes a conceptual feedback, in which a new descriptor is built using the prediction scores on the 346 target concepts. This is added to the considered 47 audio and visual descriptors (section 6.2.6).

6.2.1 Descriptors

A total of 47 audio and visual descriptors have been used. Many of them have been produced by and shared with the IRIM partners, for more details on the descriptors please refer to appendix B. These include variants of the same descriptors (e.g. same methods with different histogram size or image decomposition). These descriptors do not cover all types and variants, but they include a significant number of different approaches including state of the art ones and more exploratory ones. They are described and evaluated in the IRIM partners paper Delezoide

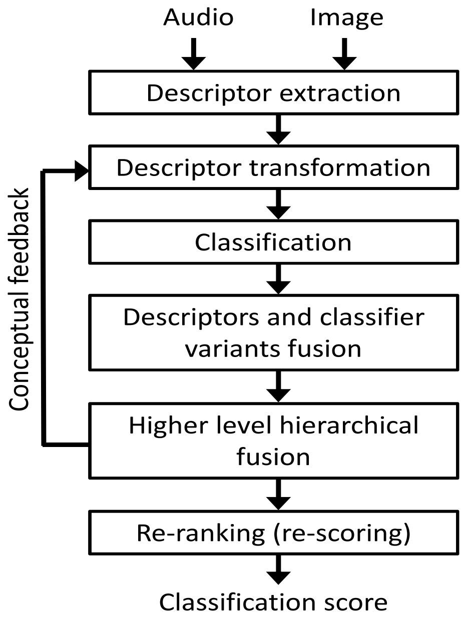


Figure 6.3: Semantic indexing system

et al. [2011]. They include color histogram, Gabor transform, quaternionic wavelets, a variety of interest points descriptors (SIFT, color SIFT, SURF, STIP), local edge patterns, saliency moments, percepts, and spectral profiles for audio description. Many of them rely on a bag of words approach.

6.2.2 Descriptor Optimization

Descriptor optimization consists of two steps: power-law normalization and the PCA-based dimensionality reduction. For the power-law normalization, the optimization of the value of the α coefficient is optimized by two-fold cross-validation within the development set. It is done in practice only using the LIG_KNNB classifier (see section 6.2.3), since it is much faster when a large number of concepts (346 here) has to be considered and since it involves a large number of combinations to be evaluated. Trials with a restricted number of varied descriptors indicated that the optimal values for the KNN based classifier are close to the ones for the multi-SVM based one. Also, the overall performance is not very sensitive to the precise values for this hyper-parameter.

The PCA was used to reduce the dimension of each descriptor, the number of components kept in the PCA reduction is also optimized by two-fold cross-validation within the development set, using the LIG_KNNB classifier. Also, the overall performance is not very sensitive to the precise values for this number.

6.2.3 Classification

We have used two types of classifiers on the contributed descriptors, as well as their combination.

LIG_KNNB : The first classifier is KNN-based. It is directly designed for simultaneously classifying multiple concepts with a single nearest neighbor search. A score is computed for each concept and each test sample as linear combinations of 1's (for positive training samples) and of 0's (for negative training samples), with weights chosen as a decreasing function of the distance between the test sample and the reference sample. As the nearest neighbor search is done only once for all concepts, this classifier is quite fast for the classification of a large number of concepts. It is generally less good than the SVM-based one, but it is much faster.

LIG_MSVM : The second one is based on a multi-learner approach with SVMs. This approach has shown its effectiveness to overcome the class-imbalance problem in large-scale datasets(see section 3.1.2).

LIG_ALLC : Fusion between the two available classifiers. The fusion is simply done by averaging the classification scores, which are produced by the two classifiers. Their output is naturally normalized in the [0:1] range. KNN computation is done using the KNNLSB package (Quénot [2008]). Even though the LIG_MSVM classifier is often significantly better than the LIG_KNNB one, the fusion of the two classifiers is often even better. This is probably due to the fact that they are very different, thus, they capture different things.

6.2.4 Fusion of Descriptor Variants

In previous work, we have introduced and evaluated the fusion of descriptor variants, which improved the performance of concept classification. We previously tested it in the case of color histograms, in which we could change the number of bins, the color space used, and the fuzziness of bin boundaries. We found that each of these parameters have an optimal value when the others are fixed and that there is also an optimal combination of them, which corresponds to the best classification that can be reached by a given classifier (KNN was used here) using a single descriptor of this type. We also tried the late fusion of several variants of such non-optimal descriptors. We found that most combinations of non-optimal descriptors have a performance which is consistently better than the individual performance of the best descriptor alone. This was the case even with a very simple fusion strategy, like taking the average of the probability scores. This was also the case for hierarchical late fusion. In the considered case, this was true when fusing consecutively according to the number of bins, to the color space and to the bin fuzziness. Moreover, this was the case even if some variants performed less well than others. This is particularly interesting because descriptor fusion is known to work well when descriptors capture different aspects of multimedia content (e.g. color and texture). However, an improvement is obtained here by using many variants of a single descriptor. That may be partly due to the fact that the combination of many variants reduces the noise. The gain is less than when different descriptor types are used, but it is still significant.

We have generalized the use of the descriptor variants fusion, and we have evaluated it on other descriptors and on TRECVid 2010. We made the evaluation on descriptors produced by the ETIS partner of the IRIM group. ETIS has provided 3×4 variants of two different descriptors (Delezoide *et al.* [2011]). Both these descriptors are histogram-based. They are computed with four different number of bins: 64, 128, 192 and 256, and with three image decompositions: 1x1 (full image), 1x3 (three vertical stripes) and 2x2 (2 by 2 blocks). Hierarchical fusion is done according to three levels: number of bins, ”pyramidal” image decomposition and descriptor type.

We have evaluated the results obtained from fusion within the same descriptor type (fusion levels 1 and 2) and between descriptor types (fusion level 3) as in (Gorisse *et al.* [2010]). The fusion of the descriptor variants varies from about 5 to 10% for the first level, and is about 4% for the second level. The gain for the second level is relative to the best result for the first level, so both gains are cumulated. For the third level, the gain is much higher. This was expected because, in this case, we fuse results from different information sources. The gain at level 3 is also cumulated with the gain of the lower levels.

6.2.5 Higher-Level Fusion

Hierarchical fusion with multiple descriptor variants and multi-classifier variants was used and optimized for the semantic indexing task. We made several experiments in order to evaluate the effect of a number of factors. We optimized directly the first levels of the hierarchical fusion, using uniform or average precision weighting. The fusion was made successively on variants of the same descriptors, on variants of classifiers, on results from the same descriptors, on different types of descriptors and finally on the selection of groups of descriptors.

6.2.6 Conceptual Feedback

System	Fusion	Rerank
Original fusion	0.1666	0.1833
Concepts descriptor	0.1144	
Fusion with concepts	0.1697	0.1864

Table 6.2: Cross-validation performance without and with conceptual feedback, with and without re-ranking

Since the TRECVid SIN 2011 task considers quite a large number of concepts (i.e. 346), and they are also organized according to a hierarchy, one may expect that the detection scores of some concepts help to improve the detection scores of related concepts. We have made a number of attempts to use the provided explicit *implies* or *excludes* relations. However, these were not successful so far, maybe due to a normalization problem between the scores of the different concepts. We tried then an alternative approach, using the implicit relations between concepts by creating a vector with the classification scores of all the available concepts. For this, we used the best hierarchical fusion result available. This vector of scores was then included as a 48^{th} one in the pool of the 47 already available descriptors. It was processed in the same way as the others, including the power and PCA optimization steps and the fusion of classifier outputs.

Table 6.2 shows the effect of including the descriptor *concepts* in the fusion process. Even though the performance of the descriptor alone is significantly less than the fusion, it can still yield a slight improvement.

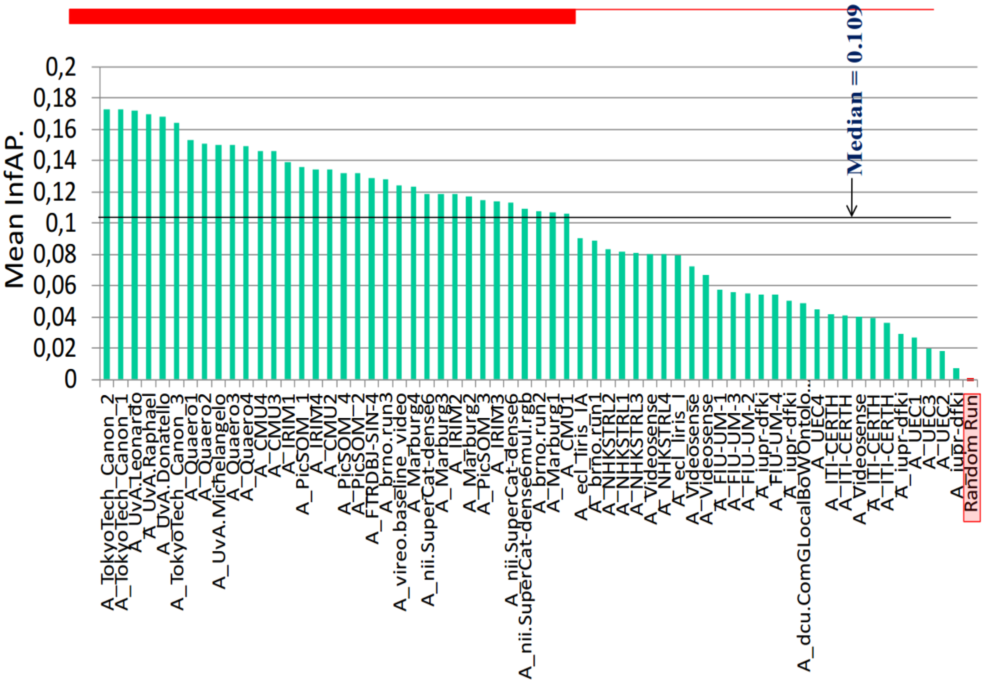


Figure 6.4: The official results on TRECVid 2011 SIN (full runs)

6.2.7 Performance on the Semantic Indexing Task

Four slightly different combinations of hierarchical fusion have been tried. The variations concerned the way the fusion was done: it can be flat or hierarchical, and the weighting of components can be uniform, MAP-based or optimized by cross-validation. Not all combinations could be submitted, thus, the following were selected:

F_A_Quaero1_1: Optimized hierarchical combination of all available descriptor × classifier combinations, including the concept score feedback descriptor;

F_A_Quaero2_2: Optimized hierarchical combination of all available descriptor × classifier combinations, excluding the concept score feedback descriptor;

F_A_Quaero3_3: Flat and uniform combination of available descriptor × classifier combinations, excluding the concept score feedback descriptor;

F_A_Quaero4_4: MAP weighted combinations of all available descriptor × classifier combinations, including the concept score feedback descriptor.

The official results of TRECVid 2011 SIN for all participants × runs are shown in figure 6.4. For more details, we present the performance of our four submitted variants in table 6.3. Our submissions (Quaero) ranked between 8 and 12 in a total of 68 for the full SIN task. Our best submission ranked us as the third group out of 19 for the full SIN task. The improvement brought by the conceptual feedback is quite small and less than what was expected from cross-validation within the development set, but it is significant. The hierarchical fusion performs better than the flat one, and the optimization of the fusion weights by cross-validation performs better than the MAP-based or uniform method.

System/run	MAP	rank
Best submission	0.1731	1
F_A_Quaero1_1	0.1529	8
F_A_Quaero2_2	0.1509	9
F_A_Quaero3_3	0.1497	11
F_A_Quaero4_4	0.1487	12
Median submission	0.1083	34

Table 6.3: MAP result and rank on the test set for all the 50 TRECVid 2011 evaluated concepts

6.2.8 Conclusion

In this section, we have presented our participation at TRECVid 2011 SIN task, in which we have applied our approaches for the classification task. This included, video description techniques, description normalization and dimensionality reduction, the KNN and the multi-SVM classifiers and the re-ranking approaches. In this campaign, we have obtained the third place out of 19 groups that participated in the full task.

6.3 Participation in the ImageCLEF Photo Annotation Task

This section describes our proposal and results at the Photo Annotation task for CLEF 2011. Our proposal focuses mainly on applying a late fusion on multiple learners based on multi-SVM approach. We have used our techniques as in TRECVid 2011, including the power-law normalization, dimensionality reduction based PCA and multi-MSVM for classification. Moreover, we made use of a simple integration with Flickr [1] tags.

The findings according to the official evaluations confirm that: the proposed approaches in this thesis can significantly enhance the image annotation task, and a simple integration of human assigned tags improves significantly the classification results. The ImageCLEF 2011 corpus (Mensink *et al.* [2010]) is composed of a training set of 8,000 images and the test set is 10,000 images large. The image annotation is a multiple classification process, where 99 labels (i.e. concepts) are given, these concepts are detailed in section A.2. They go from image elements (e.g. Flowers), to feelings generated by the images (e.g. scary). The images are possibly associated with EXIF[2] data, as well as with Flickr tags provided by humans. The main evaluation is the MAP-based, and we focus here only on this measure to evaluate our runs.

6.3.1 Extraction and Representation of Visual Descriptors

The features that were extracted are color-based as well as texture based. Some features are extracted globally from the whole image, and others are extracted from image regions, before being aggregated to represent one image. These descriptors were all generated by our research group (LIG) and they are named as follow:

Global descriptors: LIG/h3d64, LIG/gab40 and LIG/hg104.

Local descriptors: LIG/rgsift_har_ 4000; LIG/rgsift_dense_4000; LIG/opp_sift_har_1000; LIG/opp_sift_har
LIG/opp_sift_dense_1000; LIG/opp_sift_har_unc 1000; LIG/opp_sift_dense_unc_1000.

More details on the descriptors are given in appendix.B.

The descriptors were normalized using the power-law transformation, and a PCA-based dimensionality reduction was applied on the descriptors of large dimensions (the locale descriptors). The resulting features are generated using the same a power-law normalization with $\alpha = 0.500$ or$\alpha = 0.450$ (according to the 'pw' in the identifier) and PCA reduction to 400 dimensions, leading to: Rgsift_har_4000_pw0.500p400; rgsift_dense_4000_pw0.500p400; opp_sift_har_1000_pw0.450p400 and opp_sift_har_4000_pw0.450p400; opp_sift_dense_1000_pw0.450_p400
opp_sift_har_unc_1000_pw0.450p400; opp_sift_dense_unc_1000_pw0.450_p400.

For the "low dimensional" features LIG/h3d64, LIG/gab40 and LIG/hg104, similar techniques lead to h3d64_pw0.250p32, gab40_pw0.500p20 and hg104_pw0.375p54,when considering reducing the dimensions by a half.

[1] Flicker:http://www.flickr.com.

[2] Exchangeable Image File: camera settings and scene information are recorded by the camera into the image file.

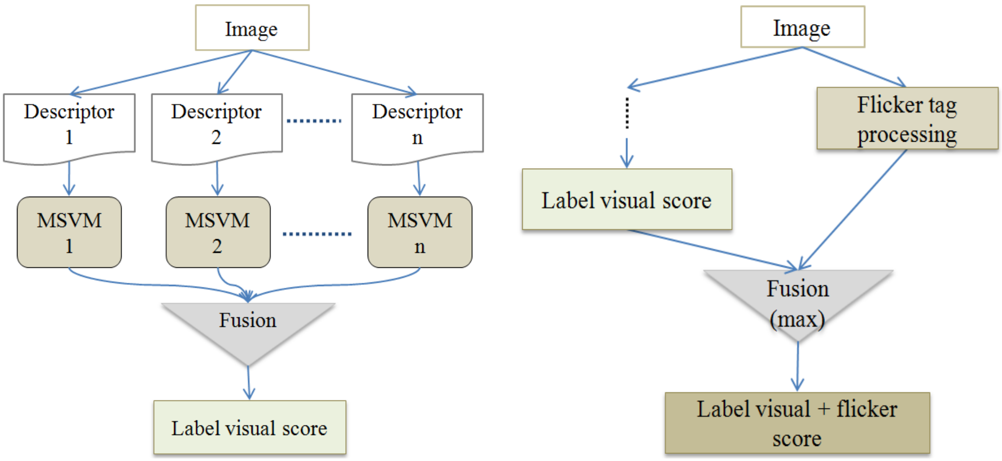
6.3.2 Extraction and Representation of Flickr Tags

As Image annotation collection is an excerpt from Flickr, the human-generated tags are available. We know that such manually input tags are not always easy to process (e.g. typos, jokes, etc.), but we propose a simple way to handle some of them. First, for each image, we split the tags into words, and we apply a Porter stemmer in a way to group similar words into classes. In a second step, if one stemmed tag equals one of the 99 stemmed labels, then the label is selected for the image. The resulting representation is a 99 dimensions binary vector, with 1 if the label describes the image and 0 otherwise.

6.3.3 Classification

6.3.3.1 Visual only

All the classification processes on the visual features use Multiple-SVM classifiers (i.e. MSVM) based on Radial Basis Function (RBF) kernels, since it was proved to be a good solution for data imbalance problems. Such problems occur for many labels in the collection under consideration here. So, for each label, we get positive and negative samples that are used as input for the learning of the Support Vectors. During the classification, each image representation is the input of the MSVM, and a binary classification is processed. For each image× label, a probability score of classification in [0,1] is obtained from each MSVM outputs. The final score for each image× label is then the average of each individual score from each classifier of the label, as shown in figure.6.5(a).



(a) Only visual descriptors (b) Visual + Flickr descriptors

Figure 6.5: The global classification process of CLEF.

6.3.3.2 Visual and Flickr Tags

For integrating Flickr tags and visual elements, we also use a late fusion approach. In this case the visual classification result for each label is fused using a max with the label value for the image according to the Flickr tags processing described earlier. The overall process is described in figure.6.5(b).

6.3.4 Validation set Results

We present the MAP results obtained on a validation set. Our training set is composed of $2/3^{rd}$ of the official training set, generated randomly with a post processing ensuring a similar distribution of the tags that on the official training set. This last point is important, especially for the labels that have only few samples (e.g. *Skateboard* has only 12 positive samples). The validation set is composed of the $1/3^{rd}$ remaining images of the official training set.

Descriptor	original	power-law+PCA (gain%)
opp_sift_har_1000	0.252	0.273 (+ 08.33)
opp_sift_har_4000	0.253	0.282 (+ 11.46)
opp_sift_dense_1000	0.255	0.272 (+ 06.67)
rgsift_har_4000	0.246	0.264 (+ 07.30)
rgsift dense_4000	0.258	0.270 (+ 04.70)
opp_sift_har_unc_1000	0.262	0.280 (+ 06.90)
opp_sift_dense_unc_1000	0.255	0.267 (+ 01.90)
h3d64	0.186	0.211 (+ 13.44)
gab40	0.213	0.215 (+ 00.94)
hg104	0.243	0.259 (+ 06.58)

Table 6.4: MAP results on the validation set for reduced feature representations.

The results obtained on the original descriptors and when considering the power-law normalization with PCA-based dimension reduction process are reported in table 6.4. We also show the gain (in percentage %) of using the power-law with PCA compared to the original (i.e., not reduced) descriptors. This table shows that the power-law with reduction of dimension proposed always outperforms the original descriptors. This result is especially visible with the opponent sift descriptors with strict assignment. In any case, it seems effective for Harris Laplace features, and less for dense sampling-based features. For the ”low dimensional” features, we notice also a large improvement with one half reduction, leading to very good results for the hg_104_pw0.375p54, which has only 54 dimensions.

The results obtained after fusing the results are given in table 6.5, according to the explanations of section 6.3.1. We chose three runs, which correspond to the configuration of the official run submitted of ImageCLEF:

1. MSVM: the late fusion of all the 20 visual descriptors considered earlier in section.6.3.1;
2. MSVM + tags: the late fusion of the visual scores and the Flickr tags scores;
3. MSVM (only two descriptors): the late fusion of the two best features according to table 6.4, but considering two different kinds of regions for the features, in practice the opp_sift_har_unc_1000pw0.450p400 and opp_sift_dense_1000pw0.450p400 descriptors.

Run identifier	MAP (increase vs. best visual feature in the fusion)
MSVM	0.314 (+ 11.35%)
MSVM +tags	0.357 (+ 26.60%)
MSVM (two descriptors)	0.297 (+ 06.07%)

Table 6.5: MAP results on the validation set for the three approaches with late fusion.

The conclusions drawn from this table is that the fusion always outperforms each of its components (such result is well known in the community). We see here that Flickr tags integration, even if the processing is quite straightforward, leads to an important increase of the results. These three configurations are the ones used for the official submissions.

6.3.5 Official Results

We present here the official MAP results obtained from our runs in table 6.6. This table shows also in the last column the rank obtained in comparable lists (i.e. list of visual results for MSVM and MSVM (two descriptors), and list of multi-modal results for MSVM tags). The results obtained place our best visual run, MSVM with a MAP of 0.336, in the first tier of the list, and above the average and the median values respectively of 0.289 and 0.323. For the multi-modal run, MSVM + tags with a MAP of 0.378, the rank is above the middle, and also above the average and the median values of respectively 0.370 and 0.371.

Run identifier	MAP	rank (in comparable list)
MSVM	0.336	15/46
MSVM + tags	0.378	11/25
MSVM (two descriptors)	0.324	23/46

Table 6.6: Official MAP results for our submitted runs at CLEF 2011.

6.4 Summary

In this chapter, we have presented our real-world application system for multimedia annotations. The system was used to lead the annotations of the development set of TRECVid 2011. We have used the active learning with multi-SVM approach and the relevance sampling strategy, since the goal was to provide as many positive samples as possible. The quality of the annotations was enhanced by the active cleaning approach. The system was run during three weeks on 500 concepts, in which 40 laboratories participated in the collaborative annotation task. Coherent numbers of annotations were produced for 346 of the 500 concepts. The TRECVid 2011 campaign was then based on the annotations of these 346 concepts, produced using our system.

Moreover, We have presented our participation at TRECVid 2011 SIN task (video indexing) and Photo Annotation task for CLEF 2011 (image annotation), in which we have applied our approaches for the classification tasks. In the TRECVid 2011 campaign, we have obtained the third place out of 19 groups that participated in the full task. In image CLEF 2011, our results placed us in the first tier for the visual runs, and in the first half for the multimedia runs.

Chapter 7

Conclusions And Perspectives

The general framework of this work is semantic indexing and information retrieval, applied to multimedia documents. More specifically, we are interested in the semantic indexing of concepts in multimedia documents (e.g. images and videos) by the active learning approaches that we use to build annotated corpus. Through out this thesis, we have seen that the main difficulties of this task are often related, in general, to the semantic-gap. Furthermore, they are related to the class-imbalance problem in large scale datasets, where concepts are mostly sparse. For corpus annotation, active learning algorithms attempt to overcome the labeling bottleneck by iteratively selecting the most informative samples to be manually labeled by annotators. Their main objective is to increase the system performance by using as few labeled samples as possible, thereby minimizing the cost of labeling data (e.g. money and time). The efforts toward building multimedia indexing systems based on active learning, consist of several levels: multimedia document description, classification, fusion and the active selection strategies.

In this thesis, we have contributed in the mentioned levels and proposed some approaches that outperform state-of-the-art systems. The proposed methods were validated through several experiments, which were conducted and evaluated on large-scale collections of the well-known international benchmark, called TRECVid.

7.1 Contributions

In the state-of-the-art, we have discussed the major components of multimedia indexing approaches. Most approaches aim to overcome the semantic-gap by extracting low-level descriptors followed by classification based on supervised learning approaches. We have explored several low-level descriptors, which can be extracted from different modalities (e.g. visual and audio) of multimedia documents. It is recommended to use descriptors that discriminate the different modalities in order to improve the indexing system. It further takes advantages of fusing systems based on these descriptors. The fusion can be done either at the beginning (i.e. early) or at the end of the indexing process (i.e. late).

Classification is a necessary phase for indexing documents with concepts, however, it can only solve the problem of bridging the semantic-gap. There exist different classification methods, including generative and discriminative algorithms. The state-of-the-art shows that, in multimedia indexing the discriminative approaches are more effective and outperform the generative ones. The most used classifier is the Support Vector Machines (SVM), which allows the use of kernel spaces. We have also reviewed different scenarios of the use of active learning approaches. Furthermore, we have introduced three well-known strategies that are used with the active learning approaches, to select the new samples to be annotated. In the following, we present the conclusions derived from our contributions.

7.1.1 Indexing Systems

We have contributed in three main stages of the multimedia indexing systems:

- We have proposed and evaluated an approach to handle the class-imbalance problem in large-scale multimedia indexing, denoted as *multi-learner* (ML). The basic idea is to select several subsets from the majority class and combine them with the minority class samples to build several training sets. One classifier is trained on each training set, thus several models are obtained. For indexing, the samples are predicted by these models and the results fused to produce final scores. We have showed that the multi-learner approach is designed to better address the problem of sparse concepts, which leads to a strong numerical imbalance between classes. The evaluation have been done in the context of the concept detection task of the TRECVid 2008 and 2009 campaigns. It was evaluated on several types of descriptors and tested with three types of classifiers: linear SVM, SVM with Gaussian kernels and logistic regression. We showed that the multi-learner approach significantly outperform the single-learner approach. In this context, the classifier based on logistic regression works better than the linear SVM classifier, but less well than the SVM classifier with Gaussian kernel. Moreover, the obtained results were fairly stable against the type of the used descriptor.

- We have proposed a re-ranking method that improves the performance of semantic video indexing and retrieval. The algorithm works on re-evaluating the scores of the shots using the homogeneous nature of the video they belong to. In other words, the score of each shot is smoothed by its neighbour shots within the same video. The proposed method was evaluated and validated on a non-homogeneous and a homogeneous content dataset, namely TRECVid 2008 and 2010 respectively. The experimental results showed that the proposed re-ranking method was able to significantly improve the performance of the system. The system was improved by about 11-13% and 18%, respectively, on non-homogeneous and homogeneous videos.

- We have evaluated the power-law normalization and compared it to other four typical normalization techniques. The evaluations were conducted on the TRECVid 2010 collection, using different types of descriptors and the MAP as the evaluation metric. The multi-SVM with the Gaussian kernel approach was used as a classifier with either the Euclidean or Chi-square distance. The power normalization was also evaluated after applying the PCA techniques for dimensionality reduction. Experimental results showed that the normalization based power-law is the best to consider, among the others, especially with the Euclidean distance. The power transformation makes the SVM with Euclidean distance as effective as the Chi-square. Moreover, by applying the power-law normalization with

Euclidean distance and PCA-based dimensionality reduction, a further computation time reduction was observed, however, the system performed as good as without PCA.

7.1.2 Active Learning and Cleaning

We have also contributed and proposed different approaches for applying the active learning for annotating multimedia corpus. This includes methods that improve the system performance and the quality of the annotations.

- We have proposed a new method based on the combination of a standard active learning and the multi-learner approach for multimedia indexing, denoted as ALML. Our fundamental idea is to apply the multi-learner as a classifier for the active learner. This method was evaluated on the corpus annotation and concept indexing task using a highly imbalanced dataset, namely the TRECVid 2008. We have conducted the evaluation using four different types of video descriptors, with two types of classifiers (i.e. logistic regression and SVM with RBF kernel) and with two different active learning strategies (i.e. relevance and uncertainty sampling). Experimental results showed that the multi-learner approach significantly increases the effectiveness of the active learner. On the considered dataset, the best performance for individual descriptors, is reached when 15 to 30% of the corpus is annotated, and for their fusion when 10 to 15% of the corpus is annotated.

- We have extended the proposed ALML approach, and proposed a new incremental active learning algorithm, based on multi-SVM, denoted as Inc-ALML. The ALML approach has showed good performance, especially in the case of highly imbalanced datasets. It involves however a large number of computations. We have evaluated the Inc-ALML approach on the same conditions as in the previous experiments (ALML). The experimental results showed that the system achieved almost the same performance as ALML, while the new approach saved 50 to 63% of the computation time.

- We have presented the *active cleaning* approach, which was used to complement the active learning approach in the TRECVid collaborative annotation. We proposed the *cross-val* strategy, which selects fractions of the samples that need to be re-annotated (wrongly predicted samples). We have evaluated active cleaning approach on the TRECVid 2007 collection, with complete annotations (three annotations per sample), it was applied with different fractions for the cross-val strategy. Experimental results showed that a higher performance can be reached with minimum double annotations of 10% of negative samples or 5% of all the annotated samples, which are selected by the proposed cleaning strategy. Moreover, with an appropriate strategy, the use of a small fraction of the annotations for cleaning improves the system's performance much more than using the same fractions to add more annotations, as in the case of a posteriori cleaning.

- We have presented our real-world application system for multimedia annotations. The system was used to lead the annotations of the development set of TRECVid 2011. We have used the active learning with multi-SVM approach and the relevance sampling strategy, since the goal was to provide as many positive samples as possible. The quality of the annotations was enhanced by the active cleaning approach. The system was run during three weeks on 500 concepts, in which 40 laboratories participated in the collaborative annotation task. Coherent numbers of annotations were produced for 346 of the 500 concepts. The TRECVid 2011 campaign was then based on the annotations of these 346 concepts, provided by our system.

- We have presented our participation at TRECVid 2011 semantic indexing task (SIN), in which we have applied our approaches for the classification task. In this campaign, our best evaluation score was 0.1529 (in MAP) that ranked us at the third place out of 19 groups, which have participated in the full task.

7.2 Perspectives

Apart from improving the performance of multimedia indexing systems, including the descriptors, the classification and the fusion techniques, we point three major challenges of which the work conducted in this thesis may be continued in.

7.2.1 The Cold-Start

The cold-start issue is defined as the construction of the initial training set, which will bootstrap the active learner in annotating a data-set for a given concept. As we discussed in this thesis, traditionally, an active learner can be bootstrapped by randomly selecting a subset, which contains few samples of the considered classes. However, it is difficult to generalize this situation in real world applications.

Another way to do this, is by training a classifier on external datasets and predicting the samples of the given data set. The problem becomes crucial when there is no annotations at all, even from different collections, thus, these techniques can not be of useful. The concept can be annotated by using other related concepts, for example, we can start the annotations of concept *Golf* from the positive samples of concept *Sports*. Therefore, the use of a good and complete set of ontologies is needed. However, this is still a very big challenge.

The cold-start issue in multimedia annotations can also be considered as the zero-data learning problem (Larochelle *et al.* [2008]) in machine learning. In which the algorithm should build models from zero-labeled data. The problem lies to build a model for classifying a new concept, which then can be used to bootstrap the annotations of the concept.

7.2.2 Active Learning on Very Large-Scale Datasets

As we have shown in this thesis, in a pool-based active learning approach for multimedia indexing, the goal is to request labels for those samples that appear to be most informative. The widely used approach for finding those informative samples is by generating a trained model, which is trained using a classifier (e.g. SVM). For each sample, the generated model is used to predict its probability score, and the most informative samples are found by ranking the samples, according to their predicted scores and the selection function (i.e. Q) of the active learner. However, when applying the active learner on databases of very large-scale, the predicting time of the all samples is crucial. Thus, it is impractical to exhaustively apply the prediction to all the unlabeled samples at each iteration. Thus, to exploit substantial unlabeled pools, a fast search method is needed. One solution for this issue is to apply the prediction, at each iteration, on only part of the unlabeled samples. Therefore, the main challenge is: how to select the samples to be predicted by the trained model?

Recently, Gorisse *et al.* [2012] have presented a new Locality Sensitive Hashing (LSH) scheme adapted to the chi-square distance for approximate nearest neighbors search in high-dimensional spaces. This method is based on sup-sampling the unlabeled samples. The authors have shown the effectiveness of the method with the Euclidean LSH algorithm in the context of image retrieval on real image databases. We aim to test their method in the context of using the active learning on very large-scale databases.

7.2.3 Crowd-sourcing: Annotations Quality and Annotators Surveillance

Employing crowd-sourcing techniques for multimedia indexing task, is one of the major perspectives of this thesis. With efficiency and cost-effectiveness, online recruitment of annotators (e.g. mTurk) brings new challenges. The annotators are not specifically trained for annotation, and might not be highly invested in producing good-quality annotations, since most of them do this only to get paid. Consequently, the obtained annotations may be noisy by nature, and might require additional study. Several interesting questions immediately arise in how to optimally use annotations in this setting: How can we handle differences among workers in terms of the quality of annotations they provide? How can we find and control noisier annotators? How useful are noisy annotations for creating indexing systems? Is it possible to identify ambiguous examples via annotator disagreements? These questions are strongly connected to active learning, where we the goal is to produce less annotation but with higher quality. The active cleaning approach, given in this thesis, is considered as a small step in this direction. It was applied to a large dataset but it still needs to be verified and used with huge datasets and more annotators, for instance, using it with mTurk.

Appendix A

Appendix A

A.1 TRECVid Concepts

Here we present the concepts of the TRECVid 2010 and 2011, which will also be evaluated in 2012. In the TRECVid 2010 there are 130 concepts, whereas in 2011 the task was enlarged, however we succeeded to annotate 346 concepts over 500. The concepts that were used in TRECVid 2010 are the first 130 concepts in the following list.

1. Actor 2. Adult 3. Airplane 4. Airplane_Flying 5. Anchorperson 6. Animal 7. Asian_People 8. Athlete 9. Basketball 10. Beach 11. Beards 12. Bicycles 13. Bicycling 14. Birds 15. Boat_Ship 16. Boy 17. Bridges 18. Building 19. Bus 20. Canoe 21. Car 22. Car_Racing 23. Cats 24. Celebrity_Entertainment 25. Chair 26. Charts 27. Cheering 28. Cityscape 29. Classroom 30. Computer_Or_Television_Screens 31. Computers 32. Conference_Room 33. Construction_Vehicles 34. Corporate-Leader 35. Court 36. Cows 37. Crowd 38. Dancing 39. Dark-skinned_People 40. Daytime_Outdoor 41. Demonstration_Or_Protest 42. Desert 43. Dogs 44. Doorway 45. Driver 46. Eaters 47. Emergency_Vehicles 48. Entertainment 49. Explosion_Fire 50. Face 51. Female_Person 52. Female-Human-Face-Closeup 53. Flowers 54. Girl 55. Golf 56. Government-Leader 57. Greeting 58. Ground_Vehicles 59. Hand 60. Handshaking 61. Harbors 62. Helicopter_Hovering 63. Highway 64. Horse 65. Hospital 66. House_Of_Worship 67. Indoor 68. Indoor_Sports_Venue 69. Industrial_Setting 70. Infants 71. Instrumental_Musician 72. Kitchen 73. Laboratory 74. Landscape 75. Male_Person 76. Maps 77. Meeting 78. Military 79. Military_Base 80. Motorcycle 81. Mountain 82. Natural-Disaster 83. News_Studio 84. Nighttime 85. Office 86. Old_People 87. Outdoor 88. Overlaid_Text 89. People_Marching 90. Person 91. Plant 92. Police_Private_Security_Personnel 93. Politicians 94. Politics 95. Press_Conference 96. Prisoner 97. Reporters 98. Road 99. Roadway_Junction 100. Running 101. Scene_Text 102. Science_Technology 103. Scientists 104. Shopping_Mall 105. Singing 106. Single_Person 107. Sitting_Down 108. Sky 109. Snow 110. Soccer_Player 111. Sports 112. Stadium 113. Streets 114. Suburban 115. Swimming 116. Teenagers 117. Telephones 118. Tennis 119. Tent 120. Throwing 121. Trees 122. Truck 123. Two_People 124. US_Flags 125. Vegetation 126. Vehicle 127. Walking 128. Walking_Running 129. Waterscape_Waterfront 130. Weather 131. 3_Or_More_People 132. Adult_Female_Human 133. Adult_Male_Human 134. Advocate 135. Airplane_Landing 136. Airplane_Takeoff 137. Airport_Or_Airfield 138. Amateur_Video 139.

Anger 140. Animal_Pens_And_Cages 141. Animation_Cartoon 142. Apartment_Complex 143. Apartments 144. Armed_Person 145. Armored_Vehicles 146. Arthropod 147. Attached_Body_Parts 148. Baby 149. Background_Static 150. Bar_Pub 151. Baseball 152. Black_Frame 153. Blank_Frame 154. Body_Parts 155. Bomber_Bombing 156. Boredom 157. Car_Crash 158. Carnivore 159. Cattle 160. Caucasians 161. Cell_Phones 162. Cetacean 163. Child 164. Church 165. Cigar_Boats 166. City 167. Civilian_Person 168. Clearing 169. Clouds 170. Colin_Powell 171. Commentator_Or_Studio_Expert 172. Commercial_Advertisement 173. Conference_Buildings 174. Construction_Site 175. Construction_Worker 176. Crane_Vehicle 177. Crustacean 178. Cul-de-Sac 179. Dining_Room 180. Disgust 181. Dolphin 182. Domesticated_Animal 183. Door_Opening 184. Dresses 185. Dresses_Of_Women 186. Earthquake 187. Election_Campaign 188. Election_Campaign_Address 189. Election_Campaign_Convention 190. Election_Campaign_Debate 191. Election_Campaign_Greeting 192. Eukaryotic_Organism 193. Event 194. Exiting_A_Vehicle 195. Exiting_Car 196. Factory 197. Factory_Worker 198. Fear 199. Female_Anchor 200. Female_Human_Face 201. Female_News_Subject 202. Female_Reporter 203. Fields 204. Fighter_Combat 205. Fight-Physical 206. Fire_Truck 207. First_Lady 208. Flags 209. Flood 210. Food 211. Football 212. Forest 213. Free_Standing_Structures 214. Freighter 215. Furniture 216. George_Bush 217. Glasses 218. Golf_Player 219. Graphic 220. Ground_Combat 221. Guard 222. Gun 223. Gun_Shot 224. Gym 225. Head_And_Shoulder 226. Helicopters 227. Herbivore 228. High_Security_Facility 229. Hill 230. Hispanic_Person 231. Hockey 232. Human_Young_Adult 233. Indian_Person 234. Insect 235. Insurgents 236. Invertebrate 237. Islands 238. Japanese 239. John_Kerry 240. Joy 241. Junk_Frame 242. Korean 243. Lakes 244. Legs 245. Machine_Guns 246. Male_Anchor 247. Male_Human_Face 248. Male_News_Subject 249. Male_Reporter 250. Male-Human-Face-Closeup 251. Mammal 252. Man_Made_Thing 253. Man_Wearing_A_Suit 254. Military_Aircraft 255. Military_Airplane 256. Military_Buildings 257. Military_Personnel 258. Military_Vehicle 259. Minivan 260. Moonlight 261. Mosques 262. Muslims 263. Network_Logo 264. News 265. Oceans 266. Office_Building 267. Officers 268. Oil_Drilling_Site 269. Pan_Zoom_Static 270. Pavilions 271. Person_Drops_An_Object 272. Pickup_Truck 273. Police 274. Police_Car 275. Police_Truck 276. Primate 277. Processing_Plant 278. Professional_Video 279. Quadruped 280. Raft 281. Religious_Building 282. Religious_Figures 283. Rescue_Helicopter 284. Rescue_Vehicle 285. Researcher 286. Residential_Buildings 287. Rifles 288. River 289. Road_Block 290. Road_Overpass 291. Rocky_Ground 292. Room 293. Rowboat 294. Rpg 295. Ruminant 296. Sadness 297. Sailing_Ship 298. School 299. Sea_Mammal 300. Security_Checkpoint 301. Single_Person_Female 302. Single_Person_Male 303. Skating 304. Ski 305. Skier 306. Skyscraper 307. Soccer 308. Sofa 309. Soldiers 310. Speaker_At_Podium 311. Speaking 312. Speaking_To_Camera 313. Sports_Car 314. Standing 315. Still_Image 316. Street_Battle 317. Studio_With_Anchorperson 318. Suits 319. Sun 320. Sunglasses 321. Sunny 322. Surprise 323. Swimming_Pools 324. Synagogue 325. Synthetic_Images 326. Table 327. Talking 328. Taxi_Cab 329. Text 330. Text_Labeling_People 331. Text_On_Artificial_Background 332. Throw_Ball 333. Tower 334. Traffic 335. Underwater 336. Urban_Park 337. Urban_Scenes 338. Valleys 339. Van 340. Vertebrate 341. Violent_Action 342. Weapons 343. Whale 344. Wild_Animal 345. Windows 346. Yasser_Arafat

A.2 ImageCLEF Concepts

1. Partylife 2. Family_Friends 3. Beach_Holidays 4. Building_Sights 5. Snow 6. Citylife 7. Landscape_Nature 8. Sports 9. Desert 10. Spring 11. Summer 12. Autumn 13. Winter 14. Indoor 15. Outdoor 16. Plants 17. Flowers 18. Trees 19. Sky 20. Clouds 21. Water 22. Lake 23. River 24. Sea 25. Mountains 26. Day 27. Night 28. Sunny 29. Sunset_Sunrise 30. Still_Life 31. Macro 32. Portrait 33. Overexposed 34. Underexposed 35. Neutral_Illumination 36. Motion_Blur 37. Out_of_focus 38. Partly_Blurred 39. No_Blur 40. Single_Person 41. Small_Group 42. Big_Group 43. No_Persons 44. Animals 45. Food 46. Vehicle 47. Aesthetic_Impression 48. Overall_Quality 49. Fancy 50. Architecture 51. Street 52. Church 53. Bridge 54. Park_Garden 55. Rain 56. Toy 57. MusicalInstrument 58. Shadow 59. bodypart 60. Travel 61. Work 62. Birthday 63. Visual_Arts 64. Graffiti 65. Painting 66. artificial 67. natural 68. technical 69. abstract 70. boring 71. cute 72. dog 73. cat 74. bird 75. horse 76. fish 77. insect 78. car 79. bicycle 80. ship 81. train 82. airplane 83. skateboard 84. female 85. male 86. Baby 87. Child 88. Teenager 89. Adult 90. old_person 91. happy 92. funny 93. euphoric 94. active 95. scary 96. unpleasant 97. melancholic 98. inactive 99. calm

Appendix B

Appendix B

B.1 Video Descriptors (Generated by IRIM partners)

Nine IRIM participants (CEA-LIST, ETIS/LIP6, EURECOM, GIPSA, INRIA, LABRI, LIF, LIG, and LSIS) provided a total of 48 descriptors, including variants of same descriptors. Here we present these descriptors:

CEALIST/tlep: texture local edge pattern (Cheng & Chen [2003]) + color histogram $\rightsquigarrow$ 576 dimensions.

ETIS/global_<feature>[<type>]x<size>: (concatenated) histogram features (Gorisse *et al.* [2011]), where:

<**feature**> is chosen among lab and qw:

lab: CIE L*a*b* colors

qw: quaternionic wavelets (3 scales, 3 orientations)

<**type**> can be

nothing: histogram computed on the whole image

m1x3: histogram for 3 vertical parts

m2x2: histogram on 4 image parts

<**size**> is the dictionary size, sometimes different from the final feature vector dimension.

For instance, with <type>=m1x3 and <size>=32, the final feature vector has $3 \times 32 = 96$ dimensions.

EUR/sm462 : the Saliency Moments (SM) feature (Redi & Merialdo [2011]), is a holistic descriptor that embeds locally-parsed information, namely the shape of the salient region, in a holistic representation of the scene, structurally similar to Oliva & Torralba [2001]. First, the saliency information is extracted at different resolutions using a spectral, lightweight algorithm. The signals obtained are then sampled directly in the frequency domain, using a set of Gabor wavelets. Each of these samples, called ”Saliency Components”, is then interpreted as a probability distribution: the components are divided into sub-windows and the first three moments are extracted, namely mean, standard deviation and skewness. The resulting signature vector is a 462-dimensional descriptor that we use as input for traditional support vector machines and combine then with the contributions of the other visual features.

GIPSA/AudioSpectro[N]-b28: Spectral profile in 28 bands on a Mel scale, N: normalized ⇝ 28 dimensions.

INRIA/dense_sift_<k>: Bag of SIFT computed by INRIA with k-bin histograms ⇝ k dimensions with k = 128, 256, 512, 1024, 2048 and 4096.

LEAR/sift_bow4096: Bag Of SIFT Words vectors with dictionary size equal to 4096.

LABRI/faceTracks: OpenCV+median temporal filtering, assembled in tracks, projected on keyframe with temporal and spatial weighting and quantized on image divided in 16 × 16 blocks ⇝ 256 dimensions.

LIF/percepts_<x>_<y>_1_15: 15 mid-level concepts detection scores computed on x × y grid blocks in each key frames with (x,y) = (20,13), (16,6), (5,3), (2,2) and (1,1), ⇝ 15 × x × y dimensions.

KIT/faces KIT contributed by proposing descriptors/predictions at the face level.

LIG/h3d64: normalized RGB Histogram 4 × 4 × 4 ⇝ 64 dimensions.

LIG/gab40: normalized Gabor transform, 8 orientations × 5 scales, ⇝ 40 dimensions.

LIG/hg104: early fusion (concatenation) of h3d64 and gab40 ⇝ 104 dimensions.

LIG/opp_sift_<method>[_unc]_1000: bag of word, opponent sift, generated using van de Sande *et al.* [2008] software. ⇝ 1000 dimensions (384 dimensions per detected point before clustering; clustering on 535117 points coming from 1000 randomly chosen images). <**method**> method is related to the way by which SIFT points are selected: **har** corresponds to a filtering via a Harris-Laplace detector and **dense** corresponds to a dense sampling; the versions with **_unc** correspond to the same with fuzziness introduced in the histogram computation.

LIG/stip_<method>_<k>: bag of word, STIP local descriptor, generated using Laptev [2005] software, <**method**> may be either histograms of oriented (spatial) gradient (**hog**) or histograms of optical flow (**hof**), ⇝ k dimensions with k = 256 or 1000.

LIG_concepts: detection scores on the 346 TRECVid 2011 SIN concepts using the best available fusion with the other descriptors, ⇝ 346 dimensions.

LISTIC/SURF_retinaMasking_<k>_cross: SURF based bag of words (BOW) with k = 1024 or 4096 dimensions using a real-time retina model (Benoit *et al.* [2010]). We consider 40 frames around each sub-shot keyframe. An automatic salient blobs segmentation is applied on each frame and a dense grid is considered only within these regions. SURF descriptors are captured within each frame blobs and are cumulated along the 40 frames.

This allows the BOW of the subshot keyframe to be defined globally. Descriptors are extracted from the retinal foveal vision model (Parvocellular pathway). It allows light and noise robustness and enhanced SURF description. The retinal motion channel (Magnocellular pathway) is used to perform the automatic blobs segmentation. This channel allows transient blobs to be detected during the 40 frames. Such transient blobs are related to salient detailed areas during the retina model transient state (during the 20 first frames). Its also corresponds to moving areas at the retina's stable state (during the last 20 frames). Such segmentation allows spatio-temporal low level saliency areas to be detected. For BOW training, vocabulary learning is performed with Kmeans on 1008 subshots taken from 2011a and 2011b keyframes lists using 6 622 198 points.

LSIS/mlhmslbp_spyr_<k>: Three kinds of parameters based on a Multi-Level Histogram of Multi-Scale features including spatial pyramid technique (MLHMS) (Paris & Glotin [2010]). In each parameters extraction method, the pictures were considered as grayscale pictures. The two first kinds of parameters are based on local binary pattern (LBP). A two levels pyramid was used with the level being the entire picture and the second level being a half in the horizontal direction and a forth in the vertical direction respectively a third and a sixth for the second kind of parameters). Moreover, an overlapping of half of the level-direction size is used. 4 levels of scaling were also computed for the LBP parameters, from 1 to 4 pixels blocks. The resulting parameter vectors are then L2-clamp normed. For the third kind of parameters, we used second order Local Derivative Pattern (LDP). We used the same kind of level, scaling and spatial pyramid than for the two preceding parameters. The dimensions of the resulting vectors are respectively 10240 and 26624 for the MLHMS-LBP parameters, and 106496 for the MLHMS-LDP parameters. For practical reasons, we were only able to use the MLHMS-LBP descriptor with 10240 dimensions.

Appendix C

Appendix C

C.1 List of Publications

International journal:

1. Bahjat Safadi and Georges Quénot. Active learning with multiple classifiers for multimedia indexing. *Multimedia Tools and Applications*, 1-15, 2010.

International conferences:

1. Bahjat Safadi, Stéphane Ayache and Georges Quénot. Active Cleaning for Video Corpus Annotation. *International MultiMedia Modeling Conference*, pages:518-528, Klagenfurt, Austria, Jan 2012.
2. Bahjat Safadi and Georges Quénot. Re-ranking by Local Re-scoring for Video Indexing and Retrieval. *CIKM 2011: 20th ACM Conference on Information and Knowledge Management*, pages:2081-2084, Glasgow, Scotland, Oct 2011.
3. Bahjat Safadi and Georges Quénot. Re-ranking for Multimedia Indexing and Retrieval. *ECIR 2011: 33rd European Conference on Information Retrieval*, pages:708-711, Dublin, Ireland, Apr 2011.
4. Bahjat Safadi, Yubing Tong and Georges Quénot. Incremental Multiple Classifier Active Learning for Concept Indexing in Images and Videos. *International MultiMedia Modeling Conference*, pages:240-250, Taipei, Taiwan, Jan 2011.
5. Bahjat Safadi and Georges Quénot. Evaluations of multi-learners approaches for concepts indexing in video documents. *RIAO*, pages:88-91, Paris, France, Apr 2010.

International workshops:

1. Bahjat Safadi and Georges Quénot. Active Learning with Multiple Classifiers for Multimedia Indexing. *The 8th IEEE Int. Workshop on Content-Based Multimedia Indexing (CBMI)*, Grenoble, France, Jun 2010.

2. Bahjat Safadi, Yubing Tong and Georges Quénot. Incremental Multi-Classifier Learning Algorithm on Grid'5000 for Large Scale Image Annotation. *ACM Workshop on Very-Large-Scale Multimedia Corpus, Mining and Retrieval*, pages:1-6, Firenze, Italy, Oct 2010.

International non-reviewed workshops:

1. Bahjat Safadi, Nadia Derbas, Abdelkader Hamadi, Franck Thollard, Georges Quénot, Hervé Jégou, Tobias Gehrig, Hazim Kemal Ekenel and Rainer Stifelhagen. Quaero at TRECVid 2011: Semantic Indexing and Multimedi Event Detection. *TREC Video Retrieval Evaluation workshop*, Gaithersburg, MD USA, Dec 2011.

2. Bertrand Delezoide, Frédéric Precioso, Philippe Gosselin, Miriam Redi, Bernard Merialdo, Lionel Granjon, Denis Pellerin, Michèle Rombaut, Hervé Jégou, Rémi Vieux, Aurlie Bugeau, Boris Mansencal, Jenny Benois-Pineau, Hugo Boujut, Stéphane Ayache, Bahjat Safadi Franck Thollard, Georges Quénot, Hervé Bredin, Matthieu Cord, Alexandre Benot, Patrick Lambert, Tiberius Strat, Joseph Razik, Sbastion Paris and Hervé Glotin. IRIM at TRECVid 2011: Semantic Indexing and Instance Search. *TREC Video Retrieval Evaluation workshop*, Gaithersburg, MD USA, Dec 2011.

3. Rami Albatal, Bahjat Safadi, Georges Quénot and Philippe Mulhem. LIG-MRIM at Image Photo Annotation Task in ImageCLEF 2011. *CLEF (Notebook Papers/Labs/Workshop)*, Sep 2011.

4. Bahjat Safadi, Georges Quénot. LIG at MediaEval 2011 affect task: use of a generic method. *MediaEval 2011 Workshop*, Pisa, Italy, Sep 2011.

5. David Gorisse, Frédéric Precioso, Philippe Gosselin, Lionel Granjon, Denis Pellerin, Michèle Rombaut, Hervé Bredin, Lionel Koenig, Hélène Lachambre, Elie El-Khoury, Rémi Vieux, Boris Mansencal, Yifan Zhou, Jenny Benois-Pineau, Hervé Jégou, Stéphane Ayache, Bahjat Safadi, Yubing Tong, Franck Thollard, Georges Quénot, Alexandre Benoit and Patrick Lambert. IRIM at TRECVid 2010: High Level Feature Extraction and Instance Search. *TREC Video Retrieval Evaluation workshop*, Gaithersburg, MD USA, Nov 2010.

6. Stéphane Ayache, Bahjat Safadi, Andy Tseng, Franck Thollard, Georges Quénot, Hazim Ekenel, Tobias Gehrig and Rainer Stifelhagen. Quaero at TRECVid 2010: Semantic Indexing. *TREC Video Retrieval Evaluation workshop*, Gaithersburg, MD USA, Nov 2010.

7. Bahjat Safadi and Georges Quénot. LIG at TRECVid 2009: Hierarchical Fusion for High Level Feature Extraction. *TREC Video Retrieval Evaluation workshop*, Gaithersburg, MD USA, Nov 2009.

8. Georges Quénot, Bertrand Delezoide, Hervé le Borgne, Pierre-Alain Moëllic, David Gorisse, Frédéric Precioso, Feng Wang, Bernard Merialdo, Philippe Gosselin, Lionel

Granjon, Denis Pellerin, Michèle Rombaut, Hervé Bredin, Lionel Koenig, Hélène Lachambre, Elie El Khoury, Boris Mansencal, Jenny Benois-Pineau, Hervé Jégou, Stéphane Ayache, Bahjat Safadi, Jonathan Fabrizio, Matthieu Cord, Hervé Glotin, Zhongqiu Zhao, Emilie Dumont and Bertrand Augereau. IRIM at TRECVid 2009: High Level Feature Extraction. *TREC Video Retrieval Evaluation workshop*, Gaithersburg, MD USA, Nov 2009.

National conferences:

1. Bahjat Safadi and Georges Quénot. Apprentissage Actif avec une Méthode de Réordonnancement pourl 'Indexation et la Recherche de Vidéos. *Actes de la conférence CORIA*, pages::231-245, Avignon, France, Mar 2011.

2. Bahjat Safadi and Georges Quénot. Évaluation des approches multi-apprenants pour l'indexation des concepts dans les documents vidéo. *Actes de la conférence CORIA*, Sousse, Tunisie, Mar 2010.

Appendix D

Appendix D: Résumé en Français

D.1 Introduction

Dans la dernière décennie, les bases de données multimédia numériques ont augmenté à une vitesse phénoménale dans de nombreux domaines, conduisant une explosion du nombre et de la taille des archives et des collections dimages et de vidéos. En particulier, avec l'utilisation généralisée des caméras numériques et l'augmentation du stockage des ordinateurs personnels, qui peuvent atteindre des centaines de giga-octets, les particuliers peuvent aujourd'hui produire facilement des milliers d'images et vidéos personnelles. En outre, la révolution dans la diffusion de médias numériques, tels que les Retransmissions TV et les magazines numériques, et la possibilité de partager des documents multimédia via Internet (par exemple, facebook, youtube, flickr, etc.) avec des personnes partout dans le monde, ont changé la façon dont nous acquérons, stockons et partageons ces documents. En conséquence de la grande croissance de ces bases de données multimédia numériques, les problèmes liés à l'indexation et à la recherche multimédia ont eux aussi augmenté. Sassurer que les collections multimédias puissent être indexées, recherchées et accédées de manière efficace est devenu un problème de plus en plus important.

L'indexation du contenu est l'un des principaux enjeux pour la recherche de documents dans la majorité des collections numériques multimédias. Cette indexation peut être faite au niveau du signal (par exemple par la couleur, la texture, mouvement, etc.) ou au niveau sémantique (concepts). Lindexation au niveau du signal (cest-à-dire par caractéristiques dites de bas niveau) peut être utile dans certains domaines, tels que le domaine médical. Par exemple, un médecin peut être intéressé par la recherche des zones sombres ou claires dans une image ou une vidéo. Toutefois, dans les vidéos publiques, comme les nouvelles, le cinéma, la télévision ou des vidéos personnelles, les utilisateurs recherchent des séquences particulières qui représentent leurs besoins sémantiquement. Par exemple, " le but fantastique de Marco van Basten contre l'URSS en finale du Championnat d'Europe en 1988 ≫. Ainsi, l'indexation sémantique, à ce jour, est la fonctionnalité la plus nécessaire pour les utilisateurs, mais elle est aussi la plus difficile car il y a un besoin d'outils puissants pour extraire correctement le contenu des documents multimédia. En raison du problème dit du fossé sémantique entre le contenu des données brutes et les éléments qui font sens pour l'homme, lindexation des concepts dans les documents multimédias est une tâche très difficile. Une solution pour combler le fossé sémantique pour une requête donnée est dannoter manuellement la base de données entière. Bien quune telle approche soit envisageable pour de petites collections, elle est extrêmement coûteuse et fastidieuse et donc inapplicable pour des collections de grande tailles. Il est nécessaire de concevoir des méthodes et des outils automatiques pour effectuer cette tâche. Ces méthodes sont basés sur des techniques d'apprentissage automatique, au travers dune modélisation des connaissances sémantiques apprises à partir de caractéristiques de bas niveau. Les systèmes doivent être entraînés à partir déchantillons positifs et négatifs (l'ensemble d'apprentissage) pour générer des modèles qui peuvent ensuite être utilisés pour prédire des étiquettes pour les documents non étiquetés (Naphade & Smith [2004]; Snoek *et al.* [2006]; Tahir *et al.* [2009a]).

Lorsque le volume de données qui peuvent être annotées manuellement est limité en raison du coût de l'intervention humaine, il reste la possibilité de sélectionner un sous-ensemble d'échantillons à faire annoter manuellement afin que leurs annotations soient aussi utiles que possible (Angluin [1988]). Décider quels sont les échantillons dont les annotations seront les plus utiles n'est pas trivial et lapprentissage actif est une approche dans laquelle un système existant est utilisé pour faire cette prédiction. Cette approche est un cas particulier dapprentissage incrémental, dans lequel un système est entraîné (ou ajusté) plusieurs fois en utilisant un nombre croissant d'échantillons étiquetés. L'objectif est de sélectionner, de manière itérative, des échantillons à faire annoter manuellement aussi peu nombreux que possible pour obtenir simultanément la meilleure performance de classification possible.

Dans cette thèse, nous avons développé une approche basée sur l'apprentissage actif qui permet de produire des ensembles de données annotées dans les bases de données multimédias qui sont utilisés dans les processus de classification et d'indexation. En outre, nous avons proposé des approches qui améliorent la performance des systèmes d'indexation multimédia. Par ailleurs, nous avons construit un système d'indexation qui utilise les approches mentionnées ci-dessus pour l'indexation multimédia dans une application réelle qui a été utilisée pour la construction de l'ensemble du développement de TRECVid 2010 à 2012 (la campagne internationale majeure pour lévaluation des systèmes dindexation et de recherche par le contenu de collections de documents vidéos). Les termes " annotation " et " étiquette " dans cette thèse font référence à des concepts représentés par des mots-clés textuels pour la description du contenu des documents multimédia.

D.1.1 Indexation multimédia basé sur le système d'apprentissage actif

Les systèmes de recherche d'information (RI) sont conçus pour répondre aux besoins des utilisateurs dans la recherche de documents. Par conséquent, l'objectif de ces systèmes est de trouver les documents les plus pertinents qui correspondent à la requête générée par l'utilisateur. Pour les documents multimédias, en général, ces systèmes sont basés sur les annotations manuelles ou des étiquettes des documents. Plus les annotations disponibles sont correctes et complètes, plus le système de RI qui les exploite sera performant. Dans les bases de données multimédia de grande taille, la qualité et la quantité des documents annotés ont tous les deux une grande importance. Puisque les deux affectent les algorithmes d'apprentissage, nous avons besoin densembles d'apprentissage avec des annotations appropriées en qualité et en quantité. Bien quannoter plus de documents conduise à de meilleures performances, annoter lensemble des données disponibles coûte beaucoup de temps et d'argent. Les méthodes dapprentissage actif peuvent être considérées comme des solutions prometteuses pour créer des ensembles dentraînement. La figure D.1 montre un exemple dun système d'indexation multimédia basée sur un apprentissage actif. Un ensemble d'apprentissage qui contient très peu déchantillons étiquetés L et un grand nombre d'échantillons non étiquetés U est considéré au départ. Les échantillons sont représentés chacun par un vecteur de taille fixe de caractéristiques de bas niveau. Le système est itératif et basé sur l'apprentissage des classifieurs à partir des échantillons disponibles étiquetés (L) pour prédire l'utilité des échantillons non étiquetés (U). En utilisant les scores prédits, le système sélectionne les échantillons non encore étiquetés dont lannotation sera la plus informative. Ceux-ci sont transmis à des annotateurs (oracles) pour quils les étiquettent. Les ensembles L et U sont ensuite mis à jour. Ce processus est répété jusqu'à ce quune condition d'arrêt soit atteinte (par exemple, une certaine fraction des échantillons a été étiquetée). La performance de lapprenant actif peut être mesurée en fonction de la taille de la fraction annotée si lon dispose dun ensemble annoté indépendant (ensemble de test).

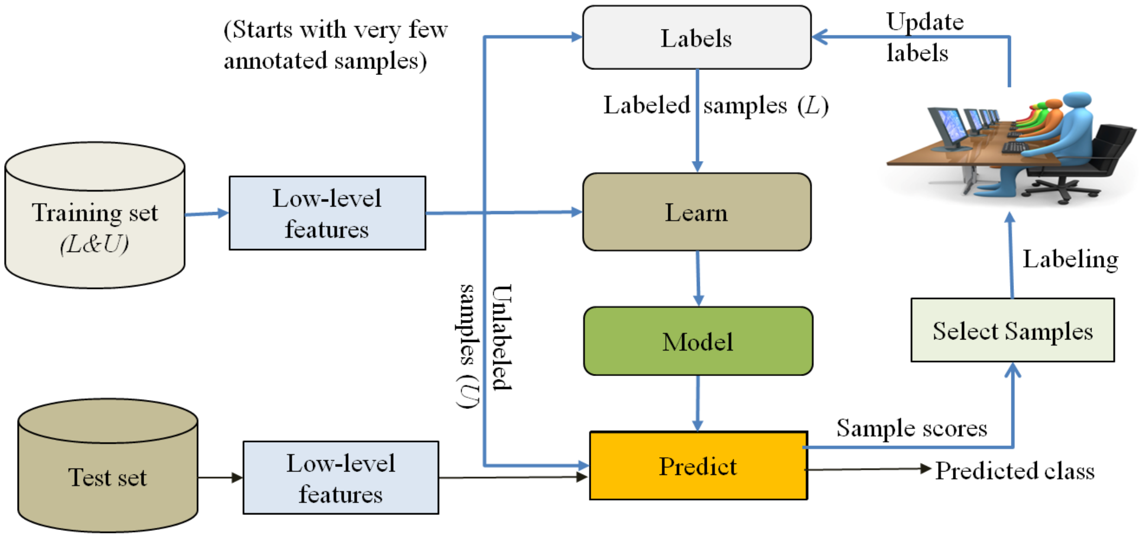


Figure D.1: Un système d'apprentissage actif.

La performance du système dépend des techniques de description des données, de lalgorithme de classification et de la stratégie de sélection des échantillons à annoter. Dans cette thèse, nous nous intéressons à l'indexation des documents multimédias, notamment les vidéos. L'objectif du système est daider à construire des ensembles d'apprentissage pour une collection vidéo en assignant des étiquettes sémantiques à chaque échantillon de vidéo. Ceci nécessite un système pour représenter efficacement les vidéos dans un corpus donné, de sorte que ces représentations puissent conduire à la récupération de données par des utilisateurs. Dans la section suivante, nous donnons un aperçu de l'indexation sémantique.

D.1.2 L'indexation sémantique

Les systèmes qui utilisent la description sémantique des documents sont considérés comme la prochaine génération d'indexation des documents car ils permettent l'accès sémantique aux documents multimédias. Plutôt que d'interagir avec le système en utilisant des descriptions de bas niveau, les utilisateurs peuvent interagir à un niveau sémantique élevé en utilisant directement et simplement des mots-clés ou des concepts sémantiques (Lim [2001]; Snoek *et al.* [2005a]).

En raison des difficultés d'extraction des symboles (haut niveau sémantique) à partir d'un signal brut (bas niveau), ces systèmes deviennent actuellement un sujet très actif dans la communauté de la recherche d'information. L'avantage de ces systèmes est leur capacité à produire une représentation abstraite à partir des caractéristiques de bas niveau. Ils doivent être en mesure d'indexer les concepts qui présentent des similitudes visuelles bien qu'un même concept puisse apparaître sous différentes couleurs et formes. Ce problème classique dans l'indexation multimédia est appelé le " fossé sémantique " (Smeulders *et al.* [2000]). L'association des concepts avec leurs caractéristiques visuelles est une tâche non triviale.

Dans les systèmes d'indexation sémantique multimédia (MSI), passer du bas niveau (signal) au haut niveau (concept) est généralement effectué par apprentissage supervisé. Un concept donné est modélisé à partir d'un ensemble d'échantillons étiquetés comme positifs ou négatifs par rapport à celui-ci. Un classificateur est alors entraîné pour reconnaître le même concept dans les documents non étiquetés. Bien que ces systèmes répondent mieux aux besoins des utilisateurs, ils sont encore difficiles à appliquer. En outre, il est difficile de définir ainsi tous les concepts connus et il est encore impossible de construire des systèmes intelligents de MSI, qui répondent parfaitement aux attentes des utilisateurs. En outre, les algorithmes d'apprentissage supervisé dépendent beaucoup de la représentation des données et de la qualité des échantillons d'apprentissage. Ceci peut être vu en apprenant des concepts dans de grandes bases de données où les systèmes sont souvent confrontés au problème du déséquilibre entre les classes minoritaires et majoritaires, la plupart des concepts cibles étant rares ou très rares. Dans la section suivante, nous abordons les deux problèmes mentionnés ci-dessus (c'est à dire celui du fossé sémantique et celui du déséquilibre entre classes) dans l'indexation multimédia.

D.1.3 Description du problème

Comme notre travail concerne l'indexation multimédia, nous identifions deux problèmes principaux que rencontrent en général les systèmes d'indexation multimédia. Le premier est le problème majeur du fossé sémantique entre les caractéristiques de bas niveau visuel (couleur, forme, texture, etc.) et les concepts sémantiques manipulés par l'utilisateur (Chang *et al.* [2009]; Idrissi *et al.* [2009]; Smeulders *et al.* [2000]). Le deuxième problème est celui du déséquilibre entre classes qui apparaît principalement dans les bases de données à grande échelle, telles que TRECVid (Smeaton *et al.* [2006]), où la plupart des concepts cibles à apprendre sont rares. C'est encore un sérieux problème pour les méthodes classiques d'apprentissage supervisé car la plupart des méthodes performantes sont conçues pour fonctionner avec un déséquilibre modéré entre les classes et se comportent mal en cas de déséquilibre important. Par ailleurs, un autre problème est l'inadéquation entre la métrique d'évaluation appropriée pour la recherche d'information et le critère interne du classificateur. L'unité de mesure commune utilisée dans la recherche d'information est la précision moyenne (MAP), qui donne plus d'importance aux échantillons trouvés dans le haut d'une liste ordonnée tandis que le critère interne de l'algorithme d'apprentissage supervisé s'attend à trouver des classes grossièrement équilibré et n'est pas influencé par le classement.

D.1.3.1 Fossé sémantique

Le terme " fossé sémantique " a été introduit en 2000 dans le cadre de l'indexation sémantique des images. Smeulders *et al.* [2000] décrit le fossé sémantique comme suit: " *Le fossé sémantique est le manque de concordance entre les informations que les machines peuvent extraire depuis les documents numériques, et les interprétations que les humains en font* ".

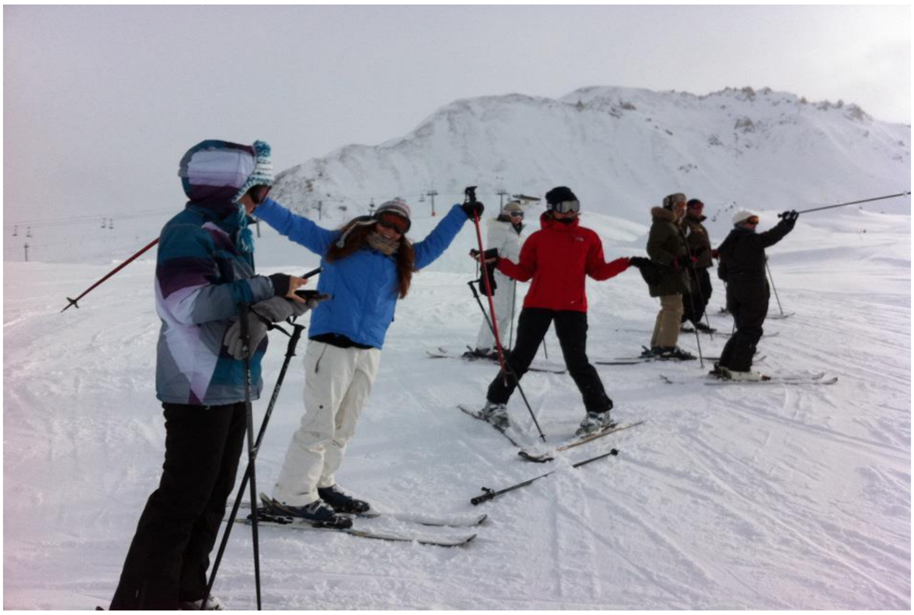


Figure D.2: Ski à Belle-Plagne, dans les Alpes françaises, un exemple du fossé sémantique.

Combler le fossé sémantique dans la recherche d'images et de vidéos est un problème encore très difficile à résoudre (Chang *et al.* [2009]; Idrissi *et al.* [2009]). Nous, en tant qu'êtres humains, sommes capables de comprendre les documents images et vidéos selon les deux points de vue. Les caractéristiques de bas niveau (par exemple la couleur, la texture, la forme, etc.) et de haut niveau de la sémantique (concepts, événements, etc.). Cependant, les machines n'utilisent encore que des caractéristiques de bas niveau pour interpréter ces documents. Les utilisateurs préfèrent chercher des documents en utilisant des requêtes de haut niveau (Liu *et al.* [2007b]; Smeulders *et al.* [2000]), tandis que les systèmes de recherche des images et des vidéos en fonction de leur contenu (CBIR et CBVR) utilisent des caractéristiques bas niveaux pour indexer les images et les vidéos. Une tentative pour combler le fossé entre la sémantique de haut niveau et des caractéristiques de bas niveau est nécessaire. Par exemple, peut-on construire des machines qui peuvent comprendre sémantiquement le contenu de l'image montre la figure D.2? Comme on le voit, c'est un défi que la machine comprenne que la photo a été prise sur une montagne, que le ciel est nuageux, qu'il y a de la neige (photo prise en hiver), que l'image se concentre sur trois dames parmi sept personnes, et que l'activité générale est le ski.

Dans le but de combler le fossé sémantique, les techniques d'annotation multimédia automatique ont suscité beaucoup de intérêt ces dernières années. Le but des techniques d'annotation automatique est d'attacher des étiquettes textuelles à des documents non étiquetés, comme descriptions du contenu des documents. Cet étiquetage doit être fait en utilisant la sémantique de haut niveau.

D.1.3.2 Le problème de classe déséquilibre dans des bases de données à Grande Échelle

Un ensemble de données est appelé " déséquilibré " si une des classes à reconnaître est représentée par un nombre significativement moins élevé d'échantillons que les autres. Ce problème se produit dans de nombreuses bases de données du monde réel, y compris ceux du multimédia. Le problème de classe déséquilibre est devenu un sujet intéressant au sein de la communauté travaillant sur l'apprentissage automatique dans de nombreux domaines particuliers, notamment dans l'indexation multimédia.

Dans le cas de la classification binaire (échantillons positifs ou négatifs), on suppose que la classe positive est la classe minoritaire, et la classe négative est la classe majoritaire. Dans de nombreux cas, la classe minoritaire est très rare, tel que moins de 1% dans les bases de données TRECVID (Smeaton *et al.* [2006]), comme le montre la figure D.3. Dans ce contexte, les algorithmes standards d'apprentissage supervisé ont une précision très faible pour la prédiction de la classe rare. Ainsi, lors de leur application en tant que méthodes de classement sur les bases de données déséquilibrées, ils sont susceptibles de tout prédire comme négatif (la classe majoritaire). Ce problème a souvent été considéré comme celui de l'apprentissage à partir d'un ensemble de données très déséquilibrée (Japkowicz & Stephen [2002]; Weiss & Provost [2001]).

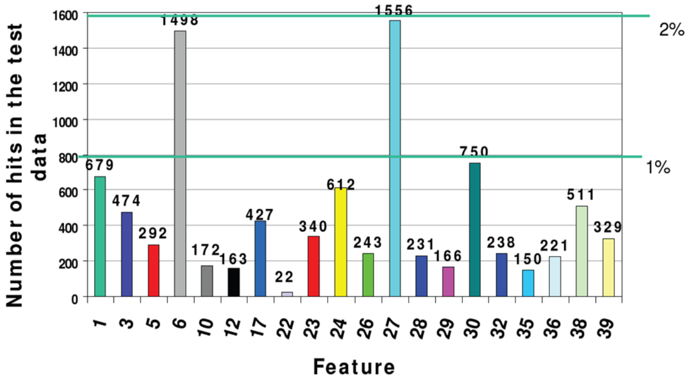


Figure D.3: Fréquence de concepts, de Smeaton *et al.* [2006].

Une possibilité pour surmonter le problème du déséquilibre de classe, est de ré-échantillonner l'ensemble de données d'apprentissage original, soit par sur-échantillonnage de la classe minoritaire soit par sous-échantillonnage de la classe majoritaire. Ceci est fait de telle manière que les classes seront présentés d'une manière plus équilibrée (Bishop [2007]; Chawla *et al.* [2002]; Weiss & Provost [2001]). Le sur-échantillonnage augmente le temps d'apprentissage et les besoins en mémoire en raison de l'augmentation du nombre d'exemples d'apprentissage. Il augmente aussi les coûts de calcul pour le prétraitement des données. Le sous-échantillonnage peut, lui, induire une perte d'information du fait qu'il peut ignorer des données qui pourraient être importantes pour le processus d'apprentissage.

La motivation sous-jacente dans les méthodes de ré-échantillonnage est d'alimenter le classifieur avec un ensemble d'apprentissage ayant des classes plus équilibrées. Nous cherchons à améliorer ces techniques qui permettent de surmonter le problème du déséquilibre entre classes dans l'indexation de documents multimédia.

D.1.4 Buts et objectifs

Notre travail se concentre principalement sur l'indexation sémantique des documents multimédias. Plus précisément, nous nous intéressons à l'indexation sémantique des concepts dans les documents multimédias (images et vidéos) en utilisant les méthodes d'apprentissage actif que nous utilisons pour construire des corpus annoté. L'objectif principal de cette thèse est de développer et d'évaluer des approches prometteuses d'apprentissage actif pour l'indexation automatique des images et des vidéos dans les ensembles de données multimédias à grande échelle. L'indexation se fait soit par le développement de nouvelles techniques soit par l'amélioration des techniques différentes dans la communauté de la recherche d'information ou de la recherche multimédia spécialement basée sur le contenu. Nous visons à ce que le système soit en mesure d'indexer un grand nombre de concepts différents. Ainsi, l'approche doit être générique et utilisable quel que soit le concept cible. En outre, nous espérons pouvoir appliquer notre système d'indexation pour l'annotation ensembles d'apprentissage de grande envergure, tels que ceux développés dans le cadre des campagnes d'évaluation " TRECVid ".

Un point important dans notre travail est l'évaluation de nos contributions. Nous souhaitons valider les modèles proposés dans des conditions optimales. Ainsi, un élément de contexte majeur pour cette thèse est la participation aux campagnes d'évaluation à grande échelle (par exemple TRECVid et ImageCLEF). Ceci implique que les approches proposées seront capables à manipuler de vastes corpus de données.

Les efforts pour atteindre les objectifs se composent de trois parties: i) le développement des descriptions appropriées des documents multimédias (les descripteurs visuels et audio), ii) le développement de techniques avancées d'apprentissage automatique pour la classification d'images et de vidéos; iii) le développement d'un système basé sur une approche d'apprentissage actif pour mener à la création de l'ensemble de développement, en demandant itérativement à des " oracles " d'annoter des documents non étiquetés de l'ensemble de données. Dans cette thèse, nous allons explorer ces trois domaines, bien que l'objectif de la recherche soit focalisé sur les deuxième et troisième parties.

D.1.5 Contributions

Notre travail se concentre principalement sur l'indexation sémantique des documents multimédias. Plus précisément, nous nous intéressons à l'indexation sémantique des concepts dans les documents multimédias (images et vidéos) en utilisant les méthodes d'apprentissage actif que nous utilisons pour construire des corpus annoté. L'objectif principal de cette thèse est de développer et d'évaluer des approches prometteuses d'apprentissage actif pour l'indexation automatique des images et des vidéos dans les ensembles de données multimédias à grande échelle. L'indexation se fait soit par le développement de nouvelles techniques soit par l'amélioration des techniques différentes dans la communauté de la recherche d'information ou de la recherche multimédia spécialement basée sur le contenu. Nous visons à ce que le système soit en mesure d'indexer un grand nombre de concepts différents. Ainsi, l'approche doit être générique et utilisable quel que soit le concept cible. En outre, nous espérons pouvoir appliquer notre système d'indexation pour l'annotation ensembles d'apprentissage de grande envergure, tels que ceux développés dans le cadre des campagnes d'évaluation " TRECVid ".

Un point important dans notre travail est l'évaluation de nos contributions. Nous souhaitons valider les modèles proposés dans des conditions optimales. Ainsi, un élément de contexte majeur pour cette thèse est la participation aux campagnes d'évaluation à grande échelle (par exemple TRECVid et ImageCLEF). Ceci implique que les approches proposées seront capables à manipuler de vastes corpus de données.

Les efforts pour atteindre les objectifs se composent de trois parties: i) le développement des descriptions appropriées des documents multimédias (les descripteurs visuels et audio), ii) le développement de techniques avancées d'apprentissage automatique pour la classification d'images et de vidéos; iii) le développement d'un système basé sur une approche d'apprentissage actif pour mener à la création de l'ensemble de développement, en demandant itérativement à des " oracles " d'annoter des documents non étiquetés de l'ensemble de données. Dans cette thèse, nous allons explorer ces trois domaines, bien que l'objectif de la recherche soit focalisé sur les deuxième et troisième parties.

1.5 Contributions Cette thèse apporte un certain nombre de contributions dans le domaine de l'indexation automatique des documents multimédias et de l'apprentissage actif. Elles sont résumées brièvement comme suit :

- L'examen de certaines questions sur la qualité des approches d'indexation d'images et de vidéos, y compris la description de l'image et les systèmes de classification qui sont utilisés pour la recherche sur l'indexation multimédia.
- La proposition d'une méthode de classification basée sur l'approche multi-apprenant pour l'indexation des concepts dans les ensembles de données multimédias très déséquilibrées, en particulier ceux contenant des images et des vidéos.
- La proposition d'une approche reclassement visant à améliorer la performance des systèmes d'indexation sémantique et la recherche de vidéo, basé sur l'homogénéité et de la nature du contenu de la vidéo.
- La proposition d'une méthode pour l'optimisation des descriptions d'image ou de vidéos, qui se compose de deux étapes: la transformation de puissance et la réduction de la dimension en utilisant l'analyse en composantes principales (ACP).
- La proposition d'une approche qui combine l'apprentissage actif standard et l'approche proposée de multi-apprenant pour l'annotation de corpus et l'indexation concept dans des ensembles de données très déséquilibrées.
- La proposition d'une nouvelle approche de l'apprentissage actif incrémental basé sur le multi-apprenant pour les annotations d'images et de vidéos.
- La proposition d'une approche appelée " nettoyage actif " pour améliorer la qualité des annotations, basée sur la sélection d'échantillons qui apparaissent comme mal annotées, pour être ré-annotés.

- Enfin, l'intégration des approches proposées pour développer un système basé sur l'apprentissage actif pour l'indexation de vidéos dans une application réelle. Ceci a été appliqué pour conduire le processus d'annotation de l'ensemble de données de développement de l'évaluation campagne TRECVID en 2010 et 2011.

Cette thèse a conduit à une publication dans une revue, cinq articles acceptés dans des conférences internationales, dix articles acceptés à des workshops internationaux et deux articles acceptés dans des conférences nationales, comme détaillé dans le l'annexe C.

D.2 Structure de thèse

Cette thèse présente les travaux réalisés pour atteindre les objectifs décrits dans la section D.1.4. La structure et le contenu de la thèse sont décrits dans ce qui suit.

Le Chapitre 2 L'état de l'art présente le contexte de ce travail. Il commence par présenter les recherches sur l'indexation et la recherche de documents multimédias en fonction de leur contenu. Il passe ensuite en revue différentes techniques d'annotation multimédia automatique trouvées dans la littérature, y compris les méthodes de description et d'optimisation d'images et de vidéos, les approches de classification et les techniques de fusion de données. En outre, il introduit l'approche d'apprentissage actif. Enfin, il décrit les mesures de performance pour l'évaluation et présente un exemple de campagne d'évaluation, TRECVID, sur les systèmes d'indexation et de recherche de vidéos.

Le Chapitre 3 L'indexation et la recherche de documents multimédias présente différentes techniques et les défis concernant l'indexation sémantique et de recherche dans les documents multimédias. Il propose une nouvelle technique basée sur l'approche multi-apprenant (ML) pour la classification d'images et de vidéos. En outre, il propose de nouvelles techniques pour améliorer les performances de la classification, y compris une approche de reclassement des échantillons dans le système de recherche de vidéos et la normalisation des descripteurs. Les expériences d'évaluation des méthodes proposées sont également présentées.

Le Chapitre 4 - Méthodes d'apprentissage actif simulé pour l'indexation multimédia propose l'intégration de l'approche multi-apprenant comme un classificateur pour l'apprenant actif, appelé ALML. Il propose également une méthode incrémental pour accélérer les performances de l'approche ALML, appelée Inc-ALML. En outre, il présente quelques expériences d'évaluation sur les modalités proposées.

Le Chapitre 5 Le nettoyage actif pour l'indexation multimédia fournit un aperçu général du problème des annotations bruitées et présente l'annotation collaborative de TRECVID. Il propose une approche, appelée " nettoyage actif " pour réduire le bruit dans les annotations. Il se termine avec des résultats et des conclusions tirées d'expériences effectuées sur la méthode.

Le Chapitre 6 Une application dans le monde réel présente notre système d'annotations vidéo appliquées à TRECVID (les TRECVID 2010 et 2011). Il présente également nos résultats au niveau des campagnes d'évaluation TRECVID 2011 et ImageCLEF 2011.

Le Chapitre 7 Conclusions et perspectives examine et conclut les résultats globaux et les contributions du travail présenté dans les chapitres précédents. Il donne ensuite quelques conseils pour les perspectives et les travaux futurs de cette thèse.

D.3 Conclusions et Perspectives

Le cadre général de ce travail est l'indexation sémantique et la recherche d'information, appliquées aux documents multimédias. Plus particulièrement, nous nous sommes intéressés à l'indexation sémantique de concepts dans les documents multimédias (images et vidéos) par les méthodes d'apprentissage actif que nous utilisons pour construire des corpus annotés. Tout au long de cette thèse, nous avons vu que les difficultés principales de cette tâche sont liés, en général, au " fossé sémantique " qui sépare le signal des descriptions sémantiques. Elles sont également liées au problème du déséquilibre entre les classes dans les ensembles de données à grande échelle, où les concepts recherchés sont pour la plupart rares. Pour l'annotation de corpus, les algorithmes d'apprentissage actif tentent de surmonter le goulot d'étranglement de l'étiquetage par la sélection itérative des échantillons les plus informatifs pour les faire étiqueter manuellement par des annotateurs. Leur objectif principal est d'augmenter la performance du système en utilisent aussi peu déchantillons étiquetés que possible, minimisant ainsi le coût de l'étiquetage des données (par exemple, le prix et le temps). Les efforts en vue de la construction des systèmes d'indexation multimédia basés sur l'apprentissage actif seffectuent à plusieurs niveaux: la description du document multimédia, la classification, la fusion et la stratégie de la sélection active.

Dans cette thèse, nous avons contribué dans les niveaux mentionnés et propose plusieurs approches qui surclassent l'état de l'art des systèmes. Les méthodes proposées ont été validées par plusieurs expérimentations qui ont été menées et évaluées à grande échelle sur des collections de référence au niveau international dans le cadre des campagnes TRECVid.

D.3.1 Les Contributions

Dans l'état de l'art, nous avons examiné les principaux éléments des approches d'indexation multimédia. La plupart des approches visent à franchir le fossé sémantique par l'extraction de descripteurs de bas niveau suivi d'une classification basée sur des méthodes d'apprentissage supervisé. Nous avons exploré plusieurs descripteurs de bas niveau, qui peuvent être extraits de différentes modalités (par exemple, visuelle et sonore) de documents multimédia. Il est recommandé d'utiliser des descripteurs qui sont discriminants afin d'améliorer le système d'indexation. Les systèmes peuvent également intégrer plusieurs de ces descripteurs dans le cadre dun processus de fusion. La fusion des modalités peut sopérer soit au début, soit à la fin du processus dindexation. En particulier, nous avons étudié les approches de fusion dites précoces et tardives. Dans le premier cas, il sagit de combiner les descripteurs issus de chaque source et, dans le deuxième cas, de combiner les scores issus des phases de classification.

La classification est une étape nécessaire pour l'indexation des documents par des concepts mais qui toutefois ne résout que partiellement le problème de fossé sémantique. Il existe différentes méthodes de classification, en particulier génératives et discriminatives. L'état de l'art montre que, dans l'indexation multimédia les approches discriminatives sont plus efficaces que les méthodes génératives. Le classifieur le plus utilisé est la Machine à Vecteurs de Support (SVM), qui permet l'utilisation de noyaux pour projeter les descripteurs dans des espaces de plus grande dimension. Nous avons également passé en revue différents scénarios d'utilisation des méthodes d'apprentissage actif. En outre, nous avons mis en place trois stratégies bien connues qui sont utilisées avec les méthodes d'apprentissage actif pour la sélection des nouveaux échantillons à annoter. Dans ce qui suit, nous présentons les conclusions tirées de nos contributions.

D.3.1.1 Systèmes d'indexation

Nous avons contribué en trois étapes principales des systèmes d'indexation multimédia:

- Nous avons proposé et évalué une approche pour traiter le problème du déséquilibre entre les classes dans l'indexation des documents de multimédia à grande échelle, notée *multi-apprenants* (ML). L'idée de base est de sélectionner plusieurs sous-ensembles de la classe majoritaire et combiner chaque sous-ensemble avec les échantillons de la classe minoritaire. Cela va créer plusieurs ensembles d'apprentissage. Un classificateur est entraîné sur chaque ensemble d'apprentissage et plusieurs modèles ainsi sont obtenus. Pour l'indexation, les échantillons sont prédits par ces modèles et les résultats sont fusionnés pour produire des scores finaux. Nous avons montré que l'approche multi-apprenant est adaptée pour le problème des concepts rares (qui conduit à un fort déséquilibre numérique entre les classes). L'évaluation a été faite dans le contexte de la tâche de détection de concepts des campagnes TRECVID 2008 et 2009. Elle a été faite sur plusieurs types de descripteurs et testée avec trois types de classifieurs : SVM linéaire, SVM avec noyaux gaussiens et la régression logistique. Nous avons montré que l'approche multi-apprenant dépasser notablement l'approche mono-apprenant. Dans ce contexte, le classificateur basé sur la régression logistique fonctionne mieux que le classifieur SVM linéaire, mais moins bien que le classifieur SVM avec noyau gaussien. Par ailleurs, les résultats obtenus ont été relativement stables par rapport au type du descripteur utilisé.

- Nous avons proposé une méthode de reclassement qui améliore les performances des systèmes d'indexation sémantique et de recherche des vidéos. L'algorithme fonctionne en réévaluant les scores des plans des vidéos en considérant la nature homogène de la vidéo à laquelle ils appartiennent. Plus précisément, le score de chaque plan est lissé par celui des plans voisins dans la même vidéo. La méthode proposée a été évaluée et validée sur un ensemble de données de contenu non-homogène et un ensemble de données de contenu homogène, à savoir TRECVID 2008 et 2010 respectivement. Les résultats expérimentaux ont montré que la méthode de reclassement proposée a amélioré significativement les performances du système. Le système a été amélioré d'environ 11-13% et 18% respectivement sur les vidéos aux contenus non homogènes et homogènes.

- Nous avons évalué la normalisation en loi de puissance et nous l'avons comparée à quatre autres techniques typiques de normalisation. Les évaluations ont été effectuées sur la collection TRECVID 2010 en utilisant différents types de descripteurs. L'approche multi-SVM avec noyau Gaussien a été utilisée comme classificateur avec la distance Euclidienne ou la distance Chi-2. La normalisation en loi de puissance a également été évaluée après l'application des techniques de l'Analyse en Composantes Principales (ACP) pour la réduction de la dimensionnalité des descripteurs. Les résultats expérimentaux ont montré que la normalisation fondée en loi de puissance est la meilleure en particulier en combinaison avec la distance Euclidienne. La transformation de puissance rend le SVM avec la distance euclidienne aussi efficace que celui avec la distance Chi-2. De plus, en appliquant la normalisation en loi de puissance avec la distance Euclidienne et la réduction de dimensionnalité basée sur l'ACP, une diminution du temps de calcul supplémentaire a été observée. Cependant, le système a fonctionné aussi bien que sans ACP.

D.3.1.2 L'apprentissage actif et nettoyage

Nous avons également contribué à et proposé différentes approches pour l'application de l'apprentissage actif à l'annotation des corpus multimédia. Ceci comprend des méthodes qui permettent d'améliorer la performance du système et la qualité des annotations.

- Nous avons proposé une nouvelle méthode basée sur la combinaison d'un apprentissage active standard et l'approche multi-apprenant pour l'indexation multimédia, noté ALML. Notre idée fondamentale est d'appliquer l'approche des apprenants multiples comme un classifieur pour l'apprenant actif. Cette méthode a été évaluée sur l'annotation de corpus et la tâche d'indexation par concept en utilisant un ensemble de données très déséquilibré, à savoir celui de TRECVID 2008. Nous avons procédé à l'évaluation au moyen de quatre différents types de descripteurs de vidéo, avec deux types de classificateurs (par exemple la régression logistique et SVM avec noyau RBF) et avec deux différentes stratégies d'apprentissage actif : l'échantillonnage par la pertinence ou par l'incertitude. Les résultats expérimentaux montrent que l'approche multi-apprenant augmente de manière significative l'efficacité de l'apprentissage actif. Sur l'ensemble de données considéré, la meilleure performance pour chaque descripteur, est atteinte lorsque 15 à 30% du corpus a été annoté, et pour leur fusion lorsque 10 à 15% du corpus a été annotés.

- Nous avons étendu l'approche ALML proposée, et proposé un nouvel algorithme incrémental pour l'apprentissage actif, basé sur les apprenants multiples de SVM, notée Inc-ALML. L'approche ALML a montré des bonnes performances, en particulier dans le cas des ensembles de données très déséquilibrées. Elle implique cependant un grand nombre de calculs. Nous avons évalué l'approche Inc-ALML dans les mêmes conditions que dans les expériences précédentes (ALML). Les résultats expérimentaux ont montré que le système a atteint presque les mêmes performances que ALML, tandis que la nouvelle approche a économisé 50 à 63% du temps de calcul.

- Nous avons présenté l'approche de *nettoyage actif*, ce qui a permis de compléter l'approche d'apprentissage actif dans l'annotation collaborative de TRECVID. Nous avons proposé la stratégie cross-val, qui sélectionne les fractions des échantillons qui doivent être réannoté (les échantillons mal prédits). Nous avons évalué l'approche de nettoyage actif sur la collection de TRECVID 2007, avec des annotations complètes (trois annotations par exemple). La méthode a été appliquée avec différentes fractions de la stratégie de cross-val. Les résultats expérimentaux ont montré qu'une meilleure performance peut être atteinte avec une ré-annotation de 10% des échantillons négatifs ou 5% de tous les échantillons annotés sélectionnés par la stratégie de nettoyage proposé. De plus, avec une stratégie appropriée, l'utilisation d'une petite fraction des annotations pour le nettoyage, améliore les performances beaucoup plus que l'utilisation de la même fraction pour ajouter plus dannotations simples, comme dans le cas d'un nettoyage a posteriori.

- Nous avons présenté l'application de notre système dans le monde réel pour l'annotation des documents multimédia. Le système a été utilisé pour mener les annotations de l'ensemble de développement de TRECVID 2011. Nous avons utilisé l'apprentissage actif avec l'approche de multi-SVM et la stratégie d'échantillonnage par la pertinence (relevance sampling), puisque l'objectif était de fournir autant déchantillons positifs que possible pour réduire le déséquilibre entre les classes. La qualité des annotations a été renforcée par la démarche dune étape de nettoyage actif. Le système a fonctionné pendant quatre semaines sur 500 concepts. 40 laboratoires ont participé à la tâche d'annotation collaborative. Plus de 4 millions d'annotations directes ont été produites sur 346 des 500 concepts. La campagne de TRECVID 2011 a ensuite été basée sur les annotations de ces 346 concepts fournies par notre système.

- Nous avons présenté notre participation à la tâche d'indexation sémantique (SIN) de TRECVid 2011, pour laquelle nous avons appliqué notre approche pour la tâche de classification. Dans cette campagne, notre meilleur score d'évaluation était 0,1529 (en MAP) qui nous classe à la troisième place des 19 groupes qui ont participé à la tâche complète.

D.3.2 Perspectives

Outre l'amélioration de la performance des systèmes d'indexation multimédia, y compris les descripteurs, la classification et les techniques de fusion, nous pointons trois défis majeurs qui peuvent être poursuivis dans le prolongement des travaux menés dans cette thèse.

D.3.2.1 Le démarrage à froid

Le problème du démarrage à froid est défini comme la construction de l'ensemble initial d'apprentissage, qui permettra damorcer l'apprenant actif pour l'annotation d'un ensemble de données pour un concept donné. Comme nous l'avons présenté dans cette thèse, traditionnellement, un apprenant actif peut être amorcé par sélection aléatoire d'un sous-ensemble, qui contient quelques échantillons des classes considérées. Cependant, en raison de la rareté des concepts cibles, il est difficile de généraliser cela dans les applications du monde réel.

Une alternative est l'apprentissage d'un classifieur sur des ensembles extérieurs déjà annotés et de faire des prédictions initiales sur les échantillons de l'ensemble de données. Le problème devient crucial quand il n'y a pas d'annotations du tout, même sur dautres collections. Dans ce cas, ces techniques ne peuvent pas être utiles. Le concept peut alors être amorcé en utilisant d'autres concepts connexes, par exemple, nous pouvons commencer les annotations du concept golf à partir des échantillons positifs du concept sport. L'utilisation d'ontologies est alors nécessaire. Cependant, cela reste un très grand défi.

La question de démarrage à froid dans les annotations multimédia peut également être considérée comme le problème de l'apprentissage à zéro des données (Larochelle *et al.* [2008]) dans l'apprentissage automatique dans lequel l'algorithme doit construire des modèles à partir de données non étiquetées. Le problème est de construire un modèle de classification d'un nouveau concept, qui peut ensuite être utilisés pour démarrer les annotations du concept.

D.3.2.2 Apprentissage actif sur de très grands ensembles de données

Comme nous l'avons montré dans cette thèse, dans une approche dapprentissage actif pour l'indexation multimédia basée sur le " pooling " , le but est de demander des étiquettes pour les échantillons qui semblent être le plus informatif. L'approche largement utilisée pour trouver ces échantillons informatifs est de générer un modèle dapprentissage, qui est appris en utilisant d'un classifieur (par exemple, le SVM). Pour chaque échantillon, le modèle généré est utilisé pour prédire le score de probabilité et les échantillons les plus informatifs sont trouvés en classant les échantillons, en fonction de leurs scores prédits et la fonction de sélection (i.e. Q) de l'apprenant actif. Toutefois, lorsque l'application de l'apprenant actif se fait sur des bases de données à très grande échelle, le temps de prédiction sur tous les échantillons tous est critique. Il peut être impossible d'appliquer de façon exhaustive la prédiction de tous les échantillons non étiquetés à chaque itération. Par conséquent, pour exploiter les " pools " non étiqutés, une méthode de recherche rapide est nécessaire. Une solution à ce problème est d'appliquer la prédiction à chaque itération sur une partie seulement des échantillons non étiquetés. Dans ce cas, un défi est : comment choisir les échantillons à prédire par le modèle appris?

Récemment, Gorisse *et al.* [2012] ont présenté un nouvelle technique de hachage " Locality Sensible Hashing " (LSH) schéma adapté à la distance du Chi-2 pour la recherche approximative des proches voisines en espaces de grande dimension. Cette méthode est basée sur le sous-échantillonnage des échantillons non étiquetés. Les auteurs ont montré l'efficacité de la méthode avec l'algorithme LSH avec distance Euclidienne dans le contexte de la recherche d'images sur les bases de données d'images réelles. Nous avons pour objectif de tester leur méthode dans le cadre de l'utilisation de l'apprentissage actif sur de très grandes bases de données.

D.3.2.3 Crowd-sourcing: la qualité et de la surveillance Annotations Annotateurs

Employer des techniques de " crowd-sourcing " pour la tâche d'indexation multimédia, est l'un des principaux points de vue de cette thèse. Avec la question du cimpromis de coût-efficacité, le recrutement en ligne des annotateurs (par exemple MTurk) apporte de nouveaux défis. Les annotateurs ne sont pas spécifiquement formés pour l'annotation, et pourraient ne pas être très investis dans la production dannotations de bonne qualité puisque la plupart d'entre eux ne le font que pour se faire payer. Par conséquent, les annotations obtenus peuvent être bruitées par nature et peuvent nécessiter des études supplémentaires. Plusieurs questions intéressantes se posent immédiatement sur la façon d'utiliser de façon optimale ce type dannotations : Comment pouvons-nous gérer les différences entre les travailleurs en termes de la qualité des annotations qu'ils fournissent ? Comment pouvons-nous détecter et contrôler les annotateurs produisant les annotations les plus bruitées ? Quelle est l'utilité des annotations bruitées pour la création de systèmes d'indexation ? Est-il possible d'identifier des exemples ambigus via les désaccords entre annotateur s ? Ces questions sont étroitement liées à l'apprentissage actif où nous avons l'objectif est de produire moins d'annotation mais avec une qualité supérieure. L'approche de nettoyage actif, donnés dans cette thèse, est considérée comme un première étape dans cette direction. Il a été appliqué à un large ensemble de données mais il doit encore être vérifié et utilisée avec de très grands ensembles de données et bien plus dannotateurs, par exemple, dans le cas dune utilisation avec MTurk.

Bibliography

ABDEL-MOTTALEB, M. & KRISHNAMACHARI, S. (2004). Multimedia descriptions based on mpeg-7: extraction and applications. *IEEE Transactions on Multimedia*, **6**, 459–468. 25

ALBATAL, R. (2010). *Annotation automatique d'images base de Phrases Visuelles*. Ph.D. thesis, Université de Grenoble. 21

ANGLUIN, D. (1988). Queries and concept learning. *Machine Learning*, **2**, 319–342. 2, 36, 140

ANGLUIN, D. (2001). Queries revisited. In *Algorithmic Learning Theory, 12th International Conference (ALT)*, 12–31, Springer. 36

ATLAS, L.E., COHN, D.A. & LADNER, R.E. (1989). Training connectionist networks with queries and selective sampling. In D.S. Touretzky, ed., *Advances in Neural Information Processing Systems 2, [NIPS Conference, Denver, Colorado, USA, November 27-30, 1989]*, 566–573, Morgan Kaufmann. 36

AYACHE, S. (2007). *Indexation de documents vidéos par concepts et par fusion de caractristiques audio, image et texte*. Ph.D. thesis, INPG. 11

AYACHE, S. & QUÉNOT, G. (2007a). Evaluation of active learning strategies for video indexing. *Signal Processing: Image Communication*. 35, 73, 77, 83, 92

AYACHE, S. & QUÉNOT, G. (2007b). Video corpus annotation using active learning. In *Proceedings of the TRECVid 2007*. 105

AYACHE, S., QUÉNOT, G., GENSEL, J. & SATOH, S. (2006). Using Topic Concepts for Semantic Video Shots Classification. In *International Conference on Image and Video Retrieval CIVR'06*, Tempe USA. 110

AYACHE, S., QUÉNOT, G. & GENSEL, J. (2007). Classifier fusion for svm-based multimedia semantic indexing. In G. Amati, C. Carpineto & G. Romano, eds., *Advances in Information Retrieval, 29th European Conference on IR Research, ECIR 2007, Rome, Italy, April 2-5, 2007, Proceedings*, vol. 4425 of *Lecture Notes in Computer Science*, 494–504, Springer. 32, 34, 82

BAKKER, E.M. & LEW, M.S. (2002). Semantic video retrieval using audio analysis. In M.S. Lew, N. Sebe & J.P. Eakins, eds., *CIVR*, vol. 2383 of *Lecture Notes in Computer Science*, 271–277, Springer. 19

BARAM, Y., EL-YANIV, R. & LUZ, K. (2004). Online choice of active learning algorithms. *J. Mach. Learn. Res.*, **5**, 255–291. 36

BARRON, J., FLEET, D. & BEAUCHEMIN, S. (1994). Performance of optical flow techniques. *International Journal of Computer Vision*, **12**, 43–77. 15

BATISTA, G.E.A.P.A., PRATI, R.C. & MONARD, M.C. (2004). A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor. Newsl.*, **6**, 20–29. 46

BAY, H., TUYTELAARS, T. & GOOL, L.V. (2006). Surf: Speeded up robust features. In *In ECCV*, 404–417. 17

BELLILI, A. (2001). An hybrid mlp-svm handwritten digit recognizer. In *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, 28–, IEEE Computer Society, Washington, DC, USA. 26

BENOIT, A., CAPLIER, A., DURETTE, B. & HERAULT, J. (2010). Using human visual system modeling for bio-inspired low level image processing. *Comput. Vis. Image Underst.*, **114**, 758–773. 132

BISHOP, C.M. (2007). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st edn. 5, 46, 144

BLOCH, I., LE CADRE, J.P. & MATRE, H. (2003). Approches probabilistes et statistiques. In I. Bloch, ed., *Fusion d'informations en traitement du signal et des images*, Trait IC2, chap. 6, 87–118, Herms. 31

BLUM, A.L. & LANGLEY, P. (1997). Selection of relevant features and examples in machine learning. *ARTIFICIAL INTELLIGENCE*, **97**, 245–271. 26

BORDES, A., ERTEKIN, S., WESTON, J. & BOTTOU, L. (2005). Fast kernel classifiers with online and active learning. *J. Mach. Learn. Res.*, **6**, 1579–1619. 84

BORECZKY, J.S. & ROWE, L.A. (1996). Comparison of video shot boundary detection techniques. In *Storage and Retrieval for Image and Video Databases (SPIE)*, 170–179. 11

BOSCH, A., ZISSERMAN, A. & MUÑOZ, X. (2008). Scene classification using a hybrid generative/discriminative approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, **30**, 712–727. 18

BOUTELL, M.R., LUO, J., SHEN, X. & BROWN, C.M. (2004). Learning multi-label scene classification. *Pattern Recognition*, **37**, 1757–1771. 49

BREIMAN, L. (1996). Bagging predictors. *Mach. Learn.*, **24**, 123–140. 29

BREIMAN, L. (2001). Random forests. In *Machine Learning*, 5–32. 30

BREIMAN, L., FRIEDMAN, J., OLSHEN, R. & STONE, C. (1984). *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, new edition **?**. 45, 46

CHANG, C.C. & LIN, C.J. (2001). *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`. 43, 50

CHANG, C.Y., WANG, H.J. & LI, C.F. (2009). Semantic analysis of real-world images using support vector machine. *Expert Syst. Appl.*, **36**, 10560–10569. 4, 142, 143

CHANG, S.F., SIKORA, T. & PURI, A. (2001). Overview of the mpeg-7 standard. *IEEE Trans. Circuits Syst. Video Techn.*, **11**, 688–695. 23, 24

CHAPELLE, O., HAFFNER, P. & VAPNIK, V. (1999). Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks*, **10**, 1055–1064. 27

CHAWLA, N.V., BOWYER, K.W., HALL, L.O. & KEGELMEYER, W.P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, **16**, 321–357. 5, 46, 144

CHAWLA, N.V., JAPKOWICZ, N. & KOTCZ, A. (2004). Editorial: special issue on learning from imbalanced data sets. *SIGKDD Explorations*, **6**, 1–6. 45

CHENG, Y.C. & CHEN, S.Y. (2003). Image classification using color, texture and regions. *Image Vision Comput.*, **21**, 759–776. 131

CHUA, T.S., CHEN, L. & WANG, J. (2002). Stratification approach to modeling video. *Multimedia Tools Appl.*, **16**, 79–97. 11

COHN, D., ATLAS, L. & LADNER, R. (1994). Improving generalization with active learning. *Mach. Learn.*, **15**, 201–221. 36

COHN, D.A., GHAHRAMANI, Z. & JORDAN, M.I. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research*, **4**, 129–145. 36

CORTES, C. & VAPNIK, V. (1995). Support-vector networks. *Machine Learning*, **20**, 273–297. 27

COURTNEY, J. (1997). Automatic video indexing via object motion analysis. *Pattern Recognition*, **30**, 607–625. 15

COVER, T. & HART, P.E. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, **13**, 21–27. 26

DEERWESTER, S., DUMAIS, S.T., FURNAS, G.W., LANDAUER, T.K. & HARSHMAN, R. (1990). Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, **41**, 391–407. 22

DELEZOIDE, B., PRECIOSO, F., GOSSELIN, P., REDI, M., MERIALDO, B., GRANJON, L., PELLERIN, D., ROMBAUT, M., JÉGOU, H., VIEUX, R., BUGEAU, A., MANSENCAL, B., BENOIS-PINEAU, J., BOUJUT, H., AYACHE, S., SAFADI, B., THOLLARD, F., QUÉNOT, G., BREDIN, H., CORD, M., BENOT, A., LAMBERT, P., STRAT, T., RAZIK, J., PARIS, S. & GLOTIN, H. (2011). IRIM at TRECVid 2011: High Level Feature Extraction and Instance

Search. In *TREC Video Retrieval Evaluation workshop*, National Institute of Standards and Technology, Gaithersburg, MD USA. 110, 112

DIETTERICH, T.G. (2000). Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, MCS '00, 1–15, Springer-Verlag, London, UK. 29

DIETTERICH, T.G. (2002). *Ensemble Learning*. The MIT Press. 30

ELFADEL, I.M. & PICARD, R.W. (1993). Gibbs random fields, co-occurrences, and texture modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **16**, 24–37. 15

FEI-FEI, L. & PERONA, P. (2005). A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, 524 – 531 vol. 2. 20

FERRARI, V., TUYTELAARS, T. & GOOL, L. (2006). Simultaneous object recognition and segmentation from single or multiple model views. *Int. J. Comput. Vision*, **67**, 159–188. 15

FINE, S., NAVRATIL, J. & GOPINATH, R. (2001). A hybrid gmm/svm approach to speaker identification. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, vol. 1, 417 –420 vol.1. 21

FLICKNER, M., SAWHNEY, H., NIBLACK, W., ASHLEY, J., HUANG, Q., DOM, B., GORKANI, M., HAFNER, J., LEE, D., PETKOVIC, D., STEELE, D. & YANKER, P. (1995). Query by image and video content: The qbic system. *Computer*, **28**, 23–32. 15

FREUND, Y. & SCHAPIRE, R.E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, **55**, 119–139. 30

FREUND, Y., SEUNG, H.S., SHAMIR, E. & TISHBY, N. (1997). Selective sampling using the query by committee algorithm. *Mach. Learn.*, **28**, 133–168. 36

GANCHEV, T., FAKOTAKIS, N. & KOKKINAKIS, G. (2005). Comparative evaluation of various mfcc implementations on the speaker verification task. In *10th International Conference on Speech and Computer (SPECOM-2005)*, 191–194. 19

GÖNEN, M. & ALPAYDIN, E. (2011). Multiple kernel learning algorithms. *J. Mach. Learn. Res.*, **12**, 2211–2268. 34

GORISSE, D., PRECIOSO, F., GOSSELIN, P., GRANJON, L., PELLERIN, D., ROMBAUT, M., BREDIN, H., KOENIG, L., LACHAMBRE, H., EL KHOURY, E., VIEUX, R., MANSENCAL, B., ZHOU, Y., BENOIS-PINEAU, J., JÉGOU, H., AYACHE, S., SAFADI, B., TONG, Y., THOLLARD, F., QUÉNOT, G., BENOIT, A. & LAMBERT, P. (2010). IRIM at TRECVid 2010: High Level Feature Extraction and Instance Search. In *TREC Video Retrieval Evaluation workshop*, National Institute of Standards and Technology, Gaithersburg, MD USA. 57, 58, 61, 65, 113

GORISSE, D., CORD, M. & PRECIOSO, F. (2011). Salsas: Sub-linear active learning strategy with approximate k-nn search. *Pattern Recognition*, **44**, 2343–2357. 131

GORISSE, D., CORD, M. & PRECIOSO, F. (2012). Locality-sensitive hashing for chi2 distance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **34**, 402–409. 125, 152

GOSSELIN, P.H. & CORD, M. (2008). Active learning methods for interactive image retrieval. *IEEE Transactions on Image Processing*, **17**, 1200–1211. 36

HARALICK, R. (1979). Statistical and structural approaches to texture. *Proceedings of the IEEE*, **67**, 786 – 804. 15

HARE, J.S. & LEWIS, P.H. (2005). Saliency-based models of image content and their application to auto-annotation by semantic propagation. In *In Proceedings of Multimedia and the Semantic Web / European Semantic Web Conference*. 22

HARRIS, C. & STEPHENS, M. (1988). A Combined Corner and Edge Detection. In *Proceedings of The Fourth Alvey Vision Conference*, 147–151. 17

HART, P.E. (1968). The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, **14**, 515–516. 46

HAUPTMANN, A.G., LIN, W.H., YAN, R., YANG, J. & CHEN, M.Y. (2006). Extreme video retrieval: joint maximization of human and computer performance. In *Proceedings of the 14th annual ACM international conference on Multimedia*, MULTIMEDIA ’06, 385–394, ACM, New York, NY, USA. 36

HERBRICH, R., GRAEPEL, T. & OBERMAYER, K. (1999). Support vector learning for ordinal regression. In *Ninth Intl. Conf. on Artificial Neural Networks*, 97–102. 55

HOI, S.C.H., JIN, R. & LYU, M.R. (2006). Large-scale text categorization by batch mode active learning. In *Proceedings of the 15th international conference on World Wide Web*, WWW ’06, 633–642, ACM, New York, NY, USA. 36

HORN, B.K.P. & SCHUNCK, B.G. (1981). Determining optical flow. *Artificial Intelligence*, **17**, 185–203. 15

HOWE, J. (2006). The rise of crowdsourcing. *Wired Magazine*, **14**. 91

HSUEH, P.Y., MELVILLE, P. & SINDHWANI, V. (2009). Data quality from crowdsourcing: a study of annotation selection criteria. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, HLT ’09, 27–35, Association for Computational Linguistics, Stroudsburg, PA, USA. 92

IDRISSI, N., MARTINEZ, J. & ABOUTAJDINE, D. (2009). Bridging the semantic gap for texture-based image retrieval and navigation. *Journal of Multimedia*, **4**, 277–283. 4, 142, 143

IYENGAR, G. & NOCK, H.J. (2003). Discriminative model fusion for semantic concept detection and annotation in video. In L.A. Rowe, H.M. Vin, T. Plagemann, P.J. Shenoy & J.R. Smith, eds., *Proceedings of the Eleventh ACM International Conference on Multimedia, November 2-8, 2003, Berkeley, CA, USA*, 255–258, ACM. 26

JAAKKOLA, T., DIEKHANS, M. & HAUSSLER, D. (2000). A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology*, **7**, 95–114. 21

JAAKKOLA, T.S. & HAUSSLER, D. (1999). Exploiting generative models in discriminative classifiers. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, 487–493, MIT Press, Cambridge, MA, USA. 21

JAIN, A.K., DUIN, R.P.W. & MAO, J. (2000). Statistical pattern recognition: A review. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, **22**, 4–37. 26

JAPKOWICZ, N. & STEPHEN, S. (2002). The class imbalance problem: A systematic study. *Intell. Data Anal.*, **6**, 429–449. 5, 45, 144

JAU-LING, S. & LING-HWEI, C. (2002). Color Image Retrieval Based on Primitives of Color Moments. 19–27. 14

JEBARA, T. (2004). *Machine Learning: Discriminative and Generative*. Kluwer Academic. 26

JÉGOU, H., DOUZE, M. & SCHMID, C. (2009). On the burstiness of visual elements. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09)*, 1169–1176, IEEE Computer society, Miami, United States. 65

JÉGOU, H., PERRONNIN, F., DOUZE, M., SÁNCHEZ, J., PÉREZ, P. & SCHMID, C. (2011). Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, qUAERO. 65

JIANG, Y.G., NGO, C.W. & YANG, J. (2007). Towards optimal bag-of-features for object categorization and semantic video retrieval. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, CIVR '07, 494–501, ACM, New York, NY, USA. 20, 55

JIANG, Y.G., YANAGAWA, A., CHANG, S.F. & NGO, C.W. (2008). CU-VIREO374: Fusing Columbia374 and VIREO374 for Large Scale Semantic Concept Detection. Tech. rep., Columbia University ADVENT #223-2008-1. 104

JOSHI, A.J., PORIKLI, F. & PAPANIKOLOPOULOS, N. (2009). Multi-class active learning for image classification. In *CVPR*, 2372–2379. 92

JUTTEN, C. & HERAULT, J. (1991). Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, **24**, 1 – 10. 23

KANKANHALLI, M.S. & CHUA, T.S. (2000). Video modeling using strata-based annotation. *IEEE MultiMedia*, **7**, 68–74. 11

KENNEDY, L.S. & CHANG, S.F. (2007). A reranking approach for context-based concept fusion in video indexing and retrieval. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, CIVR '07, 333–340, ACM, New York, NY, USA. 55

KITTUR, A., CHI, E.H. & SUH, B. (2008). Crowdsourcing user studies with mechanical turk. In *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, 453–456, ACM, New York, NY, USA. 92

KOBLA, V., DOERMANN, D.S., LIN, K.I. & FALOUTSOS, C. (1997). Compressed-domain video indexing techniques using dct and motion vector information in mpeg video. In *Storage and Retrieval for Image and Video Databases (SPIE)'97*, 200–211. 15

KOMAREK, P. (2005). *LR-TRIRLS: Logistic regression for binary classification*. Software available at `http://komarix.org/ac/lr`. 50

KUBAT, M. & MATWIN, S. (1997). Addressing the curse of imbalanced training sets: One-sided selection. In *ICML*, 179–186, Morgan Kaufmann, Nashville, Tennessee, USA. 46

KUHN, H.W. & TUCKER, A.W. (1950). Nonlinear programming. In J. Neyman, ed., *Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability*, 481–492, University of California Press, Berkeley, CA, USA. 84

KUMAR, A. & LEASE, M. (2011). Modeling annotator accuracies for supervised learning. In *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining (CSDM) at the Fourth ACM International Conference on Web Search and Data Mining (WSDM)*, 19–22, Hong Kong, China. 92

KUNCHEVA, L.I. & WHITAKER, C.J. (2003). Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy. *Machine Learning*, **51**, 181–207. 29

LAPTEV, I. (2005). On space-time interest points. *Int. J. Comput. Vision*, **64**, 107–123. 132

LAPTEV, I. & LINDEBERG, T. (2003). Space-time interest points. In *IN ICCV*, 432–439. 13, 17

LAROCHELLE, H., ERHAN, D. & BENGIO, Y. (2008). Zero-data learning of new tasks. In D. Fox & C.P. Gomes, eds., *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, 646–651, AAAI Press. 124, 152

LAURIKKALA, J. (2001). Improving identification of difficult small classes by balancing class distribution. In *Proceedings of the 8th Conference on AI in Medicine in Europe: Artificial Intelligence Medicine*, AIME '01, 63–66, Springer-Verlag, London, UK, UK. 46

LAZEBNIK, S., SCHMID, C. & PONCE, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, 2169 – 2178. 20

LEIBE, B. & SCHIELE, B. (2003). Interleaved object categorization and segmentation. In *In BMVC*, 759–768. 15

LEWIS, D.D. & CATLETT, J. (1994). Heterogeneous uncertainty sampling for supervised learning. In W.W. Cohen & H. Hirsh, eds., *Proceedings of ICML-94, 11th International Conference on Machine Learning*, 148–156, Morgan Kaufmann Publishers, San Francisco, US, New Brunswick, US. 38

LEWIS, D.D. & GALE, W.A. (1994). A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '94, 3–12, Springer-Verlag New York, Inc., New York, NY, USA. 36, 37, 38

LI, Y., SHAPIRO, L.G. & BILMES, J.A. (2005). A generative/discriminative learning algorithm for image classification. In *Proceedings of the Tenth IEEE International Conference on Computer Vision - Volume 2*, ICCV '05, 1605–1612, IEEE Computer Society, Washington, DC, USA. 26

LIM, J.H. (2001). Building visual vocabulary for image indexation and query formulation. *Pattern Anal. Appl.*, **4**, 125–139. 3, 142

LIN, W.H., JIN, R. & HAUPTMANN, A.G. (2002). Meta-classification of multimedia classifiers. In *KDMCD*, 21–27. 33

LINDEBERG, T. (1998). Feature detection with automatic scale selection. *Int. J. Comput. Vision*, **30**, 79–116. 17

LIU, J., LAI, W., HUA, X.S., HUANG, Y. & LI, S. (2007a). Video search re-ranking via multi-graph propagation. In *Proceedings of the 15th international conference on Multimedia*, MULTIMEDIA '07, 208–217, ACM, New York, NY, USA. 55

LIU, X.Y., WU, J. & ZHOU, Z.H. (2009). Exploratory undersampling for class-imbalance learning. *Trans. Sys. Man Cyber. Part B*, **39**, 539–550. 46

LIU, Y., ZHANG, D., LU, G. & MA, W.Y. (2007b). A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, **40**, 262–282. 4, 143

LIU, Y., ZHAO, W.L., NGO, C.W., XU, C.S. & LU, H.Q. (2010). Coherent bag-of audio words model for efficient large-scale video copy detection. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, CIVR '10, 89–96, ACM, New York, NY, USA. 20

LODHI, H., SAUNDERS, C., SHAWE-TAYLOR, J., CRISTIANINI, N. & WATKINS, C. (2002). Text classification using string kernels. *J. Mach. Learn. Res.*, **2**, 419–444. 33

LOWE, D. (1999). Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, 1150 –1157 vol.2. 17

LOWE, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, **60**, 91–110. 16, 17, 18

LU, L., JIANG, H. & ZHANG, H. (2001). A robust audio classification and segmentation method. In *ACM Multimedia*, 203–211. 19

MACQUEEN, J.B. (1967). Some methods for classification and analysis of multivariate observations. In L.M.L. Cam & J. Neyman, eds., *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, 281–297, University of California Press. 20

MANJUNATH, B.S. & MA, W.Y. (1996). Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.*, **18**, 837–842. 15

MARTINEZ, J., KOENEN, R. & PEREIRA, F. (2002). Mpeg-7: the generic multimedia content description standard, part 1. *Multimedia, IEEE*, **9**, 78 –87. 23

MCCALLUM, A. & NIGAM, K. (1998). Employing em and pool-based active learning for text classification. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, 350–358, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 36

MENSINK, T., CSURKA, G., PERRONNIN, F., SÁNCHEZ, J. & VERBEEK, J. (2010). Lear and xrces participation to visual concept detection task - imageclef 2010. In *In Working Notes for the CLEF 2010 Workshop*, 48. 116

M.H. SONG, S.C.K.L., J.LEE & YOO, S. (2005). Support vector machine based arrhythmia classification using reduced features. *International Journal of Control, Automation, and Systems*, **3**, 571–579. 23

MIKOLAJCZYK, K. & SCHMID, C. (2001). Indexing based on scale invariant interest points. In *Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada*, 525–531. 17

MIKOLAJCZYK, K. & SCHMID, C. (2002). An affine invariant interest point detector. In *Proceedings of the 7th European Conference on Computer Vision-Part I*, ECCV '02, 128–142, Springer-Verlag, London, UK, UK. 17

MIKOLAJCZYK, K. & SCHMID, C. (2005). A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **27**, 1615 –1630. 17

M.J. SWAIN, D.B. (1991). Color indexing. *International journal of computer vision*, **7**, 11–32. 14

NACK, F. & LINDSAY, A. (1999). Everything you wanted to know about mpeg-7: Part 2. *IEEE MultiMedia*, **6**, 64–73. 23

NAPHADE, M., SMITH, J.R., TESIC, J., CHANG, S.F., HSU, W., KENNEDY, L., HAUPTMANN, A. & CURTIS, J. (2006). Large-scale concept ontology for multimedia. *IEEE MultiMedia*, **13**, 86–91. 40, 104

NAPHADE, M.R. (2004a). On supervision and statistical learning for semantic multimedia analysis. *J. Vis. Comun. Image Represent.*, **15**, 348–369. 26

NAPHADE, M.R. (2004b). On supervision and statistical learning for semantic multimedia analysis. *J. Vis. Comun. Image Represent.*, **15**, 348–369. 31

NAPHADE, M.R. & SMITH, J.R. (2004). On the detection of semantic concepts at TRECVid. In *MULTIMEDIA'04: Proceedings of the 12th annual ACM international conference on Multimedia*, 660–667, ACM Press, New York, NY, USA. 2, 12, 13, 140

OLIVA, A. & TORRALBA, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, **42**, 145–175. 131

OVER, P., IANEVA, T., W, K. & SMEATON., A.F. (2005). TRECVid 2005 - an overview. In *TRECVid 2005 - Text REtrieval Conference TRECVid Workshop*, National Institute of Standards and Technology, MD, USA. 12, 13

OVER, P., AWAD, G., FISCUS, J., ANTONISHEK, B., F. SMEATON, A., KRAAIJ, W. & QUÉNOT, G. (2010). TRECVid 2010 – An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics. In *Proceedings of TRECVid 2010*, NIST, USA. 41

OVER, P., AWAD, G., FISCUS, J., F. SMEATON, A., KRAAIJ, W. & QUÉNOT, G. (2011). TRECVid 2011 – An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics. In *Proceedings of TRECVid 2011 (to appear)*, NIST, USA. 103, 104, 109

PARIS, S. & GLOTIN, H. (2010). Pyramidal multi-level features for the robot vision@icpr 2010 challenge. In *ICPR*, 2949–2952. 133

PEARSON, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, **2**, 559–572. 22

PERRONNIN, F. & DANCE, C.R. (2007). Fisher kernels on visual vocabularies for image categorization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, IEEE Computer Society. 21

PLATT, J.C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, 61–74. 50

POLIKAR, R. (2006). Ensemble Based Systems in Decision Making. *IEEE Circuits and Systems Magazine*, **6**, 21–45. 29, 30

PREVOST, L., OUDOT, L., MOISES, A., MICHEL-SENDIS, C. & MILGRAM, M. (2005). Hybrid generative/discriminative classifier for unconstrained character recognition. *Pattern Recogn. Lett.*, **26**, 1840–1848. 26

QI, G.J., HUA, X.S., RUI, Y., TANG, J. & ZHANG, H.J. (2008). Two-dimensional active learning for image classification. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, **0**, 1–8. 92

QUÉNOT, G. (2008). *KNNLSB: K Nearest Neighbors Linear Scan Baseline*. Software available at http://mrim.imag.fr/georges.quenot/freesoft/knnlsb/index.html. 112

QUÉNOT, G., MORARU, D. & BESACIER, L. (2003). Clips at TRECVid: Shot boundary detection and feature detection. In *TRECVid 2003 Workshop Notebook Papers*, 35–40. 105

QUÉNOT, G., DELEZOIDE, B., LE BORGNE, H., MOËLLIC, P.A., GORISSE, D., PRECIOSO, F., WANG, F., MERIALDO, B., GOSSELIN, P., GRANJON, L., PELLERIN, D., ROMBAUT, M., BREDIN, H., KOENIG, L., LACHAMBRE, H., KHOURY, E.E., MANSENCAL, B., BENOIS-PINEAU, J., JÉGOU, H., AYACHE, S., SAFADI, B., FABRIZIO, J., CORD, M., GLOTIN, H., ZHAO, Z., DUMONT, E. & AUGEREAU, B. (2009). IRIM at TRECVid 2009: High level feature extraction. In *TREC 2009 notebook*. 43, 51, 76, 96

QUÉNOT, G.M. (2001). Trec-10 shot boundary detection task: Clips system description and evaluation. In *In em 10th Text Retrieval Conference*, 13–16. 12

RAINA, R., SHEN, Y., NG, A.Y. & MCCALLUM, A. (2003). Classification with hybrid generative/discriminative models. In *In Advances in Neural Information Processing Systems 16*, MIT Press. 26

REDI, M. & MERIALDO, B. (2011). Saliency moments for image categorization. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, ICMR '11, 39:1–39:8, ACM, New York, NY, USA. 131

ROY, N. & MCCALLUM, A. (2001). Toward Optimal Active Learning through Sampling Estimation of Error Reduction. In *Proc. 18th International Conf. on Machine Learning*, 441–448, Morgan Kaufmann, San Francisco, CA. 36

RÜPING, S. (2001). Incremental learning with support vector machines. In *IEEE International Conference on Data Mining (ICDM)*, 641–642, IEEE Computer Society. 84

SAFADI, B. & QUÉNOT, G. (2010). Active learning with multiple classifiers for multimedia indexing. In *8th IEEE Int. Workshop on Content-Based Multimedia Indexing (CBMI)*, Grenoble, France. 74

SAFADI, B. & QUÉNOT, G. (2010). Active learning with multiple classifiers for multimedia indexing. *Multimedia Tools and Applications*, 1–15. 74

SAFADI, B. & QUÉNOT, G. (2010). Evaluations of multi-learners approaches for concepts indexing in video documents. In *RIAO*, 88–91, Paris, France. 47

SAFADI, B. & QUÉNOT, G. (2011a). Re-ranking by local re-scoring for video indexing and retrieval. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management (CIKM)*, 2081–2084, Glasgow, United Kingdom. 56

SAFADI, B. & QUÉNOT, G. (2011b). Re-ranking for multimedia indexing and retrieval. In *Proceedings of the 33rd European Conference on IR Research (ECIR)*, 708–711, Dublin, Ireland. 56

SAFADI, B., TONG, Y. & QUÉNOT, G. (2010a). Incremental multi-classifier learning algorithm on grid'5000 for large scale image annotation. In *Proceedings of the international workshop on Very-large-scale multimedia corpus, mining and retrieval*, VLS-MCMR '10, 1–6, ACM, Firenze, Italy. 85

SAFADI, B., YUBING, T., THOLLARD, F., QUNOT, G., EKENEL, H., GEHRIG, T. & STIFELHAGEN, R. (2010b). Quaero at TRECVid 2010: Semantic Indexing. In *TREC Video Retrieval Evaluation workshop*, National Institute of Standards and Technology, Gaithersburg, MD USA. 59

SAFADI, B., DERBAS, N., HAMADI, A., THOLLARD, F., QUÉNOT, G., JÉGOU, H., GEHRIG, T., KEMAL EKENEL, H. & STIFELHAGEN, R. (2011a). Quaero at TRECVid 2011: Semantic Indexing and Multimedi Event Detection. In *TREC Video Retrieval Evaluation workshop*, National Institute of Standards and Technology, Gaithersburg, MD USA. 65

SAFADI, B., TONG, Y. & QUÉNOT, G. (2011b). Incremental multiple classifier active learning for concept indexing in images and videos. In *Proceedings of the 17th international conference on Advances in Multimedia Modeling - Volume Part I*, MMM'11, 240–250, Springer-Verlag, Taipei, Taiwan. 85

SAFADI, B., AYACHE, S. & QUÉNOT, G. (2012). Active cleaning for video corpus annotation. In *Proceedings of the 18th international conference on Advances in Multimedia Modeling (MMM)*, MMM'12, 518–528, Klagenfurt, Austria. 93

SCHAPIRE, R.E. (1990). The strength of weak learnability. *Mach. Learn.*, **5**, 197–227. 29

SCHMID, C. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *In CVPR*, 2169–2178. 20

SCHMID, C., MOHR, R. & BAUCKHAGE, C. (2000). Evaluation of interest point detectors. *Int. J. Comput. Vision*, **37**, 151–172. 17

SCHOLKOPF, B. & SMOLA, A.J. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA. 27

SETTLES, B. (2009). Active Learning Literature Survey. Tech. Rep. 1648, University of Wisconsin–Madison. 35

SETTLES, B. & CRAVEN, M. (2008). An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, 1070–1079, Association for Computational Linguistics, Stroudsburg, PA, USA. 36

SEUNG, H.S., OPPER, M. & SOMPOLINSKY, H. (1992). Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, COLT '92, 287–294, ACM, New York, NY, USA. 36, 38

SHEN, K. & DELP, E. (1995). A fast algorithm for video parsing using mpeg compressed sequences. In *Image Processing, 1995. Proceedings., International Conference on*, vol. 2, 252 –255. 15

SHENG, V.S., PROVOST, F. & IPEIROTIS, P.G. (2008). Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, 614–622, ACM, New York, NY, USA. 92

SHILTON, A., PALANISWAMI, M., RALPH, D. & TSOI, A.C. (2005). Incremental training of support vector machines. *Neural Networks, IEEE Transactions on*, **16**, 114–131. 84

SIVIC, J. & ZISSERMAN, A. (2003). Video google: a text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, 1470 –1477 vol.2. 20

SMEATON, A.F., OVER, P. & KRAAIJ, W. (2006). Evaluation campaigns and TRECVid. In *MIR '06: Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, 321–330, ACM Press, New York, NY, USA. 4, 5, 6, 142, 144

SMEATON, A.F., OVER, P. & DOHERTY, A.R. (2010). Video shot boundary detection: Seven years of trecvid activity. *Comput. Vis. Image Underst.*, **114**, 411–418. 12

SMEULDERS, A.W.M., WORRING, M., SANTINI, S., GUPTA, A. & JAIN, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, **22**, 1349–1380. 4, 142, 143

SMITH, N., GALES, M. & NIRANJAN, M. (2001). Data-Dependent Kernels in SVM Classification of Speech Patterns. Tech. rep., Cambridge University Engineering Dept. 21

SMITH, T.G.A. & DAVENPORT, G. (1993). The stratification system - a design emvironment for random access. In *Proceedings of the Third International Workshop on Network and Operating System Support for Digital Audio and Video*, 250–261, Springer-Verlag, London, UK. 11

SNOEK, C.G.M., WORRING, M., GEMERT, J.V., MARK GEUSEBROEK, J., KOELMA, D., NGUYEN, G.P., ROOIJ, O.D. & SEINSTRA, F. (2005a). Mediamill: Exploring news video archives based on learned semantics. In *In ACM Multimedia*, 225–226. 3, 49, 142

SNOEK, C.G.M., WORRING, M. & SMEULDERS, A.W.M. (2005b). Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM international conference on Multimedia*, MULTIMEDIA '05, 399–402, ACM, New York, NY, USA. 31, 82

SNOEK, C.G.M., WORRING, M. & HAUPTMANN, A.G. (2006). Learning rich semantics from news video archives by style analysis. *ACM Trans. Multimedia Comput. Commun. Appl.*, **2**, 91–108. 2, 140

SNOW, R., O'CONNOR, B., JURAFSKY, D. & NG, A.Y. (2008). Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, 254–263, Association for Computational Linguistics, Stroudsburg, PA, USA. 91

SONNENBURG, S., RATSCH, G. & SCHAFER, C. (2006). A general and efficient multiple kernel learning algorithm. In Y. Weiss, B. Schölkopf & J. Platt, eds., *Advances in Neural Information Processing Systems 18*, 1273–1280, MIT Press, Cambridge, MA. 34

STRICKER, M.A. & ORENGO, M. (1995). Similarity of Color Images. In *Storage and Retrieval for Image and Video Databases (SPIE)*, 381–392. 14

TAHIR, M.A., KITTLER, J., MIKOLAJCZYK, K. & YAN, F. (2009a). A multiple expert approach to the class imbalance problem using inverse random under sampling. In *MCS '09: Proceedings of the 8th International Workshop on Multiple Classifier Systems*, 82–91, Springer-Verlag, Berlin, Heidelberg. 2, 46, 49, 51, 140

TAHIR, M.A., KITTLER, J., YAN, F. & MIKOLAJCZYK, K. (2009b). Concept learning for image and video retrieval: The inverse random under sampling approach. In *Eusipco 2009, 17th European Signal Processing Conference*. 46, 49, 51

TAMURA, H., MORI, S. & YAMAWAKI, T. (1978). Textural features corresponding to visual perception. *Systems, Man and Cybernetics, IEEE Transactions on*, **8**, 460 –473. 15

THOMPSON, C.A., CALIFF, M.E. & MOONEY, R.J. (1999). Active learning for natural language parsing and information extraction. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, 406–414, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 36

TOMEK, I. (1976). Two modifications to cnn. *IEEE Transactions on Systems, Man and Communications, SMC-6*, 769–772. 46

TONG, S. & CHANG, E. (2001). Support vector machine active learning for image retrieval. In *Proceedings of the ninth ACM international conference on Multimedia*, MULTIMEDIA '01, 107–118, ACM, New York, NY, USA. 36

TONG, S. & KOLLER, D. (2002). Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, **2**, 45–66. 36

TUCERYAN, M. & JAIN, A.K. (1998). The handbook of pattern recognition and computer vision (2nd edition). 14

TUR, G., HAKKANI-TÜR, D. & SCHAPIRE, R.E. (2005). Combining active and semi-supervised learning for spoken language understanding. *Speech Communication*, **45**, 171–186. 36

TURNER, M.R. (1986). Texture discrimination by gabor functions. *Biol. Cybern.*, **55**, 71–82. 15

ULUSOY, I. & BISHOP, C.M. (2005). Generative versus discriminative methods for object recognition. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, CVPR '05, 258–265, IEEE Computer Society, Washington, DC, USA. 25

VAN DE SANDE, K., GEVERS, T. & SNOEK, C. (2008). Evaluation of color descriptors for object and scene recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 1 –8. 16, 18, 20, 110, 132

VAN DE WEIJER, J. & GEVERS, T. (2005). Boosting saliency in color image features. In *CVPR*, 365–372, IEEE Computer Society. 18

VAPNIK, V.N. (1998). *Statistical learning theory*. Wiley, 1st edn. 25

VIJAYANARASIMHAN, S. & GRAUMAN, K. (2008). Multi-level active prediction of useful image annotations for recognition. In *NIPS*, 1705–1712. 92

VINOKOUROV, A. & GIROLAMI, M. (2001). Document classification employing the fisher kernel derived from probabilistic hierarchic corpus representations. *Proceedings of ECIR01 23rd European Colloquium on Information Retrieval Research*, 24–40. 21

WAN, X. & KUO, C.C. (1998). A new approach to image retrieval with hierarchical color clustering. *IEEE Transactions on Circuits and Systems for Video Technology*, **8**, 628–643. 14

WANG, F. & MERIALDO, B. (2009). Eurecom at TRECVid 2009 high-level feature extraction. In *TREC2009 notebook*. 55

WANG, H.L., HUANG, J., LIU, Z., WANG, Y., CHEN, Y. & WONG, E.K. (1999). Integration of multimodal features for video scene classification based on hmm. In *In IEEE Workshop on Multimedia Signal Processing*, 53–58. 15, 19

WEISS, G. & PROVOST, F. (2001). The effect of class distribution on classifier learning: An empirical study. Tech. rep. 5, 144

WEISS, G.M. (2004). Mining with rarity: a unifying framework. *SIGKDD Explor. Newsl.*, **6**, 7–19. 45, 46

WEISS, R., DUDA, A. & GIFFORD, D. (1995). Composition and search with a video algebra. *Multimedia, IEEE*, **2**, 12 –25. 11

WEN, Y.M. & LU, B.L. (2007). Incremental learning of support vector machines by classifier combining. In *Proceedings of the 11th Pacific-Asia conference on Advances in knowledge discovery and data mining*, PAKDD'07, 904–911, Springer-Verlag, Berlin, Heidelberg. 84

WINDER, S.A.J. & BROWN, M. (2007). Learning local image descriptors. In *In CVPR*, 1–8. 17

WOLPERT, D.H. (1992). Stacked generalization. *Neural Networks*, **5**, 241–259. 30

WU, C., WANG, X., BAI, D. & ZHANG, H. (2009). Fast incremental learning algorithm of svm on kkt conditions. In *Proceedings of the 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery - Volume 01*, FSKD '09, 551–554, IEEE Computer Society, Washington, DC, USA. 84

WU, Y., CHANG, E.Y., CHANG, K.C.C. & SMITH, J.R. (2004). Optimal multimodal fusion for multimedia data analysis. In *Proceedings of the 12th annual ACM international conference on Multimedia*, MULTIMEDIA '04, 572–579, ACM, New York, NY, USA. 33

WÜRTZ, R.P. & LOURENS, T. (1997). Corner Detection in Color Images by Multiscale Combination of End-Stopped Cortical Cells. In *ICANN '97: Proceedings of the 7th International Conference on Artificial Neural Networks*, 901–906, Springer-Verlag. 17

YAN, R., YANG, J. & HAUPTMANN, A. (2003). Automatically labeling video data using multiclass active learning. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, 516 –523 vol.1. 36

YANG, Y. & NEWSAM, S. (2010). Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '10, 270–279, ACM, New York, NY, USA. 16

YANG, Y.H. & HSU, W.H. (2008). Video search reranking via online ordinal reranking. In *ICME*, 285–288. 55

YILMAZ, E. & ASLAM, J.A. (2006). Estimating average precision with incomplete and imperfect judgments. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, CIKM '06, 102–111, ACM, New York, NY, USA. 39, 105

ZHANG, C. & CHEN, T. (2002). An active learning framework for content-based information retrieval. *IEEE TRANSACTIONS ON MULTIMEDIA*, **4**, 260–268. 36

ZHANG, J., MARSZALEK, M., LAZEBNIK, S. & SCHMID, C. (2006). Local features and kernels for classification of texture and object categories: A comprehensive study. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on*, 13. 28

ZHANG, T. & KUO, C.C.J. (1998). Content-based classification and retrieval of audio. In *IN SPIES 43RD ANNUAL MEETING - CONFERENCE ON ADVANCED SIGNAL PROCESSING ALGORITHMS, ARCHITECTURES, AND IMPLEMENTATIONS VIII*, 432–443. 19

ZHANG, Y., WANG, X. & ZHAI, J. (2009). A fast support vector machine classification algorithm based on karush-kuhn-tucker conditions. In *Proceedings of the 12th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*, RSFDGrC '09, 382–389, Springer-Verlag, Berlin, Heidelberg. 84

ZHU, J., HOVY, E. & REY, M. (2007). Active learning for word sense disambiguation with methods for addressing the class imbalance problem. *Computational Linguistics*, 783–790. 37