Hierarchical late fusion for concept detection in videos

Sabin Tiberius Strat, Alexandre Benoit, Patrick Lambert, Hervé Bredin and Georges Quénot

Abstract Current research shows that the detection of semantic concepts (animal, bus, person, dancing etc.) in multimedia documents such as videos, requires the use of several types of complementary descriptors in order to achieve good results. In this work, we explore strategies for combining dozens of complementary content descriptors (or "experts") in an efficient way, through the use of late fusion approaches, for concept detection in multimedia documents. We explore two fusion approaches that share a common structure: both start with a clustering of experts stage, continue with an intra-cluster fusion and finish with an inter-cluster fusion, and we also experiment with other state-of-the-art methods. The first fusion approach relies on a priori knowledge about the internals of each expert to group the set of available experts by similarity. The second approach automatically obtains measures on the similarity of experts from their output to group the experts using agglomerative clustering, and then combines the results of this fusion with those from other methods. In the end, we show that an additional performance boost can be obtained by also considering the context of multimedia elements.

Keywords: late fusion, hierarchical, AdaBoost, semantic concepts, video, semantic indexing

Sabin Tiberius Strat
LISTIC - University of Savoie, Annecy, France; LAPI - University "POLITEHNICA" of Bucharest, Romania, e-mail: Sabin-Tiberius.Strat@univ-savoie.fr

Alexandre Benoit
LISTIC, e-mail: Alexandre.Benoit@univ-savoie.fr

Patrick Lambert
LISTIC, e-mail: Patrick.Lambert@univ-savoie.fr

Hervé Bredin
CNRS-LIMSI, Orsay, France, e-mail: Herve.Bredin@limsi.fr

Georges Quénot
UJF-Grenoble 1 / UPMF-Grenoble 2 / Grenoble INP / CNRS, LIG UMR 5217, Grenoble, F-38041, France, e-mail: Georges.Quenot@imag.fr

1 Introduction

During the last years, society has witnessed a great increase in the amount of multimedia information, in the form of image, audio and video documents. This has led to an increase in demand for solutions aimed at automatically analyzing and organizing this content, in order to give the users the possibility to retrieve particular multimedia elements by browsing and searching the database. Formulating searches in humanly-understandable *concepts* requires that the database be indexed according to such terms, which creates the need for *automatic semantic indexing* tools.

Many advances have taken place in recent years on the topic of concept detection in multimedia collections with the goal of semantic indexing and there are several well-known, publicly-available datasets on which researchers can test and compare their different algorithms. For example, the Pascal VOC (visual object categories) challenge focuses on detecting objects in static images [12], the MediaEval series of benchmarks is dedicated to evaluating algorithms for multimedia access and retrieval in videos accompanied by metadata, therefore focusing even on human and social aspects of multimedia tasks [19], while the TRECVid series of workshops proposes several video-only analysis tasks, such as semantic indexing and surveillance event detection [24].

A basic framework for semantic indexing on a multimedia dataset consists in extracting content descriptors from the samples (e.g. images or video shots), then training supervised classifiers on each of these descriptors. This produces, for each available descriptor and for each associated classification method, a set of classification scores that describe the “likeliness” of each sample to contain a given target concept. When possible, such scores can be calibrated as probabilities for the samples to contain the target concept.

We call an *expert* any method able to produce a set of likeliness scores for multimedia samples to contain a given target concept. Such scores can then be used to produce a ranked list of the samples the most likely to contain this concept. A combination of a content descriptor and a supervised classification method constitute an *elementary expert*. These steps are represented by the “Descriptor computation” and “Supervised classification” blocks in Fig. 1 (this figure illustrates the entire processing chain that we use in our experiments, which will be explained in more detail later on).

As several content descriptors and several supervised classification methods can be considered, many elementary experts can be built. So far, information coming from different elementary experts is not jointly exploited, as experts are treated independently. However, different types of elementary experts, each based on different aspects of the multimedia samples (such as colors, textures, contour orientations, motion or sounds etc.), give *complementary* information.

Several aspects of complementarity can be discussed. The first is *inter-concept complementarity*, which means that a certain expert (based on a certain type of content descriptor) can give very good results for a particular semantic concept, yet perform poorly for another concept. For example, on the TRECVid SIN video dataset,

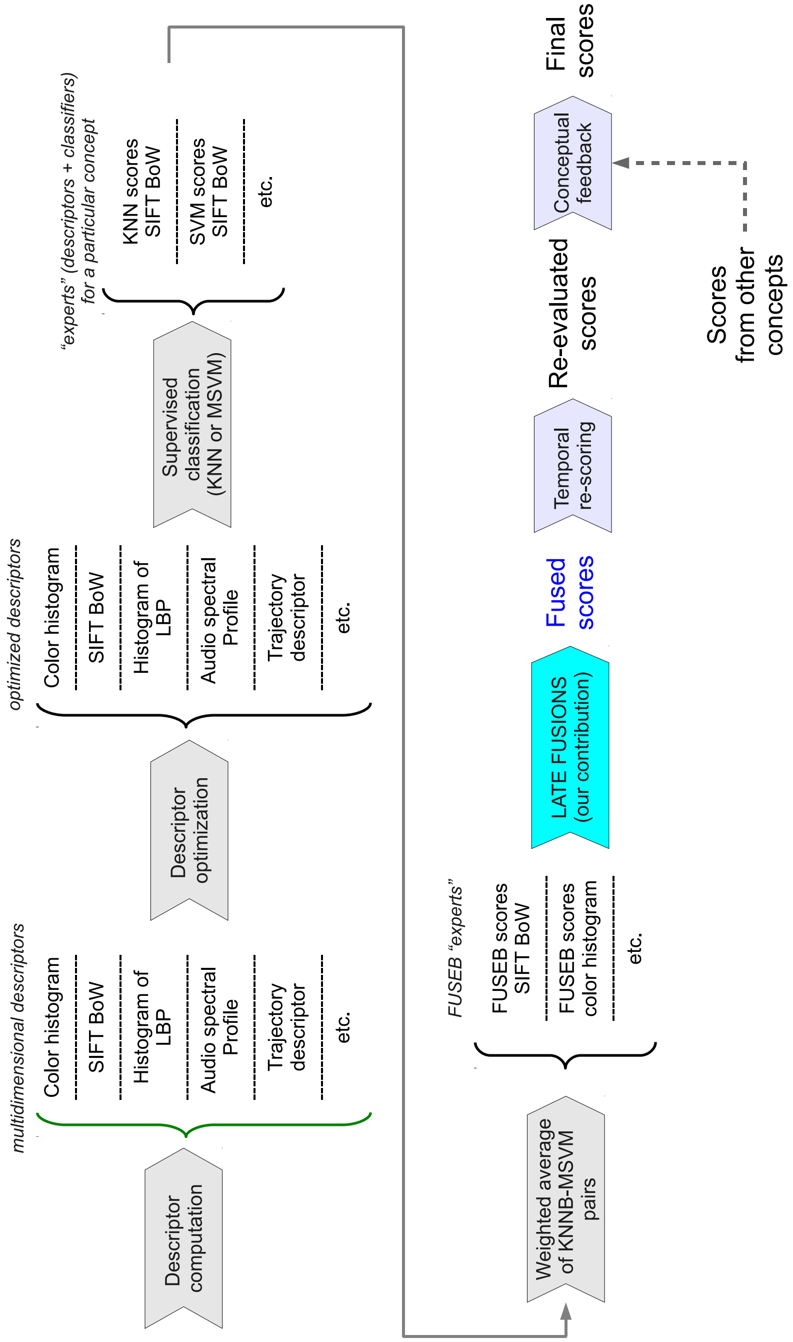

Fig. 1 The semantic indexing processing chain used by the IRIM group [4], in which our contribution (late fusion approaches) is integrated.

the concept *"Football"* is better detected by experts using trajectory descriptors than by those using SIFT Bag-of-Words descriptors, or vice-versa, the concept *"Bridges"* is better detected with SIFT Bag-of-Words than with trajectories. There is no single expert which is systematically the best for all target concepts.

The second aspect of complementarity is *intra-concept complementarity*, which means that even if two (or more) experts have modest performances for a particular concept, their combination can produce a *higher level expert* that often performs better than any of its input elementary experts. This is especially true when one of the elementary experts detects the concept better in some situations (corresponding to some of the multimedia samples where the concept is present), while the other expert works better in the rest of the situations (the rest of the samples where the concept is present), which means that there is *complementarity at the context level*.

Because of these observations, for the sake of universality and in order to exploit complementary information, many systems rely on the combination of a large set of experts (up to 100+), each based on different descriptors or descriptor versions, and using various supervised classification algorithms.

The work described here focuses on the next step in the semantic indexing pipeline, immediately following the (multiple) supervised classification: the combination by *late fusion* of a large battery of complementary experts. The goal is to exploit their complementarity as well as possible for boosting the concept detection performance as far as possible.

The rest of the chapter is structured as follows: section 2 reviews the relevant state of the art; section 3 explains the motivation of the presented work; section 4 describes the proposed approaches; section 5 describes some additional improvements to the proposed approaches; section 6 presents the experiments carried out and the obtained results; and section 7 draws some conclusions and gives some perspectives.

2 State of the art

Semantic concept detection in multimedia elements starts with computing descriptors. In the case of video datasets, we can have many types of descriptors, such as Bags-of-Words of local features (SIFT [20], SURF [5] or other type), color histograms, trajectories [2] or audio descriptors, with more examples given in Sec. 6.2. On such a descriptor, for a particular target concept, a supervised classification algorithm is trained and applied (such as K-nearest neighbours, support vector machines (SVM) with various kernels, artificial neural networks, gaussian mixture models etc.), obtaining an elementary expert [4].

Most often, combining information from several experts improves the correct recognition rates of semantic concepts. Experts can be combined at several stages within the processing chain: *Early fusions* combine descriptors before the classification step, while *late fusions* combine the outputs of supervised classifiers.

Early fusions can be as simple as concatenating two or more multidimensional descriptors, but for better results, the fact that descriptor dimensions may have values in different ranges, that descriptors may have varying numbers of dimensions and that descriptors may have varying importances for a certain concept needs to be taken into account. In [48], early fusion is performed by computing the distance between two videos as a weighted average of distances between different descriptors. In [44], a multi-channel approach is used to combine a trajectory descriptor (movements from one frame to the next) and trajectory-aligned descriptors (histograms of oriented gradients, histograms of optical flow, motion boundary histograms) as input for a SVM with a χ^2 kernel, by measuring the distance between videos as the average of distances between channels (input descriptors).

Late fusions can be as simple as averaging the output scores from classifiers based on different descriptors (averaging different experts), or can be more complex, taking into account the inter-dependencies of scores from different experts like it is done with Choquet's integral [10]. An additional level of supervised classification can also be trained on the set of experts, however this can lead to over-fitting which degrades results, and averaging output scores generally gives results just as good (or better) with less computational cost. In [48], late fusion is done by averaging output scores from different experts, but in their approach, early fusion performed better than late fusion. They also experimented with a combination of early and late fusion (double fusion) which was shown to generally outperform both the early and late fusion. In general, late fusions perform best when the experts being fused are complementary, as it was shown by [23].

In [50], a visual classifier and two textual classifiers are combined using methods from belief theory, in the context of image classification. Classifier output probabilities are first converted into consonant mass functions, and then these mass functions are combined in the belief theory using Dempster's rule [36] or the Average rule. Both rules gave significantly better results than classifiers taken independently, with Dempster's rule performing better for challenging classes.

There can also be intermediates between early fusions and late fusions. With regard to SVM classifiers, Multiple Kernel Learning (MKL) can be considered a sort of intermediate fusion. Instead of using a single kernel function for the SVM, several kernels can be combined (either working on the same data or on different data) to improve classification results [14]. For example, the multi-channel approach in [44] can be regarded as a MKL problem.

In [27], an early fusion, an intermediate fusion and three late fusions are used to combine static, dynamic and audio features for activity recognition using hierarchical hidden Markov models. The early fusion is a concatenation of descriptors, while the late fusions combine confidence scores from separate classifiers. The intermediate fusion, which gives the best results in their context, considers each modality as a stream of measurements and each state of the HMM models separately the observations of each stream by a Gaussian mixture, each stream being weighted depending on the activity in question.

Fusion strategies for detecting a concept can also concern themselves with how to deal with data imbalance problems (such as in TRECVid Semantic Indexing task, where most of the concepts have many more negative labeled examples than positive ones) or which features or descriptors are more relevant for that concept. In [48], a Sequential Boosting SVM inspired from bagging and boosting approaches is used. Bagging [7] means splitting the training database into several subparts (when there are many more training negatives than positives, the positives may be kept common to all subparts) and training a classifier on each subpart; at recognition, the outputs from those classifiers are combined (averaged) to improve the result. Boosting strategies such as AdaBoost [13, 34] train a strong classifier by combining (through weighted average) results from many weak classifiers. In TRECVid, late fusions based on AdaBoost have been used in [8, 45, 43].

In the context of the TRECVid Semantic Indexing (SIN) task and as part of our participation with the IRIM group, we opt for the use of late fusion approaches (in a concept-per-concept manner), because an early fusion would mean training supervised classifiers on very high-dimensional descriptors, which is not trivial. Late fusions are easier to apply, because they fuse simple classification scores, not complex multidimensional descriptors, and in the case of TRECVid SIN, it was shown in [4] that late fusions also give better results. As inputs for the late fusion, we have a battery of (50+) *experts*, which are classification scores for each of the multidimensional descriptors (and their versions), on each video shot and each concept. A similar fusion context is described in [9], where experts are generated from a large number of video descriptors on which different classification algorithms are applied, the classifier that yields the best result for each descriptor is retained and the resulting experts are combined in a late fusion approach.

3 Choice of late fusion strategy

When looking for an effective combination of experts, several interrogations arise. Should we use them all in the fusion process, or just the best ones? Does combining two experts always yield better results than the two of them taken separately? Should we weigh them differently in case one is much better than the other? Tackling a similar problem, *Ng and Kantor* [23] proposed a method to predict the effectiveness of their fusion approach and concluded:

> *Schemes with dissimilar outputs but comparable performance are more likely to give rise to effective naive data fusion.*

where the *similarity* between two experts *outputs* can be measured as the Spearman rank correlation coefficient [17] – and *naive data fusion* should be understood as fusion by sum of normalized scores.

3.1 Fusion of two experts

In order to validate the conclusion of *Ng and Kantor* [23] in the case of concept detection in videos, we drove a simple experiment whose outcome is summarized in Fig. 2.

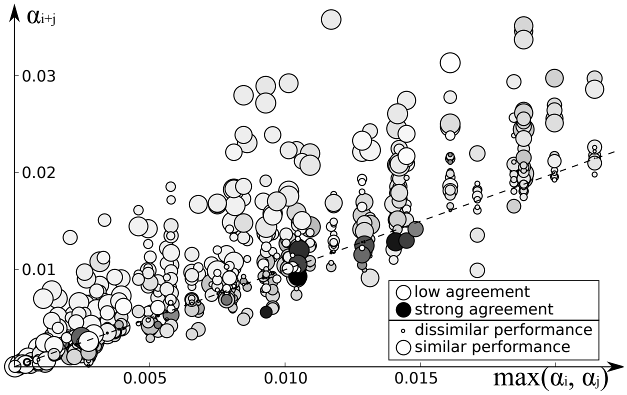

Fig. 2 Average precision gains when combining experts that have various performances and various agreement rates. Each circle represents an expert pair. The x-axis corresponds to $\max(\alpha_i, \alpha_j)$, the average precision of the best expert from a pair. The y-axis indicates α_{i+j}, the average precision of the combination of a pair. Dark (resp. bright) grey circles indicate that experts i and j strongly agree (resp. disagree) in their rankings. The circle diameter is directly proportional to the ratio of the average precisions in the pair α_i/α_j (where $\alpha_i < \alpha_j$).

Given a set of $K = 50$ experts trained for the detection of a given concept, and an estimation of their performance (average precision) α_k on the *TRECVid 2010 Semantic Indexing* task [24], we considered all pairs (i, j) of experts and evaluated the performance of their fusion by weighted sum of normalized scores:

$$\mathbf{x} = \alpha_i \cdot \mathbf{x}_i + \alpha_j \cdot \mathbf{x}_j \qquad (1)$$

As most circles are above the $x = y$ line (i.e. $\alpha_{i+j} > \max(\alpha_i, \alpha_j)$), Fig. 2 clearly shows that the weighted sum fusion from Eq. 1 is the most beneficial for experts that tend to disagree on their rankings but have similar average precisions (bright, large circles). This means that the gain is maximum when we have *intra-concept complementarity, at the context level*, as discussed in Sec. 1.

3.2 Communities of experts

We have given an example for two experts, however, as described in Sec. 6.2, the final objective is to combine a large collection of (50+) experts. The difference between those experts mostly comes from the type of descriptors they rely on, and partly from the type of classifiers trained on top of these descriptors.

We expect experts relying on similar descriptors to generate similar outputs and therefore strongly agree with each other. We ran an additional set of preliminary experiments in order to verify this hypothesis – as illustrated in Fig. 3.

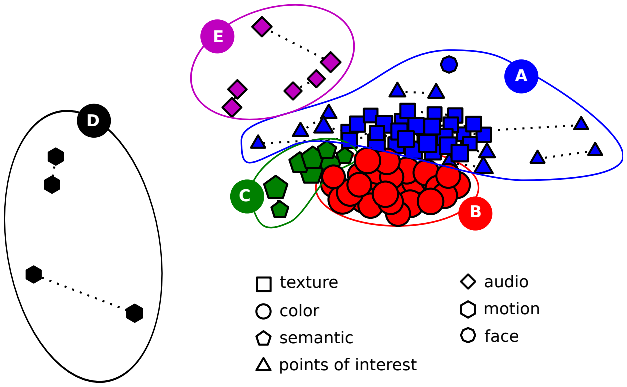

Fig. 3 Similarity of experts trained for the detection of concept *Computers*. Each node represents an expert, and edges represent the similarity between them (we only display some of the edges). The dotted edges represent experts which derive from the same descriptors, but use different classifiers.

In Fig. 3, each expert is represented by a node and similar experts (according to their Spearman rank correlation coefficient [17]) are positioned closer to each other using a standard spring-layout algorithm. It appears that some kind of community structure naturally emerges, with several groups of experts being more strongly connected internally than with the outside of their group.

This is partly due to the type of descriptors used internally by the experts (denoted by the shape of the nodes). For instance, experts based on color descriptors (circles) seem to agglutinate, as do experts based on audio descriptors (diamonds). Finally, the size of a node is directly proportional to the performance (average precision) of the corresponding expert. Therefore, best performing experts (i.e. larger nodes) also tend to agglutinate as they provide rankings that are closer to the reality – therefore closer to each other.

We also used the so-called *Louvain* algorithm to automatically detect communities of experts in this graph [22, 6]. With no objective groundtruth to compare with, it is difficult to evaluate the detected communities. However, looking at Fig. 3 and the five detected communities (A to E), it seems that the *Louvain* algorithm did a good job at finding communities related to the type of descriptors on which experts are based. In particular, a dotted edge between a pair of experts indicates that they are based on the very same descriptors and they only differ in the classifier they rely on. None of these pairs is split into two different communities.

3.3 Hierarchical fusion of multiple experts

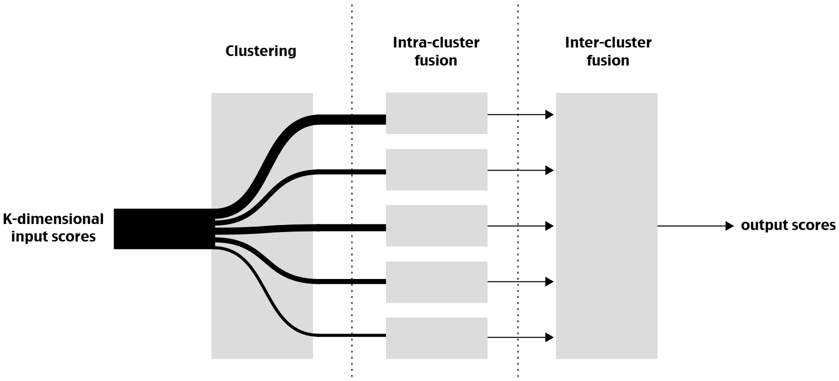

Fig. 4 Basic principle of our main fusion approaches: K input experts are available, which are clustered based on similarity into several groups, followed by an intra-cluster fusion and an inter-cluster fusion. Figure from [42].

Based on the effects noted in Sec. 3.1, and as illustrated in Fig. 4, the late fusion approaches that we propose share the following general framework:

- First, experts are grouped based on similarity into clusters of similar experts. This grouping can either be done manually, using external knowledge about the internal workings of each expert (e.g. grouping all experts that use color descriptors), or automatically, as it was done in Sec. 3.2.
- Then, intra-cluster fusions are performed, in which the experts from each cluster are fused. This balances the quantity of experts of each type, avoiding the case when numerous similar experts dominate the others (because some groups may be very numerous, while other groups may only have a few or even a single expert), and also helps to reduce classification "noise" within the group.
- Last, an inter-cluster fusion is performed, in which the different clusters (which are complementary because they contain experts of different types) are fused together. This gives the main performance boost due to complementarity, based on the remark of *Ng and Kantor* [23] and on our preliminary tests from Sec. 3.1.

4 Proposed approaches

Our goal is to combine information coming from different experts in a way close to the optimum, so that the gain from complementarity is maximized. Following their successful use in our previous work [42], we propose two approaches: one that relies on manually grouping experts, and the other that determines the group and the weight of each expert automatically. Our main fusion approaches are the following:

- *Manual hierarchical fusion*: Expert groups are chosen manually, in a hierarchical manner, based on how the expert was obtained. There are several fusion levels, corresponding to the levels of the expert hierarchy.
- *Agglomerative clustering*: This is our automatic approach; experts are fused progressively based on similarity into groups, followed by inter-group fusion. We also extend this approach compared to what was done in [42].

4.1 Manual hierarchical fusion

The manual hierarchy was designed according to a high-level knowledge about the descriptors and the classifiers. The main principle considered is to fuse first descriptors or classifiers that are expected to be closer considering their nature or principle of operation. The manual hierarchy incorporates more levels than the automatic ones, with branches with different depths. In practice, we fused first the output of all the available machine learning algorithms for each descriptor (e.g. kNN and SVM, corresponding to block *"Weighted average of KNNB-MSVM pairs"* in Fig. 1). We then fuse different variants of the same descriptor (e.g. BoW of the same local descriptor but with different dictionary sizes). Afterwards, we fuse the experts corresponding to different image spatial decompositions (pyramid) if available. Finally, the last level concerns descriptors of different types within the same modality (e.g. color, texture, interest points, percepts or faces) and descriptors from different modalities (audio and visual).

Various experiments with manually defined hierarchies suggested that going from the most similar to the most different was a good strategy. These experiments also showed that the best results are obtained when using as many combinations as possible of descriptors and machine learning algorithms. Even combinations with low performance can contribute to a global performance increase, especially if they are complementary to better ones.

Late fusion was performed at all levels using a weighted arithmetic mean of normalized scores. Several other and more complex methods were tried but produced no or very small improvements. Three weighting strategies were considered: uniform (simple arithmetic mean), MAP based (simple function of the Mean Average Precision of the different inputs), and direct optimization by cross-validation. Cross-validation experiments showed that in the early stages, uniform weighting was preferable for robustness while in latter stages MAP-based or directly optimized weighting provided better results.

4.2 Agglomerative clustering and extensions

The original version of this approach from [42] is based on grouping and fusing experts progressively based on similarity, until a minimum similarity threshold is reached; it clusters experts into groups and performs intra-group fusion at the same time. Because of this functioning, we call this fusion method *agglomerative clustering*. After this step, inter-group fusion is performed to obtain the fused result.

Compared to what was done in [42], we extend this agglomerative clustering approach by also performing, in parallel, four additional fusions: two versions of AdaBoost fusions inspired from [8, 45, 43], one weighted arithmetic mean of experts, and the best expert for each concept. At the end, the results of the five fusions are combined by choosing, for each semantic concept, the fusion method among the five that gave the best result for that concept on the training set.

We will first present the original approach, utilizing only agglomerative clustering, and then we will detail the other fusions with which we compare and also extend the agglomerative clustering.

4.2.1 Agglomerative clustering of experts

The agglomerative clustering fusion method treats each semantic concept independently, and for each concept, applies the following steps:

1. *Relevance of experts estimation*: The relevance of each of the input elementary experts is estimated on the training set, for the concept in question. The relevance is measured as the average precision of the expert normalized with respect to chance (the result of randomly choosing samples). An expert with a relevance of 1 means that it performs just as poorly as chance.
2. *Selection of experts*: Experts with a relevance less than 1 are thrown away, because they are irrelevant to the concept in question. Experts with a relevance 8 times smaller than that of the best are also thrown away, in order not to “pollute” the best expert with others that are much worse. This second selection is not critical, neither is its threshold, but using it tends to reduce performance degradation from fusion for the (very few) concepts that have an extremely good best expert.
3. *Iterative fusion*: Some of the retained experts are highly correlated, so we look for the pair of experts *with the maximum correlation* and fuse it into a single expert (through arithmetic mean). The correlation between the resulting expert and the remaining ones is updated, and the process is repeated. The iterative fusion stops when a sufficiently correlated pair of experts can no longer be found. The iterative fusion corresponds to the first 2 steps in Fig. 4, as it groups and fuses similar

experts at the same time (progressively, as pairs of highly-correlated experts are found).

4. *Weighted arithmetic mean*: The iterative fusion does not give a large gain, because it only groups and fuses *similar* experts. The main performance boost comes now, when we fuse *different* groups via a weighted mean of experts. The weights are given by the average precisions (for the current concept on the training dataset) of the experts from the previous step. A single expert is obtained, the result of our agglomerative clustering fusion approach. This weighted arithmetic mean corresponds to the last step in Fig. 4.

The correlation measure used in the iterative fusion step is the Pearson product-moment correlation coefficient ρ of the raw classification scores. $\rho \in [-1;1]$, with values in the range of 0.6-1 corresponding to high correlation. In order to fuse a pair of experts, not only does the correlation coefficient for the classification scores of *all* samples need to be at least 0.75 (the two experts give similar information on a global scale), but also the correlation coefficient for the scores of *only the positive* samples must be at least 0.65 (to ensure that the two experts tend to detect more or less the same true positives of the semantic concept being analyzed). The constraint related to positives was added again with regards to the remark of *Ng and Kantor*, as at this stage, we want to group similar (not very complementary) experts; also, without this constraint, because of the imbalance between positives and negatives, the scores for negatives would have dominated the correlation measure.

The goal of iterative fusion is to balance the contribution of each family of experts, as we will see in Sec. 6.2 that some families are very numerous, while other families are small. This method is automatic and avoids needing to specify the families manually, making it practical for often-changing expert sets and for automatically grouping experts of similar types but from different contributors. The groups formed by the iterative fusion correspond in a large degree to the expectations based on descriptor type.

In addition to the agglomerative clustering fusion, we also experiment with other fusion approaches and with combining the results from these different fusion approaches, as described in the following.

4.2.2 AdaBoost score-based fusion

AdaBoost [13], short for “adaptive boosting”, is an algorithm that constructs a strong expert through a weighted average of a large number of weak experts. AdaBoost functions properly when each of the weak experts is at least slightly better than chance, and when the different involved experts are complementary (they each correctly classify different parts of the dataset). This is very much the case of TRECVid, where we have a large battery of experts, most of them not having spectacular individual performance (but better than chance), organized into complementary families.

The AdaBoost algorithm that we use is inspired from the original one in [13] with adaptations for TRECVid. It is very similar to that of [45], however they applied it in a different context of TRECVid. It is also very similar to that used by [43] in the 2008 edition of TRECVid, but they did not use it on such a large battery of experts as we do in our experiments.

For a particular concept, given the training set $(x_1,y_1),\ldots,(x_m,y_m)$ where x_i are the multimedia samples, and $y_i \in \{0,1\}$ is the groundtruth of the sample x_i (0 if it does not contain the concept, 1 if it does), the algorithm that we use is the following:

1. We initialize a set of weights D_1 where $D_1(i)$ is the weight of sample x_i:

$$D_1(i) = \begin{cases} \frac{0.5}{nPos}, & \text{if } y_i = 1 \text{ (a positive sample)} \\ \frac{0.5}{nNeg}, & \text{if } y_i = 0 \text{ (a negative sample)} \end{cases} \tag{2}$$

   where *nPos* and *nNeg* are the number of positive and negative samples respectively in the training set.
2. At iteration t ($t = 1,\ldots T$), we choose the input expert h_t that minimizes the weighted classification error $\varepsilon_t = \sum_{i=1}^{m} D_t(i)I(y_i \neq h_t(x_i))$. I is called the indicator function, and it gives the cost associated to the classification result of a sample being different than the groundtruth. In our case, $I(y_i \neq h_t(x_i)) = |y_i - h_t(x_i)|$, the absolute value of the difference between the classification score (between 0 and 1) and the groundtruth (0 *or* 1).
3. Compute the weight updating factor $\alpha_t = ln\frac{1-\varepsilon_t}{\varepsilon_t}$;
4. Update the weights of the samples according to:

$$D_{t+1}(i) = D_t(i)exp(\alpha_t I(y_i \neq h_t(x_i))) \tag{3}$$

   and normalize the weights for positive samples and for negative samples separately, so that $\sum_{i,y_i=1} = 0.5$ and $\sum_{i,y_i=0} = 0.5$ (always keep the total weight of positives and the total weight of negatives equal).
5. Repeat steps 2-4 until all input experts have been considered (each expert is only considered once).
6. At the end, the *strong expert* $H(x)$ will be a weighted sum of the weak experts chosen at each iteration t:

$$H(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \tag{4}$$

AdaBoost works on the following principle: at each step, we select the expert that correctly classifies the multimedia samples for which the previous expert failed, this way achieving *intra-concept complementarity at the context level*. Unlike agglomerative clustering, it does not first group experts into families and then obtain complementarity between families; instead, AdaBoost tries to exploit complementarity directly by choosing, at each step, the most complementary expert.

For datasets with severe class imbalance (as is the case of the TRECVid SIN video dataset, in which, for many concepts, there are only a few tens of positives and hundreds of thousands of negatives), we have added the additional constraint that the total weight of positives and the total weight of negatives should have fixed values on 0.5 each, at every iteration, as in [45], so that the classification result for true positives would still matter in the fusion.

Also for the case of TRECVid, we performed a similar expert preselection as for the agglomerative clustering fusion: we rejected experts with relevances less than 1 or less than 8 times that of the best expert for that concept, for similar reasons as in the case of the agglomerative clustering.

4.2.3 AdaBoost rank-based fusion

When quering a dataset for a particular concept, we receive a ranked list of multimedia samples, in descending order of their likelihood to contain the concept. Ideally, in this ranked list, all the true positives should be concentrated towards the beginning, and all the negatives should follow until the end of the list. The previous AdaBoost method was made to improve the classification scores, which would indirectly improve the ranked list. We now try to optimize directly the ranks of the true positives, by altering the indicator function (the cost function when a classification error appears).

We therefore propose the following indicator function: for a positive sample, the associated cost is equal to the number of negatives that are in front of it in the ranked list, divided by the total number of negatives; for a negative sample, the cost is zero (we don't care about its rank, as long as the positives are in front):

$$I(y_i \neq h_t(x_i)) = \begin{cases} \frac{negPreceeding}{nNeg}, & \text{if } y_i = 1 \text{ (a positive sample)} \\ 0, & \text{if } y_i = 0 \text{ (a negative sample)} \end{cases} \tag{5}$$

where *negPreceeding* is the number of negatives preceeding the positive sample in question in the ranked list, according to the weak expert h_t, and *nNeg* is the total number of negatives.

As with the agglomerative clustering fusion and the AdaBoost fusion based on scores, we perform similar expert selections before starting the actual fusion.

4.2.4 Weighted average of experts

As a reference for comparing the performances of the fusion methods presented so far, we consider the weighted average of the input experts, with weights given by the average precisions of experts on the training set, for the concept in question (the weights can vary from one concept to another, depending on how the experts react to the concepts). We can say that in the end, the other methods are also weighted means of experts, but with more elaborate ways of choosing the weights. We wish to compare the more elaborate methods with this simple baseline.

As with the other fusion methods presented so far, we perform similar expert selections before starting the actual fusion.

4.2.5 Best expert per concept

We add a second reference for evaluating the performance of our fusion methods, namely the best expert per concept. This method consists in simply choosing, for each semantic concept individually, the expert that gives the best average precision on the training set. This is our most basic reference when examining other methods, as the goal of fusions is to obtain gains compared to simply considering the best expert for the concept of interest.

4.2.6 Combining fusions

After applying all of the previous approaches in parallel, we now dispose of a battery of five fused experts: agglomerative clustering, score-based AdaBoost, rank-based AdaBoost, weighted average and best expert per concept. Our preliminary experiments have shown that for some concepts, some (or all) of the fusion methods degrade performance on the training set when compared to simply choosing that concept's best expert. To prevent this, we propose that for each concept, we see which of the fusion methods (including the best expert per concept) performs best on the training set, and *choose* that fusion method as the final result for that concept.

5 Improvements: higher-level fusions

After the late fusion step, we dispose, for each concept, of the classification scores on all video shots. So far, we have treated each concept independently, disregarding any relationship that may exist between concepts. Moreover, the video shots from TRECVid result from the temporal segmentation of longer videos, therefore there may also exist temporal relations between shots. The next step is to integrate this *temporal context* and *semantic context* information.

A concept that is present in a shot of a video also tends to be present in the neighboring shots of the same video due to temporal correlation. We exploit this *temporal context information* by applying the method from [30] to *temporally re-score* shots, which was shown to increase performance in this application context [30] (block '*'Temporal re-scoring"* in Fig. 1).

After temporal re-scoring, we exploit the *semantic context information* by applying *conceptual feedback* on the classification scores with the algorithm from [16]. This exploits the semantic relations between concepts by constructing a new descriptor with 346 dimensions (exactly the number of concepts), the i^{th} dimension of this descriptor being the classification score of the shot with the i^{th} concept. Supervised classification is applied on this descriptor as if it were a normal descriptor, and the resulting classification scores are re-fused with the previous results (block *"Conceptual feedback"* in Fig. 1). This step was also shown to increase performance in our application context [16].

6 Experiments

6.1 The TRECVid Semantic Indexing task

The work presented here has been carried out and evaluated in the context of the Semantic Indexing Task (SIN) of the TRECVid evaluation campaign. The 2013 dataset associated with this task is composed of $\approx$ 1400 hours of web video data decomposed into $\approx$ 35,000 video documents and $\approx$ 880,000 *shots*. Shots are short video fragments of lengths varying between a few seconds to a few tens of seconds; they generally correspond to continuous camera recordings and are expected to have a homogeneous content and they constitute natural indexing and retrieval units.

A list of 346 various concepts is also provided. These can be objects (*Bus, Tree, Car, Telephone, Chair*), actions (*Singing, Eating, Handshaking*), situations/scene types (*Waterscape, Indoor, Kitchen, Construction site*), abstract concepts (*Science/-technology*), types of people (*Corporate leader, Female person, Asian people, Government leader*) or even specific people (*Hu Jintao, Donald Rumsfeld*). These concepts may or may not be present in a shot. Semantic indexing, as defined in TRECVid, consists in automatically detecting the presence of these visual concepts in video shots [37].

The dataset is split in two parts, the first one (dev or 2013d), for developing and fine-tuning semantic indexing systems, and the second one (test or 2013t) for evaluating the performances of the task participants. On the test part of the dataset, semantic indexing systems are required to produce, for each target concept, a ranked list of up to 2000 shots the most likely to contain it. The quality of the returned lists (how well the relevant shots for that concept are concentrated towards the beginning of the list) is evaluated using the *mean inferred average precision (mean infAP)* [46, 47]. Common annotations are given on the dev part for system training and assessments are provided on the test part for system evaluation.

The TRECVid SIN dataset is very challenging, for the following reasons:

- Videos come from a wide array of sources, of varying quality and content, ranging from professional news footage to amateur videos recorded with a camera phone. They can be from various environments, such as from inside a kitchen or from outside in the street or at the beach. They can be acquired in various lighting conditions, ranging from a sunny day outdoors to a dark interior of a night club.
- The large amount of concepts to be detected requires a generic approach to be used for all concepts. However, it is not easy to develop a generic system that works well-enough with every concept.
- Many concepts are quite rare in the dataset; they may only appear in a few tens of shots out of the total $\approx$ 880,000, which poses a problem for training classifiers.
- For a shot to be considered as an occurrence of a concept, it is enough that the concept is present in at least one frame of the shot. However, the training annotation only says if a shot contains or does not contain a concept, but it does not say *when and where* that concept appears. This poses a challenge because we do not know which part of the shot is relevant and needs to be described.

We have chosen to perform our experiments on this dataset because it is so challenging (for example, the peak performances in the 2012 edition were in the order of 0.3 mean infAP [38], far from the ideal value of 1) and because, as we have participated in the task as a member of the IRIM[1] group, we have had access to a large battery of multimodal video descriptors (and corresponding experts) on which we could experiment with information fusion approaches, which is the topic of this work.

6.2 Elementary experts

Recalling the processing chain from Fig 1, the first step for semantic indexing is to extract descriptors from the video shots. For its participation in the TRECVid challenge, the laboratories that form the IRIM group have all shared their descriptors, creating a very rich and multimodal representation of the video shots. The IRIM partners have contributed many descriptors and descriptor versions, and a full listing of them is beyond the scope of this work. Instead, we will just list some of the main descriptors, without going into details:

- A large family of color descriptors was submitted by ETIS, with color represented in the Lab color space, with an optional spatial division of the keyframe [15]. A color histogram in the RGB color space was also submitted by LIG.
- ETIS also contributed quaternionic wavelets, which are a texture descriptor, also with an optional spatial division of the keyframe [15].
- A normalized Gabor transform of the keyframe was contributed by LIG, as well as an early fusion of their RGB color histogram and this normalized Gabor transform.
- BoW descriptors based on Local Binary Patterns were contributed by LIRIS [49], and texture local edge patterns enhanced by color histograms [49] were contributed by CEALIST. Multi-level histograms of multi-scale LBP with spatial pyramids were contributed by LSIS [26].
- BoW of Opponent SIFT features: contributed by LIG in versions with keypoints either from a Harris-Laplace corner detector, or from a dense grid [33]. From the same family, CEALIST contributed BoW of dense SIFT with spatial pyramids [35, 3] and LISTIC contributed BoW of dense SIFT employing retinal preprocessing [40, 41, 39].
- Vectors of locally-aggregated tensors (VLAT) [21], which also deal with local SIFT features clustered on a visual vocabulary, but use a pooling mechanism different than BoW to generate image signatures, were submitted by ETIS.
- Saliency moments, a descriptor that exploits the shape and contours of salient regions [28], was submitted by EUR.
- BoW of space-time interest points, described with histograms of oriented gradients or with histograms of optical flow, as in [18], were submitted by LIG.

[1] `http://mrim.imag.fr/irim/`

- EURECOM submitted spatio-temporal edge histograms, based on temporal statistics of the (2D) MPEG-7 edge histogram.
- Descriptors based on tracking and describing faces in successive frames (face tracks) were submitted by LABRI.
- LISTIC submitted Bags of Words of trajectories for motion description.
- Audio descriptors in the form of a BoW of Mel-frequency cepstral coefficients (MFCC) were contributed by LIRIS.
- Detection scores of various semantic concepts from the ILSVC and ImageNet datasets [11] (with detectors trained on ImageNet) were submitted by XEROX [32]. From the same family of highly-semantic descriptors, LIF contributed a descriptor based on detection scores for a set of 15 mid-level concepts called "percepts" [1].

Before supervised classification, most of the descriptors went through an optimisation consisting in applying a power transformation to normalize the values of the descriptor dimensions, followed by Principal Component Analysis (PCA) to make each descriptor more compact, and at the same time, more robust [31], corresponding to the *"Descriptor optimization"* block in Fig. 1.

The next step was to train and apply supervised classification algorithms (classifiers) on each of the (optimized) descriptors (*"Supervised classification"* in Fig. 1). A classifier gives, for each concept and for each video shot, the estimated "likeliness" of the shot to contain the concept (a classification score between 0 and 1).

Two classifiers were applied to each video shot descriptor. The first one is based on a K-Nearest Neighbours search [2]. The second one, called MSVM, applies a multiple learner approach based on Support Vector Machines [29]. MSVM generally performs better than KNN, but it is more computationally expensive [4].

KNN and MSVM classifiers applied to a given descriptor constitute two different elementary experts. These can be combined (or fused) into a first level non-elementary expert. The combination can be done in a number of ways. For this first level, we use a weighted mean of classification scores, the weights between KNN and MSVM being their infAP performance estimated by cross-validation within the training (dev) set. The corresponding expert is called FUSEB; it is most often better than either KNN or MSVM. We later *use the FUSEB experts as elementary ones* for the next steps in our proposed late fusion approaches.

The most numerous family of FUSEB experts is that of ETIS color histograms in the Lab color space (12 experts), while their quaternionic wavelets family numbered 9 experts. LISTIC had in total 11 SIFT-based BoW experts, some with and some without retinal preprocessing, and for 5 experts using trajectories. 6 OpponentSIFT BoW experts from LIG were also used, as well as two more dense SIFT experts from CEALIST. There were 5 experts based on percepts, while the experts corresponding to the remaining descriptors from the previous list were less numerous (only one or two).

[2] `http://mrim.imag.fr/georges.quenot/freesoft/knnlsb/index.html`

6.3 Results

All of the compared fusion methods are tested using the same input elementary experts, the FUSEB experts for the descriptors listed in Sec. 6.2. The classifiers are trained on 2013d and applied on 2013t. The fusions are also trained on experts from 2013d, and fusion results are evaluated on 2013t. In the case of parameter optimizations for experts or fusions, they are done in cross-validation on 2013d.

We report mean infAP averaged over a subset of 38 concepts out of the total 346, the same concepts that are used for evaluating official TRECVid SIN 2013 submissions [25].

6.3.1 Global results

Table 1 (column *"basic"*) shows the mean infAP obtained by the proposed fusion methods. The *manual hierarchical fusion* performs the best, thanks to the carefully-optimized weights of experts, the additional score normalization steps between fusion stages and the manual grouping of experts that ensures more homogeneous properties within a group.

Table 1 Mean (over all concepts) inferred average precisions of fusion approaches: basic (without any post-processing), +RS (with temporal re-scoring, *temporal context* integration), +RS+CF (with RS followed by conceptual feedback, *semantic context* integration), +RS+CF+RS (+RS+CF followed by a second RS).

	basic	+RS	+RS+CF	+RS+CF+RS
Manual hierarchical fusion	0.2576	0.2695	0.2758	0.2848
Adaboost score-based fusion	0.2500	0.2630	-	-
Adaboost rank-based fusion	0.2346	0.2534	-	-
Agglomerative clustering fusion	0.2383	0.2516	-	-
Weighted average fusion	0.2264	0.2409	-	-
Best expert per concept	0.2162	0.2367	-	-
Selected best from 5 above	0.2495	0.2631	-	-

Among the automatic methods, the *Adaboost score-based fusion* performs the best, with performances not far behind the manually-optimized hierarchical fusion. The *Adaboost rank-based fusion* performs less good, because the rank of a shot can vary greatly with small variations in the classification score, which makes the method more sensitive to classification noise. The *agglomerative clustering fusion* is relatively close in global results to the *Adaboost rank-based fusion*. Among the fusion methods, the *weighted average fusion* is the least good, showing that a performance boost can be obtained with more careful expert weight choosing strategies; for example, the *Adaboost score-based fusion* performs 10% better than the weighted average.

In any case, it can be seen that whatever the fusion method, the global result is always better than what would have been obtained if we would have taken, for each concept, its best expert on the training dataset (*Best expert per concept*). The *manual hierarchical fusion* is 19% better, the *Adaboost score-based fusion* is 16% better and the even the *weighted average* has a 5% improvement, proving that late fusion schemes, even naive ones, generally improve concept detection performances.

The *selected best fusion* selects, for each concept, the fusion approach (among *Adaboost score-based fusion*, *Adaboost rank-based fusion*, *agglomerative clustering*, *weighted average* and the *best expert for that concept*) that performed the best on the training set. The *Adaboost score-based fusion* was by far chosen the most often, for 230 out of the 346 concepts, which is in agreement with it having the highest mean infAP. The *Adaboost rank-based fusion* was chosen for 60 concepts, the *agglomerative clustering* for 14 concepts and the *weighted average* for only 8 concepts. For the rest of the 34 concepts, the *best expert* was chosen, because the fusions were found to degrade performances on the training dataset. Considering this, it was to be expected that the mean infAP of the *selected best fusion* would be close but slightly above that of the *Adaboost score-based fusion*. However, no global gain is observed for the emphselected best fusion, because the choices made on the training set are not always the best also for the test dataset, due to variations between the two datasets.

6.3.2 Concept-per-concept results

Moving on to a concept-per-concept analysis, Table 2 shows the infAP gains for the 38 semantic concepts used in the official TRECVid evaluation, when comparing the best of the automatic methods (the *Adaboost score-based fusion*) with the baseline *best expert per concept*. For the majority of concepts, the fusion gives a significant performance boost (such as for *Airplane, Bus, Hand, Running, Throwing*). For some concepts, the boost is not too high, especially for concepts that already have large infAP to start with (such as *Beach, Government leader, Instrumental musician, Skating*); this happens when the other experts do not bring any pertinent and complementary information compared to the best expert. There are only 6 concepts that experience performance degradations from the fusion, namely *Animal, Computers, Explosion or fire, Female face closeup, Girl and Kitchen*.

As a preliminary conclusion, we can say that fusing a large battery of complementary experts yields a significant performance increase. It is now time to examine the gains of higher-level fusions, at the temporal and semantic context levels.

6.3.3 Results for higher-level fusions

Table 1, column *"RS"* shows the mean infAP after applying the temporal re-scoring algorithm described in Sec. 5. Our best-performing method, the *manual hierarchical fusion*, has a gain of 4,6%, while the other methods also experience gains in the range of 5-10%. This shows that the temporal context can also bring useful information, resulting in a performance increase for all methods.

Table 2 Comparison of inferred average precisions for the *best expert per concept* and the *AdaBoost score-based fusion*, for particular concepts.

concept	best expert	AdaBoost sc.	rel. gain (%)
Airplane	0.0573	0.0923	61
Anchorperson	0.4850	0.5988	23
Animal	0.0659	0.0078	-88
Beach	0.4658	0.4722	1
Boat or ship	0.2907	0.3083	6
Boy	0.0291	0.0316	9
Bridges	0.0372	0.0393	6
Bus	0.0273	0.0598	119
Chair	0.1621	0.2394	48
Computers	0.2647	0.1919	-28
Dancing	0.2990	0.4019	34
Explosion or fire	0.1780	0.1617	-9
Female face closeup	0.3741	0.3550	-5
Flowers	0.1752	0.1895	8
Girl	0.0462	0.0360	-22
Government leader	0.4387	0.4546	4
Hand	0.1532	0.2847	86
Instrumental musician	0.5141	0.5782	12
Kitchen	0.1072	0.0952	-11
Motorcycle	0.1778	0.2369	33
News studio	0.7213	0.8223	14
Old people	0.3719	0.4096	10
People marching	0.0388	0.0470	21
Running	0.0863	0.1405	63
Singing	0.1096	0.1459	33
Sitting down	0.0003	0.0023	667
Telephones	0.0063	0.0133	111
Throwing	0.1121	0.2506	124
Baby	0.1317	0.2234	70
Door opening	0.0369	0.0410	11
Fields	0.0753	0.1375	83
Flags	0.2607	0.2819	8
Forest	0.0911	0.1150	26
George Bush	0.6092	0.6624	9
Military airplane	0.0172	0.0381	122
Quadruped	0.0807	0.1133	40
Skating	0.4956	0.5328	8
Studio with anchorperson	0.6228	0.6871	10

After temporal re-scoring, we apply the conceptual feedback step described in Sec. 5 (*+RS+CF* in Table 1). Because of the significant computational cost, we limit this experiment to our best-performing method, the *manual hierarchical fusion*, for which an additional gain of 2,3% is obtained compared to the previous result. Adding a second temporal re-scoring step after the conceptual feedback (*+RS+CF+RS*) increases results by another 3,3%. In the end, the successive temporal re-scoring and conceptual feedback steps give an increase of 10,5% compared to the basic approach.

7 Conclusion

In this work, we proposed several methods of combining dozens of input experts into better ones, and applied these methods in the context of the *TRECVid 2013 Semantic Indexing* task. We have shown that all of the methods globally outperform taking the best expert for each concept, and that more elaborate fusions can perform better than a naive weighted arithmetic mean. Two late fusion methods distinguish themselves, a manually-optimised hierarchical grouping of experts and an automatic fusion based on AdaBoost, both with a relatively low computational complexity. Even though we experimented on the TRECVid SIN video dataset, these approaches are generic and can be extended to other multimedia collections as well. We have also shown that additional levels of fusions that exploit context can give an additional performance increase: in the case of a video dataset, the temporal and semantic context were tested, while for other multimedia datasets, different types of contextual fusions could be devised, for example by considering the identity of the multimedia sample's uploader, the date and time when the material was created and/or uploaded etc. In the future, we plan to extend our work to such types of multimedia datasets.

Acknowledgements This work was supported by the Quaero Program and the QCompere project, respectively funded by OSEO (French State agency for innovation) and ANR (French national research agency). The authors would also like to thank the members of the IRIM consortium for the expert scores used throughout the experiments described in this paper.

References

1. Ayache, S., Quénot, G., Gensel, J.: Image and video indexing using networks of operators. J. Image Video Process. **2007**(3), 1:1–1:13 (2007). DOI 10.1155/2007/56928. URL http://dx.doi.org/10.1155/2007/56928
2. Ballas, N., Delezoide, B., Prêteux, F.: Trajectories based descriptor for dynamic events annotation. In: Proceedings of the 2011 joint ACM workshop on Modeling and representing events, J-MRE '11, pp. 13–18. ACM, New York, NY, USA (2011). DOI 10.1145/2072508.2072512. URL http://doi.acm.org/10.1145/2072508.2072512
3. Ballas, N., Labbé, B., Shabou, A., Borgne, L.: Cea list at trecvid 2012: Semantic Indexing and Instance Search. In: Proc. TRECVid Workshop. Gaithersburg, MD, USA (2012)
4. Ballas, N., Labbé, B., Shabou, A., Le Borgne, H., Gosselin, P., Redi, M., Merialdo, B., Jégou, H., Delhumeau, J., Vieux, R., Mansencal, B., Benois-Pineau, J., Ayache, S., Hamadi, A., Safadi, B., Thollard, F., Derbas, N., Quenot, G., Bredin, H., Cord, M., Gao, B., Zhu, C., Tang, Y., Dellandrea, E., Bichot, C.E., Chen, L., Benoit, A., Lambert, P., Strat, T., Razik, J., Paris, S., Glotin, H., Trung, T.N., Petrovska-Delacrétaz, D., Chollet, G., Stoian, A., Crucianu, M.: IRIM at TRECVid 2012: Semantic Indexing and Instance Search. In: Proceedings of the workshop on TREC Video Retrieval Evaluation (TRECVid), p. 12p. Gaithersburg, MD, États-Unis (2012). URL http://hal.archives-ouvertes.fr/hal-00770258. CNRS, RENATER, several Universities, other funding bodies (see https://www.grid5000.fr).
5. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). Comput. Vis. Image Underst. **110**(3), 346–359 (2008). DOI 10.1016/j.cviu.2007.09.014. URL http://dx.doi.org/10.1016/j.cviu.2007.09.014

6. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast Unfolding of Communities in Large Networks. Journal of Statistical Mechanics: Theory and Experiment **2008**(10), P10,008 (2008). URL `http://stacks.iop.org/1742-5468/2008/i=10/a=P10008`
7. Breiman, L., Breiman, L.: Bagging predictors. In: Machine Learning, pp. 123–140 (1996)
8. Cai, N., Li, M., Lin, S., Zhang, Y., Tang, S.: Ap-based adaboost in high level feature extraction at trecvid. In: Pervasive Computing and Applications, 2007. ICPCA 2007. 2nd International Conference on, pp. 194–198 (2007). DOI 10.1109/ICPCA.2007.4365438
9. Cao, L., Chang, S.F., Codella, N., Cotton, C., Ellis, D., Gong, L., Hill, M., Hua, G., Kender, J., Merler, M., Mu, Y., Smith, J.R., Felix, X.Y.: Ibm research and columbia university trecvid-2012 multimedia event detection (med), multimedia event recounting (mer), and semantic indexing (sin) systems. In: NIST TRECVid Workshop. Gaithersburg, MD (2012)
10. Cliville, V., Berrah, L., Mauris, G.: Information fusion in industrial performance: a 2-additive choquet-integral based approach. In: Systems, Man and Cybernetics, 2004 IEEE International Conference on, vol. 2, pp. 1297–1302 vol.2 (2004). DOI 10.1109/ICSMC.2004.1399804
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09 (2009)
12. Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision **88**(2), 303–338 (2010)
13. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences **55**(1), 119–139 (1997). DOI http://dx.doi.org/10.1006/jcss.1997.1504. URL `http://www.sciencedirect.com/science/article/pii/S002200009791504X`
14. Gönen, M., Alpaydın, E.: Multiple kernel learning algorithms. J. Mach. Learn. Res. **12**, 2211–2268 (2011). URL `http://dl.acm.org/citation.cfm?id=1953048.2021071`
15. Gosselin, P.H., Cord, M., Philipp-Foliguet, S.: Combining visual dictionary, kernel-based similarity and learning strategy for image category retrieval. Comput. Vis. Image Underst. **110**(3), 403–417 (2008). DOI 10.1016/j.cviu.2007.09.018. URL `http://dx.doi.org/10.1016/j.cviu.2007.09.018`
16. Hamadi, A., Quénot, G., Mulhem, P.: Conceptual Feedback for Semantic Multimedia Indexing. In: Content-Based Multimedia Indexing (CBMI), 2013 11th International Workshop on. Veszprém, Hungary (2013)
17. Kendall, M.G.: Rank correlation methods. Griffin, London (1948)
18. Laptev, I.: On space-time interest points. International Journal of Computer Vision **64**(2-3), 107–123 (2005)
19. Little, S., Llorente, A., Rüger, S.: An overview of evaluation campaigns in multimedia retrieval. In: H. Müller, P. Clough, T. Deselaers, B. Caputo (eds.) ImageCLEF, *The Information Retrieval Series*, vol. 32, pp. 507–525. Springer Berlin Heidelberg (2010). DOI 10.1007/978-3-642-15181-1_27. URL `http://dx.doi.org/10.1007/978-3-642-15181-1_27`
20. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60**(2), 91–110 (2004). DOI 10.1023/B:VISI.0000029664.99615.94. URL `http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94`
21. Negrel, R., Picard, D., Gosselin, P.: Compact tensor based image representation for similarity search. In: Image Processing (ICIP), 2012 19th IEEE International Conference on, pp. 2425–2428 (2012). DOI 10.1109/ICIP.2012.6467387
22. Newman, M.E.J.: Modularity and Community Structure in Networks. Proceedings of the National Academy of Sciences of the United States of America **103**(23), 8577–8582 (2006). DOI 10.1073/pnas.0601602103. URL `http://www.pnas.org/cgi/content/abstract/103/23/8577`
23. Ng, K.B., Kantor, P.B.: Predicting the Effectiveness of Naive Data Fusion on the Basis of System Characteristics. Journal of the American Society for Information Science **51**, 1177–1189 (2000). DOI 10.1002/1097-4571(2000)9999:9999⟨::AID-ASI1030⟩3.0.CO;2-E. URL `http://dl.acm.org/citation.cfm?id=357868.357870`

24. Over, P., Awad, G., Michel, M., Fiscus, J., Kraaij, W., Smeaton, A.F., Quénot, G.: Trecvid 2011 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In: Proceedings of TRECVid 2011. NIST, USA (2011)
25. Over, P., Awad, G., Michel, M., Fiscus, J., Sanders, G., Kraaij, W., Smeaton, A.F., Quénot, G.: Trecvid 2013 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In: Proceedings of TRECVID 2013. NIST, USA (2013)
26. Paris, S., Glotin, H.: Pyramidal multi-level features for the robot vision@icpr 2010 challenge. In: Pattern Recognition (ICPR), 2010 20th International Conference on, pp. 2949–2952 (2010). DOI 10.1109/ICPR.2010.1143
27. Pinquier, J., Karaman, S., Letoupin, L., Guyot, P., Megret, R., Benois-Pineau, J., Gaestel, Y., Dartigues, J.F.: Strategies for multiple feature fusion with hierarchical hmm: Application to activity recognition from wearable audiovisual sensors. In: Pattern Recognition (ICPR), 2012 21st International Conference on, pp. 3192–3195 (2012)
28. Redi, M., Merialdo, B.: Saliency moments for image categorization. In: Proceedings of the 1st ACM International Conference on Multimedia Retrieval, ICMR '11, pp. 39:1–39:8. ACM, New York, NY, USA (2011). DOI 10.1145/1991996.1992035. URL `http://doi.acm.org/10.1145/1991996.1992035`
29. Safadi, B., Quénot, G.: Evaluations of multi-learner approaches for concept indexing in video documents. In: Adaptivity, Personalization and Fusion of Heterogeneous Information, RIAO '10, pp. 88–91. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, Paris, France, France (2010). URL `http://dl.acm.org/citation.cfm?id=1937055.1937075`
30. Safadi, B., Quénot, G.: Re-ranking for Multimedia Indexing and Retrieval. In: ECIR 2011: 33rd European Conference on Information Retrieval, pp. 708–711. Springer, Dublin, Ireland (2011)
31. Safadi, B., Quénot, G.: Descriptor Optimization for Multimedia Indexing and Retrieval. In: CBMI 2013, 11th International Workshop on Content-Based Multimedia Indexing. Veszprem, HUNGARY (2013)
32. Sánchez, J., Perronnin, F., Mensink, T., Verbeek, J.: Image classification with the fisher vector: Theory and practice. International Journal of Computer Vision **105**(3), 222–245 (2013). DOI 10.1007/s11263-013-0636-x. URL `http://dx.doi.org/10.1007/s11263-013-0636-x`
33. van de Sande, K.E.A., Gevers, T., Snoek, C.G.M.: Evaluating color descriptors for object and scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(9), 1582–1596 (2010). URL `http://www.science.uva.nl/research/publications/2010/vandeSandeTPAMI2010`
34. Schapire, R.E., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Mach. Learn. **37**(3), 297–336 (1999). DOI 10.1023/A:1007614523901. URL `http://dx.doi.org/10.1023/A:1007614523901`
35. Shabou, A., Borgne, H.L.: Locality-constrained and spatially regularized coding for scene categorization. In: CVPR, pp. 3618–3625. IEEE (2012). URL `http://dblp.uni-trier.de/db/conf/cvpr/cvpr2012.html#ShabouL12`
36. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, Princeton (1976)
37. Smeaton, A.F., Over, P., Kraaij, W.: High-Level Feature Detection from Video in TRECVid: a 5-Year Retrospective of Achievements. In: A. Divakaran (ed.) Multimedia Content Analysis, Theory and Applications, pp. 151–174. Springer Verlag, Berlin (2009)
38. Snoek, C.G.M., van de Sande, K.E.A., Habibian, A., Kordumova, S., Li, Z., Mazloom, M., Pintea, S.L., Tao, R., Koelma, D.C., Smeulders, A.W.M.: The mediamill trecvid 2012 semantic video search engine. In: Proceedings of the TRECVid Workshop (2012). URL `http://www.science.uva.nl/research/publications/2012/SnoekPTRECVid2012a`
39. Strat, S., Benoit, A., Lambert, P.: Retina enhanced sift descriptors for video indexing. In: Content-Based Multimedia Indexing (CBMI), 2013 11th International Workshop on, pp. 201–206 (2013). DOI 10.1109/CBMI.2013.6576582

40. Strat, S., Benoit, A., Lambert, P., Caplier, A.: Retina-enhanced surf descriptors for semantic concept detection in videos. In: Image Processing Theory, Tools and Applications (IPTA), 2012 3rd International Conference on, pp. 319–324 (2012). DOI 10.1109/IPTA.2012.6469557
41. Strat, S.T., Benoit, A., Lambert, P., Caplier, A.: Retina enhanced surf descriptors for spatio-temporal concept detection. Multimedia Tools and Applications pp. 1–27 (2013). DOI 10.1007/s11042-012-1280-0. URL http://dx.doi.org/10.1007/s11042-012-1280-0
42. Strat, T., Benoit, A., Bredin, H., Quenot, G., Lambert, P.: Hierarchical Late Fusion for Concept Detection in Videos. In: V.M.R.C. Andrea Fusiello (ed.) Proceedings of Computer Vision - ECCV 2012. Workshops and Demonstrations, Part III, *Lecture Notes in Computer Science (LNCS)*, vol. 7585, pp. 335–344. Springer Berlin, Firenze, Italie (2012). DOI 10.1007/978-3-642-33885-4_34. URL http://hal.archives-ouvertes.fr/hal-00732740. Oral session 1: WS21 - Workshop on Information Fusion in Computer Vision for Concept Recognition OSEO (French State agency for innovation) and ANR (French national research agency)
43. Tang, Z., Yanai, K.: Uec at trecvid 2008 high level feature task. In: P. Over, G. Awad, R.T. Rose, J.G. Fiscus, W. Kraaij, A.F. Smeaton (eds.) TRECVid. National Institute of Standards and Technology (NIST) (2008)
44. Wang, H., Kläser, A., Schmid, C., Cheng-Lin, L.: Action Recognition by Dense Trajectories. In: IEEE Conference on Computer Vision & Pattern Recognition, pp. 3169–3176. Colorado Springs, United States (2011). URL http://hal.inria.fr/inria-00583818
45. Wu, L., Guo, Y., Qiu, X., Feng, Z., Rong, J., Jin, W., Zhou, D., Wang, R., Jin, M.: Fudan university at trecvid 2003. In: Notebook of TRECVid (2003)
46. Yilmaz, E., Aslam, J.A.: Estimating average precision with incomplete and imperfect judgments. In: Proceedings of the 15th ACM international conference on Information and knowledge management, CIKM '06, pp. 102–111. ACM, New York, NY, USA (2006). DOI 10.1145/1183614.1183633. URL http://doi.acm.org/10.1145/1183614.1183633
47. Yilmaz, E., Kanoulas, E., Aslam, J.A.: A Simple and Efficient Sampling Method for Estimating AP and NDCG. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '08, pp. 603–610. ACM, New York, NY, USA (2008). DOI http://doi.acm.org/10.1145/1390334.1390437. URL http://doi.acm.org/10.1145/1390334.1390437
48. Zhang, L., Jiang, L., Bao, L., Takahashi, S., Li, Y., A., H.: Informedia@trecvid 2011: Surveillance event detection. TRECVid Video Retrieval Evaluation Workshop, Gaitherburg, USA (2011)
49. Zhu, C., Bichot, C.E., Chen, L.: Image region description using orthogonal combination of local binary patterns enhanced with color information. Pattern Recogn. **46**(7), 1949–1963 (2013). DOI 10.1016/j.patcog.2013.01.003. URL http://dx.doi.org/10.1016/j.patcog.2013.01.003
50. Znaidia, A., Borgne, H.L., Hudelot, C.: Belief theory for large-scale multi-label image classification. In: T. Denoeux, M.H. Masson (eds.) Belief Functions, *Advances in Soft Computing*, vol. 164, pp. 205–212. Springer (2012)