IRIM at TRECVID 2013: Semantic Indexing and Instance Search

Nicolas Ballas[1], Benjamin Labbé[1], Hervé Le Borgne[1], Philippe Gosselin[2], David Picard[2], Miriam Redi[3], Bernard Mérialdo[3], Iván González-Díaz[4], Boris Mansencal[4], Jenny Benois-Pineau[4], Stéphane Ayache[5], Abdelkader Hamadi[6], Bahjat Safadi[6], Thi-Thu-Thuy Vuong[6], Han Dong[6], Nadia Derbas[6], Georges Quénot[6], Boyang Gao[7], Chao Zhu[7], Yuxing tang[7], Emmanuel Dellandrea[7], Charles-Edmond Bichot[7], Liming Chen[7], Alexandre Benoît[8], Patrick Lambert[8], and Tiberius Strat[8]

[1]CEA, LIST, Laboratory of Vision and Content Engineering, Gif-sur-Yvettes, France.

[2]ETIS UMR 8051, ENSEA / Université Cergy-Pontoise / CNRS, Cergy-Pontoise Cedex, F-95014 France

[3]EURECOM, Campus SophiaTech, 450 Route des Chappes, CS 50193, 06904 Biot Sophia Antipolis cedex, France

[4]LABRI UMR 5800, Université Bordeaux 1 / Université Bordeaux 2 / CNRS / ENSEIRB, Talence Cedex, France

[5]LIF UMR 6166, CNRS / Université de la Méditerranée / Université de Provence, F-13288 Marseille Cedex 9, France

[6]UJF-Grenoble 1 / UPMF-Grenoble 2 / Grenoble INP / CNRS, LIG UMR 5217, Grenoble, F-38041, France

[7]LIRIS, UMR 5205 CNRS / INSA de Lyon / Université Lyon 1 / Université Lyon 2 / École Centrale de Lyon, France

[8]LISTIC, Domaine Universitaire, BP 80439, 74944 Annecy le vieux Cedex, France

Abstract

The IRIM group is a consortium of French teams working on Multimedia Indexing and Retrieval. This paper describes its participation to the TRECVID 2013 semantic indexing and instance search tasks. For the semantic indexing task, our approach uses a six-stages processing pipelines for computing scores for the likelihood of a video shot to contain a target concept. These scores are then used for producing a ranked list of images or shots that are the most likely to contain the target concept. The pipeline is composed of the following steps: descriptor extraction, descriptor optimization, classification, fusion of descriptor variants, higher-level fusion, and re-ranking. We evaluated a number of different descriptors and tried different fusion strategies. The best IRIM run has a Mean Inferred Average Precision of 0.2796, which ranked us 4th out of 26 participants.

1 Semantic Indexing

1.1 Introduction

The TRECVID 2013 semantic indexing task is described in the TRECVID 2013 overview paper [1, 2]. 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 the test collection, master shot reference, and concept/feature definitions, return for each feature a list of at most 2000 shot IDs from the test collection ranked according to the possibility of detecting the feature." 60 concepts have been selected for the TRECVID 2013 semantic indexing task. Annotations on the development part of the collection were provided for 346 concepts including the 60 target ones in the context of a collaborative annotation effort [16].

Nine French groups (CEA-LIST, CNAM, ETIS, EURECOM, LABRI, LIF, LIG, LIRIS, LISTIC) collaborated to participate to the TRECVID 2013 semantic indexing task. Xerox (XRCE), though not being member of IRIM, also shared descriptors with us.

The IRIM approach uses a six-stages processing pipeline that compute scores reflecting the likelihood of a video shot to contain a target concept. These scores are then used for producing a ranked list of images or shots that are the most likely to contain the target concept. The pipeline is composed of the following steps:

1. Descriptor extraction. A variety of audio, image and motion descriptors have been produced by the participants (section 1.2).
2. Descriptor optimization. A post-processing of the descriptors allows to simultaneaously improve

their performance and to reduce their size (section 1.3).

3. Classification. Two types of classifiers are used as well as their fusion (section 1.4).

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 1.6).

5. Higher-level fusion. We fuse here descriptors of different types, e.g. color, texture, interest points, motion (section 1.7).

6. Re-ranking. We post-process here the scores using the fact that videos statistically have an homogeneous content, at least locally (section 1.8).

This approach is quite similar to the one used by the IRIM group last year [15]. The main novelties are again the inclusion of new descriptors, some improvements in the pre-processins step and improvements in the automatic fusion methods.

1.2 Descriptors

Eight IRIM participants (CEA-LIST, ETIS, EURECOM, LABRI, LIF, LIG, LIRIS and LISTIC) provided a total of 55 descriptors, including variants of a same descriptors. Xerox (XRCE) also provided two descriptors with us. 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. The relative performance of these descriptors has been separately evaluated using a combination of LIG classifiers (see section 1.5). Here is a description of these descriptors:

CEALIST/tlep: texture local edge pattern [3] + color histogram ⇝ 576 dimensions.

CEALIST/bov_dsiftSC_8192: : bag of visterm[37]. Dense SIFT are extracted every 6 pixels. The codebook of size 1024 is built with K-means. The bag are generated with soft coding and max pooling. The final signature result from a three levels spatial pyramid ⇝ $1024 \times (1+2\times2+3\times1) = 8192$ dimensions: see [17] for details.

CEALIST/bov_dsiftSC_21504: : bag of visterm[37]. Same as CEALIST/bov_dsiftSC_8192 with a different spatial pyramid ⇝ $1024 \times (1 + 2 \times 2 + 4 \times 4) = 21504$ dimensions.

ETIS/global_<feature>[<type>]x<size>: (concatenated) histogram features[4], where:

<feature> is chosen among lab and qw:

lab: CIE L*a*b* colors

qw: quaternionic wavelets (3 scales, 3 orientations)

<type> can be:

m1x1: 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.

ETIS/vlat_<desc type>_dict<dict size>_<size>: compact Vectors of Locally Aggregated Tensors (VLAT [6]). <desc type> = low-level descriptors, for instance hog6s8 = dense histograms of gradient every 6 pixels, 88 pixels cells. <dict size> = size of the low-level descriptors dictionary. <size> = size of feature for one frame. Note: these features can be truncated. These features must be normalized to be efficient (e.g. L_2 unit length).

EUR/sm462: The Saliency Moments (SM) feature [5] is a holistic descriptor that embeds some locally-parsed information, namely the shape of the salient region, in a holistic representation of the scene, structurally similar to [7].

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.

LIG/h3d64: normalized RGB Histogram $4 \times 4 \times 4$ ⇝ 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 Koen Van de Sande's software[8] ⇝ 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/concepts: detection scores on the 346 TRECVID 2011 SIN concepts using the best available fusion with the other descriptors, $\rightsquigarrow$ 346 dimensions.

LIRIS/OCLPB_DS_4096: Dense sampling OCLBP [38] bag-of-words descriptor with 4096 k-means clusters. We extract orthogonal combination of local binary pattern (OCLBP) to reduce original LBP histogram size and at the same time preserve information on all neighboring pixels. Instead of encoding local patterns on 8 neighbors, we perform encoding on two sets of 4 orthogonal neighbors, resulting two independent codes. Concatenating and accumulating two codes leads to a final 32 dimensional LBP histogram, compared with original 256 dimensions. The 4096 bag-of-words descriptors are finally generated by the pre-trained dictionary.

LIRIS/MFCC_4096: MFCC bag-of-words descriptor with 4096 k-means clusters. To reserves video's sequential information, we keep 2 seconds audio wave around the key frame, 1 second before and after. 39 dimensional MFCC descriptors with delta and delta delta are extracted with 20ms window length and 10ms window shift. The 4096 bag-of-words descriptors are finally generated by the pre-trained dictionary.

LISTIC/SIFT_*: Bio-inspired retinal preprocessing strategies applied before extracting Bag of Words of Opponent SIFT features (details in [25]) using the retinal model from [9]). Features extracted on dense grids on 8 scales (initial sampling=6 pixels, initial patch=16x16pixels, using a linear scale factor 1.2). K-means clusters of 1024 or 2048 visual words. The proposed descriptors are similar to those from [25] except the fact that multiscale dense grids are used. Despite showing equivalent mean average performance, the various prefiltering strategies present different complementary behaviours that boost performances at the fusion stage.

LISTIC/trajectories_*: Bag of Words of trajectories of tracked points. Various ways of describing a trajectory are used, such as the spatial appearance along a trajectory, the motion along a trajectory or a combination of both. Each type of trajectory description generates its own Bag of Words representation. K-means clustering of 256-1024 visual words, depending on the type of description.

XEROX/ilsvrc2010: Attribute type descriptor constituted as vector of classification score obtained with classifiers trains on external data with one vector component per trained concept classifier. For XEROX/ilsvrc2010, 1000 classifiers were trained using annotated data from the Pascal VOC / Imanget ILSVRC 2010 challenge. Classification was done using Fisher Vectors [12].

XEROX/imagenet10174: Attribute type descriptor similar to XEROX/ilsvrc2010 but with 10174 concepts trained using ImageNet annotated data.

1.3 Descriptor optimization

The descriptor optimization consists of a principal component analysis (PCA) based dimensionality reduction with pre and post power transformations [24]. A L_1 or L_2 unit length normalization can optionally by applied after the first power transformation.

First power transformation: The goal of the power transformation is to normalize the distributions of the values, especially in the case of histogram components. It simply consists in applying an $x \leftarrow x^\alpha$ ($x \leftarrow -(-x)^\alpha$ if $x < 0$) tranformation on all components individually. The optimal value of α_1 can be optimized by cross-validation and is often close to 0.5 for histogram-based descriptors.

Principal component analysis: The goal of PCA reduction is both to reduce the size (number of dimensions) of the descriptors and to improve performance by removing noisy components.

Second power transformation: The second power transformation has an affect which is similar to a post-PCA whitening but is has been proven to be more efficient and easy to tune. The optimal value of α_2 can be optimized by cross-validation and is often close to 0.7.

The optimization of the value of the α_1 and α_2 coefficient and of the number of components kept in the PCA reduction is optimized by two-fold cross-validation within the development set. In practice, it is done with the LIG_KNNB classifier only (see section 1.4), 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. Moreover, the overall performance is not very sensitive to the precise values for these hyper-parameters.

1.4 Classification

The LIG participant ran 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 a 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 usually gives lower classification rates than the SVM-based one but is much faster.

LIG_MSVM: The second one is based on a multiple learner approach with SVMs. The multiple learner approach is well suited for the imbalanced data set problem [13], which is the typical case in the TRECVID SIN task in which the ratio between the numbers of negative and positive training sample is generally higher than 100:1.

LIG_FUSEB: Fusion between classifiers. The fusion is simply done by a MAP weighted average of the scores produced by the two classifiers. Their output is naturally (or by construction) normalized in the the [0:1] range. kNN computation is done using the KNNLSB package [14]. Even though the LIG_MSVM classifier is often significantly better than the LIG_KNNB one, the fusion is most often even better, probably because they are very different in term of information type capture. The MAP values used for the weighting are obtained by a two-fold cross-validation within the development set.

1.5 Evaluation of classifier-descriptors combinations

We evaluated a number of image descriptors for the indexing of the 346 TRECVID 2012 concepts. This has been done with two-fold cross-validation within the development set. We used the annotations provided by the TRECVID 2013 collaborative annotation organized by LIG and LIF [18]. The performance is measured by the inferred Mean Average Precision (MAP) computed on the 346 concepts. Results are presented for the two classifiers used, as well as for their fusion. Results are presented only for the best combinations of the descriptor optimization hyper-parameters.

Table 1 shows the two-fold cross-validation performance (trec_eval MAP) within the development set and the performance (sample_eval MAP) on the test set for all used descriptors with the LIG_FUSEB classifier combination; dim is the original number of dimensions of the descriptor vector, Pdim is the number of dimensions of the descriptor vector kept after PCA reduction, and α_1 and α_2 are the optimal values of the pre- and post-PCA power transformation coefficients.

1.6 Performance improvement by fusion of descriptor variants and classifier variants

As in previous years, we started by fusing classification scores from different variants of a same descriptor and from different classifiers of a same variant of a same descriptor. This is done as first levels of hierarchical late fusion, the last ones being done using dedicated methods as described in section 1.7. Three levels are considered when applicable: fusions of different classifiers of a same variant of a same descriptor, fusion of different variants of a same descriptor according to a dictionary size, and fusion of different variants of a same descriptor according to a pyramidal decomposition. While the last levels of fusion attempt to improve the overall performance by fusing information of different types (e.g;. color, texture, percepts or SIFT), the first fusion levels attempt to improve the robustness of the classification from a given type. More details on this approach can be found in the previous TRECVid IRIM papers [20, 15].

1.7 Final fusion

The IRIM participant LISTIC worked on the automatic fusion of the classification results (experts). The fusion started with the original classification scores and/or with the results of previous fusions of descriptor variants and/or classifier variants as described in the previous section. A comparison of the LISTIC and LIMSI automatic fusion methods, along with another fusion method tried in the context of the Quaero group using some of the same classification results, and an arithmetic mean and the best attribute per concept, is given in [36].

We combine all of the available FUSEB experts (55 experts in total), in a concept-per-concept manner, by performing five late fusions in parallel. The first fusion is the agglomerative clustering approach which we have previously seen in in [36] and in [15]. The second fusion is based on optimising classification scores by using AdaBoost. The third fusion also uses AdaBoost, but this time attempting to optimise the rankings of video shots instead of their scores. The fourth fusion is a weighted arithmetic mean of the input experts, with weights given by the average precisions of the expert for the semantic concept in question. The fifth fusion consists in taking just the best expert for the concept in question. All of these five fusions are combined, by choosing for the concept in question, the late fusion approach that worked best on the training set.

1.8 Temporal re-scoring (re-ranking) and conceptual feedback

At the end, temporal re-scoring [23] and conceptual feedback [26] are performed. For reasons of time constraints, conceptual feedback is performed using information from a manual hierarchical late fusion [16] instead of our own fusions.

1.9 Evaluation of the submitted runs

We submitted 4 runs, each using the same 55 input experts:

- M_A_IRIM1_1: the best of the 5 fusion approaches for each concept, followed by temporal re-scoring, conceptual feedback and a second temporal rescoring;
- M_A_IRIM2_2: similar to the above system, but without conceptual feedback and without the second temporal re-scoring;
- M_A_IRIM3_3: Score-optimising AdaBoost fusion, followed by temporal re-scoring;
- M_A_IRIM4_4: Agglomerative clustering fusion, followed by temporal re-scoring;

IRIM officially submitted the four M_A_IRIM1_1 to M_A_IRIM2_4 runs that are described in section 1.7. Table 2 presents the result obtained by the four runs submitted as well as the best and media runs for comparison. The best IRIM run corresponds to a rank of 4 within the 16 participants to the TRECVID 2012 full SIN task.

Table 2: InfMAP result and rank on the test set for all the 38 TRECVID 2013 evaluated concepts (main task).

System/run	MAP	rank
Best run	0.3211	1
M_A_IRIM1_1	0.2796	13
M_A_IRIM2_2	0.2588	15
M_A_IRIM4_4	0.2521	17
M_A_IRIM3_3	0.2508	20
Median run	0.1275	46
Random run	0.0009	-

Table 2 shows the results of our submitted runs. The best run is IRIM1_1, which is the concept-per-concept selection of the best of the 5 late fusion approaches, with added temporal re-scoring, conceptual feedback and a second temporal re-scoring. Compared with IRIM2_2, which is similar but does not use conceptual feedback and the second temporal re-scoring, there is an increase of 8%, thereby showing the added benefit of the conceptual feedback approach from [26]. Further tests are needed for this step, because for reasons due to time constraints, we used the hierarchical late fusion from [16] as the source of conceptual feedback, instead of our IRIM2_2 run. However, we expect to see a similar increase in performance, because our IRIM2_2 run has very close results to this hierarchical late fusion, and both are based on the same input experts.

Putting aside conceptual feedback, IRIM2_2 is the best of our approaches, as it selects, for each concept, the best of the 5 late fusion approaches on the training set. This prevents results from being affected by occasional (for some concepts) performance decreases due to fusion, by reverting to an approach that is unaffected. IRIM2_2 is therefore 2.6% better than our third-best approach, the agglomerative clustering fusion. The agglomerative clustering and the AdaBoost score-optimising fusion perform close on a global level, with differences of only 0.5%.

We have also compared our fusion approaches with selecting the best expert for each concept individually (this is in fact one of the 5 fusion methods used for IRIM2_2), which gives a MAP of 0.2367. This places all of our fusion methods above this baseline, with the greatest increase, of 9%, belonging to IRIM2_2.

2 Instance Search

Given visual examples of entities of limited number of types: person, character, object or location, Instance Search (INS) task [2] consists in finding segments of videos in the data set which contain instances of these entities. Each instance is represented by a few example images. Hence if we can consider the set of video clips as a visual database, the problem consists in retrieval of each instance in this database.

2.1 Global approach

To represent the clips we extract several keyframes of each individual video clip. For a given instance, we use each example image, from the available set, as a query image. We compute a similarity between this query example image and the keyframes of all video clips. We then produce an intermediary result where we have the similarity $S_{e,i,k,c}$ between each example image (e) of each instance (i) and each keyframe (k) of each video clip (c). We then have to fuse these intermediary results to obtain a final result that is similarity $S_{i,c}$ between each instance (i) and each clip (c) Within the IRIM consortium, several methods of four members (CEA, CNAM, LaBRI, LISTIC) were tested and their results fused.

2.2 Members methods

CEA_Markrs The Markrs are local features for geometrical registration of objects in couple of images. The Markrs process of image description and matching follows the well-known framework of keypoint matching described in [39]. For this experiment, we used the SURF scheme [27] to detect salients keypoints and compute corresponding descriptors, but other descriptors may be used as well within the process described below. They are normalized with respect to their self scale and local orientation of gradient. Then the SURF description is quantized from 64 real values in $(-1, 1)$ into integer values in $[0, 255]$. This leads to a compact description for each keypoint in less than 80 bytes (including 64 bytes for the descriptor).

The image matching process includes two filtering step to drop keyframes of the database that are not close enough to the query. The first filtering step finds matching keypoints with respect to their appearance in a query-candidate couple of images. Valid keypoint matches are considered if they pass the test of relative nearest-neighbors proposed by D.Lowe in [39]. The images with the highest number of matches are top ranked.

The second filtering step selects within the previous results those that provides a similar geometrical configuration of keypoints in the query-candidate couple of images. We avoid considering complete homographies, preferring simple similarities that are much fastest to compute. This reduces the complexity of the exhaustive test of models for this geometrical confirmation. Hence, even a small set of matching keypoints between two images can lead to a fit. The final result list is composed of images having more than p keypoints fitting the geometrical model ($p \geq 5$).

This matching process can detect the co-occurrence of small objects in a query-candidate couple of images, leading to relative good precision for CBIR tasks similar to instance search or duplicate-detection.

CEA_Bag-of-visterm The Bag-of-Visual-Words (BoVW) approach [28, 29] is a state-of-the-art representation for visual content description used in image classification. Extended to image description, the usual BoVW design pipeline consists of learning a codebook from a large collection of local features extracted from a training dataset, then creating the global feature of visual signature through coding, pooling and spatial layout. Recent works addressing this problem [30, 31, 37] proved the importance of tuning each of these steps to improve scene classification and object recognition accuracy on different benchmarks.

The pipeline we used is as follows:

- *Local visual descriptors:* dense SIFTs of size d are extracted within a regular spatial grid and only one scale. The patch-size is fixed to 16×16 pixels and the step-size for dense sampling to 6 pixels;
- *Codebook:* a visual codebook of size 1024 is created using the K-means clustering method on a randomly selected subset of SIFTs from the training dataset.
- *Coding/pooling:* for coding the local visual descriptors SIFTS, we also fix the patch-size to 16×16 pixels and the step-size for dense sampling to 6 pixels. Then for the extracted visual descriptors associated to one image, we consider a neighborhood in the visual feature space of size 5 for local soft coding and the softness parameter β is set to 10. The max-pooling operation is performed to aggregate the obtained codes and a spatial pyramid decomposition into 3 levels $(1 \times 1, 2 \times 2, 3 \times 1)$ is adopted for the visual-signature. The weight is the same on each pyramid level.

Thus, the size of the visual-signature is equal to $1024 \times (1 + 2 \times 2 + 3 \times 1) = 8192$.

The **CEA_Bov_0** descriptor is built with SIFT extracted from the grayscale image ($d = 128$) while the **CEA_Bov_1** is built from color SUFT extracted from the Hue-Saturation-Value image ($d = 384$). For both descriptors, a L2 distance was used to compara a keyframe to a query.

Global features Global descriptors were used; some of them had a specific distance, else a L2 Minkowski one was used.

- **CEA_tlep** : a descriptor that is itself the concatenation of a Local Edge Pattern (LEP) descriptor (derived from [3]) and a color histogram, with a global normalisation on the 576 dimensions.
- **CEA_cime** : a compact histogram that count how many pixels are 4-connected according to their colors [40].
- **CEA_histo64** : a classic color histogram of size 64.
- **CEA_snow** : a RGB color histogram of size 125.
- **CEA_pigment** : a HSV color histogram of size 162.
- **CEA_projection** : the sum of the grayscale pixel values according to all lines and columns

Name	Formula
CombMAX	MAX(individual similarities)
CombSUM	SUM(individual similarities)
CombANZ	CombSUM / Number of non zero similarities
CombMNZ	CombSUM * Number of non zero similarities

Table 3: Definitions of different combination operators

These methods were also used in individual CEA LIST submission[17].

LABRI_BOW_<desc>_<clus>_<k>_[R][H][M] : several variations of the bag of visual words approach were used. **<desc>** refers to the descriptor used and was SIFT[39], RootSIFT[32] or SURF[27]. **<clus>** refers to the clustering method used for dictionary computation: **K** corresponds to K-means++, **A** correspond to Approximate K-means [33]. Approximate K-means optimizes the step of retrieving nearest neighbors between feature points and cluster centers by using an approximate nearest neighbor technique, such as FLANN [34]. A forest of multiple randomized kd-trees is built over the cluster centers at the beginning of each iteration. The size of the random forest was set to 8 kd-trees. **<k>** is the dimension of the dictionary, divided by 1000. **R**, if present, indicates that a spatial re-ranking step is used on the first 100 top-ranked results. **H**, if present, indicates that the clustering was done on the INRIA Holiday dataset[35] and not on the TRECVID dataset. **M**, if present, indicates that only descriptors for the object (inside the provided mask) were used for query. The complement of histogram intersection was used to compare signatures.

LISTIC Several methods used in the SIN task were also used in the INS task, with k=768 or 1024, and complement of histogram intersection for signatures comparison. Codebook computed on dataset of SIN task was used.

2.3 Fusion

Each described members method was used to produce intermediary results. Thus for each method (m) , we have a similarity $S_{m,e,i,k,c}$ between each example image (e) of each instance (i) and each keyframe (k) of each video clip (c). We have to fuse these similarities to obtain a similarity for an instance (i) and a clip (c).

We used a limited number combination operators: CombMAX, CombSUM, CombANZ, CombMNZ[41], defined in table 3.

We have tested two late fusion schemes. A truly late fusion scheme considers all the similarities $S_{m,e,i,k,c}$ at once. In a two-step late fusion scheme, we first merge the results for a given method (m), and then globally. Besides, weights can be used to give an asymmetric importance to the various intermediary results. Here, all intermediary results have been previously normalized. These two fusion schemes are described by the equations 1 and 2, where α_m and β_m are weights that sum to 1.

$$S_{i,c} = Comb_1(\alpha_m * S_{m,e,i,k,c}) \tag{1}$$

$$\begin{aligned} S_{i,c} &= Comb_2(\beta_m * S_{e,i,k,c}) \\ with\ S_{e,i,k,c} &= Comb_m(\alpha_m * S_{m,e,i,k,c}) \end{aligned} \tag{2}$$

A Combination operator will be noted $Comb[S]$ if applied to score, and $Comb[R]$ if applied to rank. We have tested several combination operators with these two fusion schemes, applied both to score and to rank. We have also tested with a limited combination of weights. The best results were obtained with the two-step fusion scheme. Moreover, as performance of various methods is not homogeneous, we tried to find the best combination operator and the similarity to use for each individual method, both for 2010 and 2011 queries and datasets. Theses choices are presented in table 4.

This year, we did not use any weight function on combinations: $\alpha_m = \beta_m = 1$.

We submitted the four following runs :

$$\begin{aligned} &Run4 = CombMAX[R](\\ &CombSUM[S](CEA\_Bov\_0), \\ &CombANZ[S](CEA\_Bov\_1), \\ &CombMAX[S](LABRI\_BOW\_SIFT\_A\_200\_H), \\ &CombMAX[S](LABRI\_BOW\_SIFT\_A\_200), \\ &CombMAX[S](LABRI\_BOW\_SURF\_A\_200), \\ &CombMAX[S](LABRI\_BOW\_SURF\_K\_16), \\ &CombMAX[S](LABRI\_BOW\_RootSIFT\_K\_16\_H), \\ &CombMAX[S](LISTIC\_*)) \end{aligned} \tag{3}$$

$$\begin{aligned} Run3 = CombMAX[R](& \\ & CombANZ[S](CEA\_*), \\ & CombMAX[S](LABRI\_*), \\ & CombMAX[S](LISTIC\_*)) \end{aligned} \tag{4}$$

Method	Best operator	MAP
CEA_markrs	CombSUM[S]	0.0224
CEA_Bov_0	CombMAX[S]	0.0356
CEA_Bov_1	CombMAX[S]	0.0023
CEA_tlep	CombMAX[R]	0.0111
CEA_cime	CombMAX[R]	0.0048
CEA_histo64	CombMAX[R]	0.0054
CEA_snow	CombMAX[S]	0.0212
CEA_pigment	CombMAX[R]	0.0039
CEA_projection	CombMAX[R]	0.0039
LABRI_BOW_SIFT_A_200_H	CombMAX[S]	0.0400
LABRI_BOW_SIFT_A_200	CombMAX[S]	0.0540
LABRI_BOW_SIFT_A_200_R	CombMAX[S]	0.0392
LABRI_BOW_SURF_A_200	CombMAX[S]	0.0328
LABRI_BOW_SURF_A_200_R	CombMAX[S]	0.0260
LABRI_BOW_SURF_A_200_RM	CombMAX[S]	0.0165
LABRI_BOW_SURF_K_16	CombMAX[S]	0.0246
LABRI_BOW_SURF_K_16_M	CombMAX[S]	0.0025
LABRI_BOW_SURF_K_16_R	CombMAX[S]	0.0224
LABRI_BOW_RootSIFT_K_16	CombMAX[S]	0.0252
LABRI_BOW_RootSIFT_K_16_R	CombMAX[S]	0.0333
LISTIC_768	CombMAX[R]	0.0068
LISTIC_1024	CombMAX[R]	0.0072
LISTIC_retina_768	CombMAX[S]	0.0278
LISTIC_retina_1024	CombMAX[S]	0.0287
LISTIC_retinaMasking_768	CombMAX[S]	0.0198
LISTIC_retinaMasking_1024	CombMAX[S]	0.0183
LISTIC_mcRetinaMasking_1024	CombMAX[S]	0.0230
LISTIC_mcRetinaMasking_d_1024	CombMAX[S]	0.0081

Table 4: Best combination operator with similarity type used for each individual methods, and corresponding MAP on 2013 dataset

$$\begin{aligned}
&Run2 = CombMAX[R](\\
&CombSUM[S](CEA\_markrs),\\
&CombSUM[S](CEA\_Bov\_0),\\
&CombANZ[S](CEA\_Bov\_1),\\
&CombMAX[S](LABRI\_BOW\_SIFT\_A\_200),\\
&CombMAX[S](LABRI\_BOW\_SURF\_A\_200\_R),\\
&CombMAX[S](LISTIC\_retina\_1024),\\
&CombMAX[S](LISTIC\_retinaMasking\_1024))
\end{aligned} \tag{5}$$

$$\begin{aligned}
&Run1 = CombMAX[R](\\
&CombSUM[S](CEA\_markrs),\\
&CombSUM[S](CEA\_snow),\\
&CombSUM[S](CEA\_Bov\_0),\\
&CombANZ[S](CEA\_Bov\_1),\\
&CombMAX[S](LABRI\_BOW\_SIFT\_A\_200\_H),\\
&CombMAX[S](LABRI\_BOW\_SIFT\_A\_200),\\
&CombMAX[S](LABRI\_BOW\_SURF\_A\_200),\\
&CombMAX[S](LABRI\_BOW\_SURF\_A\_200\_R),\\
&CombMAX[S](LABRI\_BOW\_SURF\_K\_16),\\
&CombMAX[S](LABRI\_BOW\_SURF\_K\_16\_R),\\
&CombMAX[S](LABRI\_BOW\_RootSIFT\_K\_16),\\
&CombMAX[S](LABRI\_BOW\_RootSIFT\_K\_16\_R),\\
&CombMAX[S](LISTIC\_*))
\end{aligned} \tag{6}$$

Run4 uses all the BoW methods. Run3 uses all the available descriptors. Run2 uses a minimum number of descriptors. Run1 is equaivalent to run4 completed with CEA_markrs, CEA_snow and all LABRI descriptors with re-ranking LABRI_BOW_*_R.

There was an error in our submitted runs, and thus results of evaluation by NIST are not representative of the performance of the method.

3 Data sharing

As last year, we propose to reuse and extend the organization that has been developed over five years within the members of the IRIM project of the French ISIS national Research Group (see [15] and section 1 of this paper). It is based on a limited number of simple data formats and on a (quite) simple directory organization. It also comes with a few scripts and procedures as well as with some sections for reporting intermediate results. The supporting structure is composed of a wiki (http://mrim.imag.fr/trecvid/wiki) and a data repository (http://mrim.imag.fr/trecvid/sin12). The wiki can be accessed using the TRECVid 2013 active participant username and password and the data repository can be accessed using the TRECVid 2013 IACC collection username and password.

A general rule about the sharing of elements is that:

- any group can share any element he think could be useful to others with possibly an associated citation of a paper describing how it was produced;
- any group can use any element shared by any other group provided that this other group is properly

cited in any paper presenting results obtained using the considered element,

exactly as this was the case in the previous years for the shared elements like shot segmentation, ASR transcript or collaborative annotation. Groups sharing elements get "rewarded" via citations when their elements are used.

Shared elements can be for instance: shot or key frame descriptors, classification results, fusion results. For initiating the process, most IRIM participants agreed to share their descriptors. Most classification and fusion results obtained are also shared. These are available on the whole 2010-2015 TRECVID SIN collection. Descriptors, classification scores or fusion results from other TRECVid particpants are most welcome. See the wiki for how to proceed.

4 Acknowledgments

This work has been carried out in the context of the IRIM (Indexation et Recherche d'Information Multimédia) of the GDR-ISIS research network from CNRS.

Experiments presented in this paper were carried out using the Grid'5000 experimental testbed, being developed under the INRIA ALADDIN development action with support from CNRS, RENATER and several Universities as well as other funding bodies (see https://www.grid5000.fr).

The authors also wish to thank Florent Perronnin from XRCE for providing descriptors based on classification scores from classifiers trained on ILSVRC/ImageNet data.

References

[1] A. Smeaton, P. Over and W. Kraaij, Evaluation campaigns and TRECVid, In *MIR'06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pp321-330, 2006.

[2] P. Over, G. Awad, J. , B. Antonishek, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A. Smeaton and G. Quénot, TRECVID 2013 – An Overview of the Goals, Tasks, Data, Evaluation Mechanisms, and Metrics In *Proceedings of the TRECVID 2013 workshop*, Gaithersburg, USA, 20-22 Nov. 2013.

[3] Y.-C. Cheng and S.-Y. Chen. Image classification using color, texture and regions. In *Image and Vision Computing,* 21:759-776, 2003.

[4] P.H. Gosselin, M. Cord, Sylvie Philipp-Foliguet. Combining visual dictionary, kernel-based similarity and learning strategy for image category retrieval. In *Computer Vision and Image Understanding,* Special Issue on Similarity Matching in Computer Vision and Multimedia. Volume 110, Issue 3, Pages 403-441, 2008.

[5] M. Redi and B. Merialdo. Saliency moments for image categorization, In *ICMR 2011, 1st ACM International Conference on Multimedia Retrieval,* April 17-20, 2011, Trento, Italy.

[6] D. Picard and P.H. Gosselin. Efficient image signatures and similarities using tensor products of local descriptors, In *Computer Vision and Image Understanding,* Volume 117, Issue 6, Pages 680-687, 2013.

[7] A. Oliva and A. Torralba, Modeling the shape of the scene: A holistic representation of the spatial envelope, In *International Journal of Computer Vision,* vol 42, number 3, pages 145-175, 2001.

[8] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. A comparison of color features for visual concept classification. In *ACM International Conference on Image and Video Retrieval*, pages 141–150, 2008.

[9] A. Benoit, A. Caplier, B. Durette, and J. Herault. Using Human Visual System Modeling for Bio-inspired Low Level Image Processing, In *Computer Vision and Image Understanding,* vol. 114, no. 7, pp. 758-773, 2010.

[10] S. T. Strat, A. Benoit, P. Lambert and A. Caplier, Retina Enhanced SURF Descriptors for Spatio-Temporal Concept Detection, In *Multimedia Tools ans Applications,* to appear, 2012.

[11] S. Paris, H .Glotin, Pyramidal Multi-level Features for the Robot Vision@ICPR 2010 Challenge, In *20th International Conference on Pattern Recognition,* pp.2949-2952, 2010

[12] Jorge Sánchez, Florent Perronnin, Thomas Mensink, Jakob Verbeek Image Classification with the Fisher Vector: Theory and Practice In *International Journal of Computer Vision,* Volume 105, Issue 3, pp 222-245, December 2013.

[13] B. Safadi, G. Quénot. Evaluations of multi-learners approaches for concepts indexing in video documents. In *RIAO,* Paris, France, April 2010.

[14] Georges Quénot. *KNNLSB: K Nearest Neighbors Linear Scan Baseline*, 2008. Software available at `http://mrim.imag.fr/georges.quenot/freesoft/knnlsb/index.html`.

[15] Ballas et al. IRIM at TRECVID 2012: Semantic Indexing and Multimedia Instance Search, In *Proceedings of the TRECVID 2011 workshop*, Gaithersburg, USA, 26-28 Nov. 2012.

[16] Safadi et al. Quaero at TRECVID 2013: Semantic Indexing and Collaborative Annotation, In *Proceedings of the TRECVID 2013 workshop*, Gaithersburg, USA, 20-22 Nov. 2013.

[17] Nicolas Ballas, Benjamin Labbé, Hervé Le Borgne, Aymen Shabou CEA LIST at TRECVID 2013: Instance Search, In *Proceedings of the TRECVID 2013 workshop*, Gaithersburg, USA, 20-22 Nov. 2013.

[18] Stéphane Ayache and Georges Quénot, Video Corpus Annotation using Active Learning, In 30th European Conference on Information Retrieval (ECIR'08), Glasgow, Scotland, 30th March - 3rd April, 2008.

[19] J.L. Gauvain, L. Lamel, and G. Adda. The LIMSI Broadcast News In *Transcription System. Speech Communication*, 37(1-2):89-108, 2002.

[20] D. Gorisse et al., IRIM at TRECVID 2010: High Level Feature Extraction and Instance Search. In *TREC Video Retrieval Evaluation workshop*, Gaithersburg, MD USA, November 2010.

[21] Alice Porebski, Color texture feature selection for image classification. Application to flaw identification on decorated glasses printing by a silk-screen process. *Phd thesis,* Université Lille 1, Sciences et Technologies, Nov. 2009

[22] V. D. Blondel and J. Guillaume and R. Lambiotte and E. Lefebvre, Fast Unfolding of Community Hierarchies in Large Networks, In *Computing Research Repository,* abs/0803.0, 2008.

[23] B. Safadi, G. Quénot. Re-ranking by Local Rescoring for Video Indexing and Retrieval, *CIKM 2011: 20th ACM Conference on Information and Knowledge Management,* Glasgow, Scotland, oct 2011.

[24] Bahjat Safadi, Georges Quénot. Descriptor Optimization for Multimedia Indexing and Retrieval. *CBMI 2013, 11th International Workshop on Content-Based Multimedia Indexing,* Veszprem, HUNGARY, jun 2013.

[25] Strat, S.T. and Benoit, A. and Lambert, P., Retina enhanced SIFT descriptors for video indexing, *CBMI 2013, 11th International Workshop on Content-Based Multimedia Indexing,* Veszprem, HUNGARY, jun 2013.

[26] Abdelkader Hamadi, Georges Quénot, Philippe Mulhem. Conceptual Feedback for Semantic Multimedia Indexing, *CBMI 2013, 11th International Workshop on Content-Based Multimedia Indexing,* Veszprem, HUNGARY, jun 2013.

[27] H. Bay, Herbert, T.Tuytelaars,and L. Van Gool. SURF: Speeded Up Robust Features, In *ECCV 2006*, pp 404-417, 2006.

[28] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *ICCV'03*, volume 2, pages 1470–1477, 2003.

[29] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision, ECCV*, pp. 1–22, 2004.

[30] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2169–2178, 2006.

[31] L. Liu, L. Wang, and X. Liu, "In Defense of Soft-assignment Coding," in *IEEE International Conference on Computer Vision*, 2011.

[32] R. Arandjelović, A. Zisserman. Three things everyone should know to improve object retrieval, In *CVPR 2012*, 2012.

[33] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman Object retrieval with large vocabularies and fast spatial matching In *CVPR 2007*, 2007.

[34] M. Muja, D. G. Lowe. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration, In *VISAPP'09*, 2009.

[35] Hamming Embedding and Weak geometry consistency for large scale image search, In *ECCV 2008*, 2008.

[36] S. T. Strat, A. Benoit, H. Bredin, G. Quénot and P. Lambert. Hierarchical Late Fusion for Concept Detection in Videos. In *ECCV workshop on Information Fusion in Computer Vision for Concept Recognition,* Firenze, Italy, 13 Oct. 2012.

[37] A. Shabou and H. Le Borgne. Locality-constrained and spatially regularized coding for scene categorization, In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3618–3625, 2012.

[38] C. Zhu, C.-E. Bichot, L. Chen. Color orthogonal local binary patterns combination for image region description. In *Technical Report, LIRIS UMR5205 CNRS*, Ecole Centrale de Lyon.

[39] D.G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004

[40] R. O. Stehling, M. A. Nascimento, and A.X. Falcão. A compact and efficient image retrieval approach based on border/interior pixel classification In *11th International Conference on Information and Knowledge Management* 2002

[41] E. Fox and J. Shaw Combination of Multiple searches In *Proceedings of the 2nd Text Retrieval Conference* Gaithersburg, USA, 1994

[42] G. Csurka and S. Clinchant An empirical study of fusion operators for multimodal image retrieval In *10th Workshop on Content-Based Multimedia Indexing* Annecy, France, 2012

Table 1: Performance of the classifier and descriptor combinations

Descriptor	dim	α_1	Unit length	Pdim	α_2	MAP dev	MAP test
CEALIST/tlep_576	576	0.424	-	120	0.719	0.1237	0.0972
CEALIST/bov_dsiftSC_8192	8192	0.700	-	292	0.575	0.1486	0.1227
CEALIST/bov_dsiftSC_21504	21504	0.600	-	364	0.714	0.1557	0.1547
ETIS/labm1x1x256	256	0.334	-	132	0.641	0.1096	0.0813
ETIS/labm1x1x512	512	0.340	-	178	0.712	0.1115	0.0832
ETIS/labm1x1x1024	1024	0.345	-	208	0.742	0.1122	0.0836
ETIS/labm1x3x256	768	0.338	-	208	0.633	0.1213	0.1007
ETIS/labm1x3x512	1536	0.351	-	310	0.651	0.1215	0.1010
ETIS/labm1x3x1024	3072	0.380	-	333	0.720	0.1211	0.1008
ETIS/labm2x2x256	1024	0.324	-	240	0.577	0.1173	0.0960
ETIS/labm2x2x512	2048	0.353	-	308	0.621	0.1175	0.0954
ETIS/labm2x2x1024	4096	0.378	-	324	0.739	0.1184	0.0970
ETIS/qwm1x1x256	256	0.450	-	144	0.742	0.0982	0.0735
ETIS/qwm1x1x512	512	0.437	-	166	0.718	0.1044	0.0838
ETIS/qwm1x1x1024	1024	0.449	-	182	0.724	0.1088	0.0900
ETIS/qwm1x3x256	768	0.421	-	205	0.696	0.1134	0.1000
ETIS/qwm1x3x512	1536	0.413	-	230	0.725	0.1193	0.1089
ETIS/qwm1x3x1024	3072	0.410	-	253	0.666	0.1225	0.1138
ETIS/qwm2x2x256	1024	0.431	-	203	0.720	0.1098	0.0918
ETIS/qwm2x2x512	2048	0.427	-	229	0.771	0.1150	0.1007
ETIS/qwm2x2x1024	4096	0.423	-	277	0.788	0.1184	0.1068
ETIS/vlat_hog3s4-6-8-10_dict64_4096	4096	0.875	L_1	4096	1.000	0.1624	0.1801
EUR/sm462	462	0.167	-	215	0.380	0.1269	0.0949
LABRI/faceTracks16x16	256	0.240	-	210	0.480	0.0180	0.0113
LIF/percepts_1_1_1_15	15	0.495	-	15	0.735	0.0860	0.0402
LIF/percepts_2_2_1_15	60	0.470	-	60	0.669	0.1056	0.0676
LIF/percepts_5_3_1_15	225	0.623	-	148	0.575	0.1092	0.0722
LIF/percepts_10_6_1_15	900	0.619	-	169	0.381	0.1092	0.0710
LIF/percepts_20_13_1_15	3900	0.550	-	193	0.420	0.1093	0.0765
LIG/gab40	40	0.629	-	40	0.629	0.0809	0.0322
LIG/h3d64	64	0.286	-	52	0.813	0.0916	0.0577
LIG/hg104	104	0.348	-	89	0.700	0.1148	0.0816
LIG/opp_sift_har_1000	1000	0.513	-	103	0.782	0.1194	0.0946
LIG/opp_sift_dense_1000	1000	0.489	-	206	0.466	0.1276	0.1104
LIG/opp_sift_har_unc_1000	1000	0.331	-	116	0.592	0.1262	0.1072
LIG/opp_sift_dense_unc_1000	1000	0.415	-	303	0.384	0.1354	0.1218
LIG/opp_sift_har_1024_fu8	1024	0.409	-	170	0.324	0.1264	0.1013
LIRIS/MFCC_4096	4096	0.426	L_2	200	1.000	0.0584	0.0241
LIRIS/OCLBP_4096	4096	0.374	L_2	167	0.681	0.1122	0.1156
LISTIC/SIFT_768	768	0.488	-	271	0.435	0.1257	0.1247
LISTIC/SIFT_1024	1024	0.444	-	272	0.436	0.1274	0.1263
LISTIC/SIFT_2048	2048	0.912	-	175	0.420	0.1115	0.0897
LISTIC/SIFT_retina_768	768	0.495	-	178	0.502	0.1266	0.1108
LISTIC/SIFT_retina_1024	1024	0.504	-	204	0.515	0.1288	0.1123
LISTIC/SIFT_retina_2048	2048	0.768	-	134	0.455	0.1208	0.1050
LISTIC/SIFT_retinaMasking_768	768	0.417	-	126	0.422	0.1250	0.1115
LISTIC/SIFT_retinaMasking_1024	1024	0.400	-	136	0.399	0.1274	0.1149
LISTIC/SIFT_retinaMasking_2048	2048	0.434	-	171	0.187	0.1013	0.0732
LISTIC/SIFT_multiChannelsRetinaMasking_1024	1024	0.398	-	123	0.369	0.1287	0.1199
LISTIC/SIFT_multiChannelsRetinaMaskingDual1024_2048	2048	0.438	-	160	0.258	0.1291	0.1298
LISTIC/expe6_trajectories_7_256	256	0.592	-	55	0.820	0.0651	0.0735
LISTIC/expe6_trajectories_13_1024	1024	0.542	-	64	0.849	0.0726	0.0886
LISTIC/expe6_trajectories_14_1024	1024	0.547	-	64	0.849	0.0724	0.0886
LISTIC/expe6_trajectories_69_384	384	0.451	-	72	0.930	0.0657	0.0632
LISTIC/expe6_trajectories_74_256	256	0.469	-	100	0.945	0.0547	0.0636
XEROX/ilsvrc2010	1000	0.575	-	592	0.650	0.1710	0.2190
XEROX/imagenet10174	10174	0.200	-	1024	0.650	0.1721	0.2258