LIG and LIRIS at TRECVID 2008: High Level Feature Extraction and Collaborative Annotation

Stéphane Ayache and Georges Quénot

Abstract

This paper describes our participations of LIG and LIRIS to the TRECVID 2008 High Level Features detection task. We evaluated several fusion strategies and especially rank fusion. Results show that including as many low-level and intermediate features as possible is the best strategy, that SIFT features are very important, that the way in which the fusion from the various low-level and intermediate features does matter, that the type of mean (arithmetic, geometric and harmonic) does matter. LIG and LIRIS best runs respectively have a Mean Inferred Average Precision of 0.0833 and 0.0598; both above TRECVID 2008 HLF detection task median performance.

LIG and LIRIS also co-organized the TRECVID 2008 collaborative annotation. 40 teams did 1235428 annotations. The development collection was annotated at least once at 100%, at least twice at 37.6%, at least three times at 3.99% and at least four times at 0.06%. Thanks to the active learning and active cleaning used approach, the annotations that were done multiple times were those for which the risk of error was maximum.

1 Introduction

This paper describes the participations of LIG and LIRIS to the TRECVID 2008 [12] High Level Features (HLF) detection task [13]. The framework for HLF detection is based on network of operators [1][3]. We made use of INRIA LEAR SIFT bag of features descriptors [10][15][7]. Our main interest to this year's TRECVID challenge was to compare various ways to combine low-level features and/or systems output. Our system for video shots indexing is fully based on key frames analysis; our generic classification system follows a classical pipeline architecture which includes low-level features extractor, mid-level semantic classification and fusion modules. We used one or more keyframes per shot according to the shot and subshot segmentation provided by Fraunhofer HHI [11]. The following sections present the considered features, the fusion strategies and the submitted runs.

2 Visual analysis

We performed visual analysis at several level of granularity from global to fine blocks analysis, as well as various semantic levels. Our low-level feature extractors first split images on overlapped blocks to form a grid of N×M blocks. For our submissions, we chose N and M such as we obtained a satisfying trade-off between classification performance and time computing. Finally, those values depends on which feature is considered. The analysis first treats each key frame to extract several feature vectors, secondly merge them using standard early or late fusion schemes, or a combination of them and then merge key frames to assign a score to each shot.

2.1 Low-level features

At global level, we consider classical color and texture features. Color is represented by a 3-dimensional histogram on RGB space. We discretize the color space to form a 4×4×4 bins histogram. Texture information is described with Gabor bank of filters; we used 8 orientations and 5 scales. Finally, global features are normalized and concatenated on a 104 dimensions vector. We also extracted color and texture features at block levels, features obtained from each block are then concatenated to form a rich description of key frames:

Color (1): is represented by 3×3×3 3D histogram on a grid of 8×6 blocks. The overall local color feature vector has 1296 dimensions.

Color (2): is represented by the first two moments on a grid of 8×6 blocks. This local color feature vector has 432 dimensions.

Edge Direction Histogram: is computed on a grid of 4×3 blocks. Each bin is defined as the sum of the magnitude gradients from 50 orientations. Thus, overall EDH feature has 600 dimensions. EDH feature is known to be invariant to scale and translation.

Local Binary Pattern: [9] is computed on grid of 2×2 blocks, leading to a 1024 dimensional vector. The LBP operator labels the pixels of an image by

thresholding the 3×3-neighborhood of each pixel with the center value and considering the result as a decimal number. LBP is known to be invariant to any monotonic change in gray level.

2.2 Feature on interest points

One of the more relevant feature for visual indexing is the SIFT descriptor combined with a "bag of words" representation. The SIFT descriptor [8] is describes the local shape of points of interest using edge histograms. To make the descriptor invariant, the interest region is divided into a 4×4 grid and every sector has its own edge direction histogram (8-bin). We used a codebook of 1000 visual word, provided by INRIA LEAR.

2.3 Semantic feature

This feature aims at modeling co-occurrence between high-level features using a "bag of concepts" approach. First, we consider each block from key frames which are relevant for a concept, as relevant for this concept too. This is a very strong assumption but it could be reasonable depending upon the concepts. Thus, we use existing concepts annotations (from a part of the learning set) at global level, to train SVM classifiers at the blocks level, where blocks are represented with moments color and edge direction histogram features. Then blocks of key frames are classified using models of all the concepts, leading to $nb\_blocks \times nb\_concepts$ classification scores per key frame. The final semantic feature is defined by the sum of scores on nb_blocks for each concepts, leading to a $nb\_concepts$ dimensional feature.

3 HLF extraction framework

3.1 Supervised classification

We use Support Vector Machine [6] as binary classifier for supervised classification of HLF with RBF Kernel and probabilistic output scores. We obtain SVM parameter by testing all combination of parameters (σ, c) with 5-fold cross validation. The models are learned using standard annotation provided by LIG Collaborative Annotation by considering up to 800 randomly positive examples and twice as negative randomly selected.

3.2 Early and Late fusion

We merged our various features with combinations of early and late fusion schemes. While the early fusion proceeds in the feature space, the late fusion combines classification scores. Combination of those two schemes is possible when more than two features is available and yields much flexibility on the way to merge features. For example we can combine features with early fusion once then combine with others feature with late fusion. We know from [4] that such combinations outperform, for some concepts, classical early and late fusion.

3.2.1 Early fusion

Early fusion is basically defined by a simple concatenation of the features from the various modalities. While the number of extracted features depends upon the modalities and the type of the features, an early fusion scheme based on simple vector concatenation is much affected by the vector which has the highest number of inputs. Such fusion should have an impact on the classification, especially with a RBF kernel which is based on Euclidian distance. Thus, we normalize each entry of the concatenated vector so that the average norm of each source vector is about the same. The normalization formula is:

$$x_i\prime = \frac{x_i - min_i}{(max_i - min_i) \times \sqrt{Card(x_i)}}$$

where x_i is an input of the feature vector x, min_i and max_i are respectively the minimum and maximum value of the i^{th} input among the training samples and $Card(x_i)$ is the number of dimensions of the source vector of x_i.

3.2.2 Late fusion

The late fusion scheme first classifies each concept using individual modalities and then merges the scores on top of those classifiers. A second layer of classifier can be considered but it does not always conduct to expected performance as it is needed to split training set to learn this classifier while avoiding over fitting. We chose here to use a usual function to combine modalities with neither parameters nor learning phase:

$$v = \frac{1}{N} \sum_{i=0}^{N} log(v_i)$$

where N is the number of modalities and v_i is the score for the i^{th} modality.

3.3 Rank fusion

Rank fusion is a specific case of late fusion. Its principle is to assign to each shot and for each source a score which is equal to its rank according to a classifier that uses this source and then to build a global score for each shot as a mean of the scores (ranks). This global score is then used to re-rank the shots. Three variants of the rank fusion correspond to the use of the arithmetic

(classical) mean, the geometric mean and the harmonic mean. The harmonic mean is often considered as the best choice for rank fusion. All three variants can be implemented with weights for the various sources. We tried the three variants with the expected best results from LIG and LSIS in order to compare them.

4 Experiments

4.1 Runs description

The LIG and LIRIS runs are described in a Functional Programing (FP) Style [5] since it was observed that FP expressions are a powerful way for describing networks of operators [14] as those we are using in our approach [1].

LIG runs:

LIG_1 = LATE(
  EARLY(Local-Color , EDH , Global-Features),
  EARLY(Semantic , EDH , Global-Features),
  EARLY(Local-Color , SIFT , Global-Features)
)

LIG_2 = LATE(
  EARLY(Local-Color , EDH , Global-Features),
  EARLY(Semantic , Global-Features),
  EARLY(Local-Color , SIFT)
)

LIG_3 = EARLY(Local-Color, EDH, Global-Features, Semantic, SIFT, LBP)

LIG_4 = LATE(
  EARLY(Local-Color, EDH, Global-Features),
  EARLY(Semantic, Global-Features)
)

LIG_5 = EARLY(Global-Features, Semantic)

LIG_6 = EARLY(Local-Color, EDH, Global-Features)

These were empirically determined as good combinations from tests on the development collection.

LIRIS runs:

LIRIS_1 = EARLY(Global-Features, Semantic, EDH)

LIRIS_2 = HarmonicRANK (
  EARLY(Global-Features, Semantic, EDH),
  (Semantic),
  (EDH),
  (LBP)
)

LIRIS_3 = ArithmeticRANK (
  EARLY(Global-Features, Semantic, EDH),
  (Semantic),
  (EDH),
  (LBP)
)

LIRIS_4 = Semantic

LIRIS_5 = EDH

LIRIS_6 = LBP

4.2 Results

We list in table 1 the submitted runs of LIG and LIRIS groups. For more details for this run, please refer to the corresponding TRECVID paper. We show in bold the two LIG best runs and the LIRIS best run. The metrics are the Mean Inferred Average Precision on the 20 target concepts (HLFs).

LIG_1	**0.0833**	LIRIS_1	0.0525
LIG_2	**0.0793**	LIRIS_2	0.0267
LIG_3	0.0785	**LIRIS_3**	**0.0598**
LIG_4	0.0659	LIRIS_4	0.0254
LIG_5	0.0423	LIRIS_5	0.0221
LIG_6	0.0568	LIRIS_6	0.0189

Table 1: results fo LIG and LIRIS HLF submissions

4.3 Discussion

LIG compared various combinations of features both in the choice of the features and the way they are combined (see above for example of non-trivial combinations for LIG_1 and LIG_2). Their relative performances were judged on the development set and the run priority was defined according to it. The run performance ordering was quite well predicted (only LIG_5 and LIG_6 were swapped).

LIRIS conducted some runs using only a single feature (LIRIS_4, LIRIS_5 and LIRIS_6). As expected, the performance using single features is low and varies according to the feature. LIRIS also tried arithmetic and harmonic mean based rank fusion. Both do better than each of the single feature but, surprisingly, the arithmetic mean leads to much better results than the harmonic mean (usually considered as better).

2007	Annotated	% Annotated	Negative	Skipped	Positive	% Positive
Pass 1	641223	82.7	578299	13163	49761	7.76
Pass 2	46864	6.05	11904	7478	27482	58.6
Pass 3	21987	2.84	9383	4040	8564	39.0
Pass 4	1492	0.19	324	940	228	15.3
Synthesis	641223	82.7	578683	15348	47192	7.36

2008	Annotated	% Annotated	Negative	Skipped	Positive	% Positive
Pass 1	872320	100.0	830087	30608	11625	1.33
Pass 2	327744	37.6	310815	7886	9043	2.76
Pass 3	34810	3.99	26746	4772	3292	9.46
Pass 4	554	0.06	396	50	108	19.5
Synthesis	641223	82.7	578683	15348	47192	7.36

Table 2: Annotation statistics by pass, average on all concepts.

5 Collaborative annotation

The active learning based collaborative annotation system was described in details in [2] and ecir08. The main evolution between the 2007 and 2008 versions consists in an improvement of the active cleaning feature.

Another major difference is that for a comparable number of keyframes × concepts to annotate (21532 × 36 in 2007 and 20 × 22084 in 2008), more participants did more annotations and 100% of the corpus was annotated at least once in 2008 against 82.7% in 2007.

40 (respectively 32) teams participated to the 2008 (respectively 2007) TRECVID collaborative annotation effort and produced a total of 1235428 (respectively 711566) annotations. Table 2 gives some statistics on these annotations. "Pass 1", "Pass 2", "Pass 3" and "Pass 4" corresponds to the number of annotations that were done respectively at least once, at least twice, at least three times and at least four times for a given key frame × concept. The "Synthesis" correspond to the global annotation when a "majority" rule is applied if there is more than one annotation for a key frame × concept.

Figure 1 shows how the collaborative effort was spread over time for years 2007 and 1008. Horizontal units correspond to the days of May 2007 or 2008 between 1 and 31 included and extrapolated outside. The 2008 annotation was more spread over time but was shorter in total. The peak of daily annotations is of a bit less than 100,000 in both cases.

6 Conclusion

LIG and LIRIS evaluated several fusion strategies including rank fusion for LIRIS. Results show that including as many low-level and intermediate features as possible is the best strategy, that SIFT features are very important, that the way in which the fusion from the various low-level and intermediate features does matter, and that the type of mean (arithmetic, geometric and harmonic) does matter. LIG and LIRIS best runs respectively have a Mean Inferred Average Precision of 0.0833 and 0.0598; both above TRECVID 2008 HLF detection task median performance.

LIG and LIRIS also co-organized the TRECVID 2008 collaborative annotation. 40 teams did 1235428 annotations. The development collection was annotated at least once at 100%, at least twice at 37.6%, at least three times at 3.99% and at least four times at 0.06%. Thanks to the active learning and active cleaning used approach, the annotations that were done multiple times were those for which the risk of error was maximum.

References

[1] S. Ayache and G. Quénot. Image and video indexing using networks of operators. *EURASIP Journal on Image and Video Processing*, Volume 2007, 2007.

[2] S. Ayache and G. Quénot. TRECVID 2007 Collaborative Annotation using Active Learning. In *Proc. of TRECVID Workshop*, 5-6 Nov. 2007.

[3] S. Ayache and G. Quénot. LIG and LIRIS at TRECVID 2008: High Level Feature Extraction and Collaborative Annotation. In *Proc. of TRECVID Workshop*, 17-18 Nov. 2008.

[4] S. Ayache, G. Quénot, J. Gensel, and S. Satoh. Using topic concepts for semantic video shots classification. In *International Conference on Image and Video Retrieval (CIVR'06)*, 13-15 Jul. 2006.

[5] J. Backus. Can programming be liberated from the von neumann style?: a functional style and its algebra of programs. *Commun. ACM*, 21(8):613–641, August 1978.

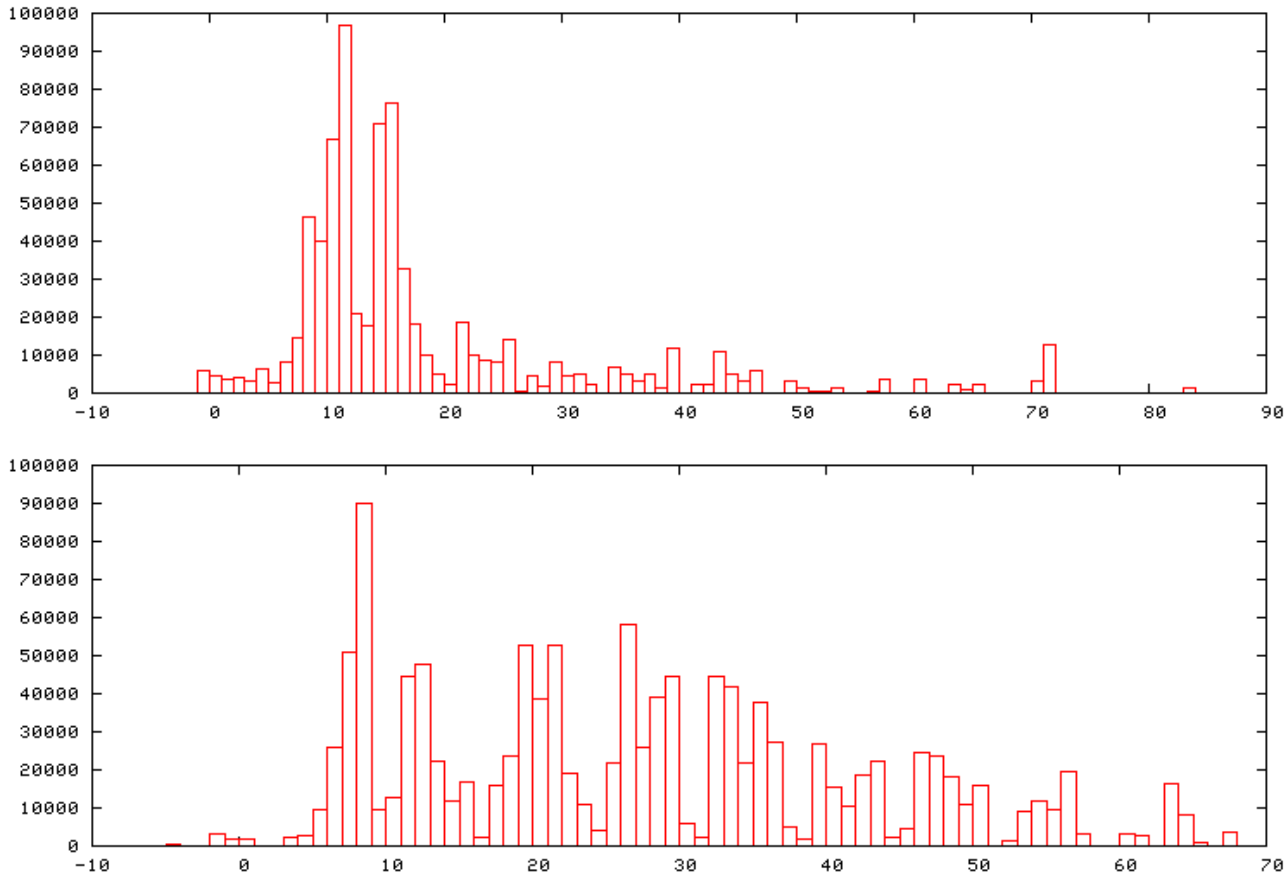

Figure 1: Daily annotations in the collaborative annotation project (GMT time, May 2007 or 2008 days).

[6] C.-C. Chang and C.-J. Lin. *LIBSVM: A Library for Support Vector Machines*, 2001. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[7] M. Douze, A. Gaidon, H. Jegou, M. Marszalek, and C. Schmid. INRIA-LEAR's Video Copy Detection System. In *Proc. of TRECVID Workshop*, 17-18 Nov. 2008.

[8] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[9] T. Maenpaa, M. Pietikainen, and T. Ojala. Texture classication by multi-predicate local binary pattern operators. In *Proc. 15th International Conference on Pattern Recognition*, pages 951–954, 2000.

[10] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[11] C. Petersohn. Fraunhofer HHI at TRECVID 2004: Shot Boundary Detection System. In *Proc. of TRECVID Workshop*, 15-16 Nov. 2004.

[12] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.

[13] A. F. Smeaton, P. Over, and W. Kraaij. High-Level Feature Detection from Video in TRECVid: a 5-Year Retrospective of Achievements. In A. Divakaran, editor, *Multimedia Content Analysis, Theory and Applications*. Springer Verlag, Berlin, 2009.

[14] B. Zavidovique, J. Sérot, and G. M. Quénot. Massively parallel data flow computer dedicated to real-time image processing. *Integr. Comput.-Aided Eng.*, 4(1):9–29, 1997.

[15] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: a comprehensive study. *International Journal of Computer Vision*, 73(2):213–238, jun 2007.