High-Level Feature Detection with Forests of Fuzzy Decision Trees combined with the RankBoost Algorithm

Christophe Marsala, Marcin Detyniecki, Nicolas Usunier, Massih-Reza Amini

Université Pierre et Marie Curie-Paris6
CNRS UMR 7606, LIP6,
104 avenue du Président Kennedy,
Paris, F-75016, France

{Christophe.Marsala,Marcin.Detyniecki,Nicolas.Usunier,Massih-Reza.Amini}@lip6.fr

Abstract

In this paper, we present the methodology we applied in our submission to the NIST TRECVID'2007 evaluation. We participated in the High-level Feature Extraction task. Our approach is based on the use of a Forest of Fuzzy Decision Trees combined with the RankBoost algorithm.

1 Structured Abstract - Summary

Here we present the contribution of the University of Paris 6 at TRECVID 2007 [6]. It concerns only the High-Level Feature Extraction task. The approach focuses on the use of Forests of Fuzzy Decision Trees (FFDT) that can be possibly combined with the RankBoost algorithm, and is based on a rather simple image description.

In the following, we start with a short summary of the used method and starting from Section 3, our approach is detailed. First, we describe the particularities of our set of descriptors. Then we explain how the training (Section 4) and classification (Section 5) was performed. Before concluding, the submitted runs are discussed in details (Section 6).

1.1 Brief Description of the Submitted Run

Here is the general information about the submitted run:

The task	High-Level Feature Extraction.
Type	A - system trained only on common TRECVID development collection data: annotations and truth data publically available to all participants.
Data	- XML files that provide the time-codes of each shot (Master shot references by [5]), - All the devel image files (the devel keyframes), - Annotations files for the devel keyframes.
Pre-treat.	- Each keyframe was segmented into 5 overlapping regions (see Section 3.1), - An HSV histogram was computed for each region (see Section 3.1), - Temporal information about shots was extracted from the XML files. (see Section 3.2).
Training	- A Forest of Fuzzy Decision Trees (FFDT) was constructed and trained from the Devel data set (see Section 4). - In one run, the RankBoost algorithm was settled from the Devel data set to optimized the ranking of the Devel data set according to the classification degrees provided by the FDT of a forest.
Ranking	- The Forest of Fuzzy Decision Trees (FFDT) was used to classify shots from the Test data set (see Section 5.4). - In one run, the RankBoost algorithm was used to rank test shots.

1.2 Comments on the Run

1.2.1 Relative Contribution of each Component

Visual Information Descriptors . We choose to segment the keyframes into a set of rectangular regions and work on their color description. By doing so, important descriptors could be isolated and, thus, could help the learning algorithm to focus on the discriminative variables. Moreover, more complementary visual descriptors should be added in order to enhance the possibilities of choice of the learning algorithm (FDT) for its decisions.

Video Information Descriptors . We chose to include temporal information brought by the shot's position in the video, its duration and temporal information about the keyframes.

Training (Forest of Fuzzy Decision Trees) . The use of decision trees enables us to automatically discover the discriminating features. Moreover, the fuzzy logic theory provides a more robust treatment of numerical values of the descriptors. In fact, we have soft decisions avoiding any threshold effects.

Ranking . Two methodologies have been proposed here. On the one hand, the forest of FDT was used to rank shots, on the other hand, the RankBoost algorithm was optimized this ranking. Here again, the fuzzy logic theory implies a robustness when handling numerical values. Moreover, it enables us to obtain a degree for each feature, for each keyframe [3]. Since we have a Forest of Decision Trees, we obtain a set of degrees (decisions) for each keyframe. The final degree for each shot, used for the ranking, is an aggregation of the individual decisions. Afterwards, in one run, the final degrees are ranked with the RankBoost algorithm to optimize the final ranking.

1.2.2 Overall Analysis

In the previous TRECVID competitions, we presented the use of Fuzzy Decision Trees for this kind of application [3]. The approach provided as result a set of classification rules which were human understandable, thus allowing further developments. This approach enables us to discover that, when addressing large, unbalanced, multiclass data sets, a single classifier as the FDT is not sufficient. For instance, the space of negative examples is so large (proportionally to the positive examples) that we can not model it correctly.

Thus, based on this observation, in [4], forests of FDT have been introduced to cover better the whole input space.

Moreover, in the presented runs of this year, we introduced some novel methods to rank the shots after their classification by means of a forest of FDT.

2 Introduction

The method used for the NIST TRECVID'2007 evaluation task is based on Forests of Fuzzy Decision Trees (FFDT) and the RankBoost algorithm. More precisely, after the used of the Salammbô software to construct a set of fuzzy decision trees for each feature, the RankBoost algorithm has been introduced to optimize the ranking of shots.

A first preliminary step, before the construction and the use of a FFDT, consists in transforming the data (devel and test set of the shots extracted from the video) in order to be processed by the Salammbô software.

Then, the main process is decomposed in three steps. In Section 3, the generation of vectors of descriptors from the keyframes and the XML files is presented. In Section 4, the training process, i.e. the constitution of training sets that should be process by the Salammbô software to construct FDT, is described. In Section 5, the method of processing FDT to classify keyframes is explained. In particular, we focus on the process of aggregation of the individual FDT decisions, thus enabling the ranking of the test keyframes. Before concluding, in Section 6 each of the performed runs is detailed.

3 Extraction of Image Descriptors

3.1 Visual Information Descriptors

The *Visual Information Descriptors* are obtained directly and exclusively from the keyframes. In order to obtain spatial-related information, we segmented the image into 5 overlapping regions (see Figure 1). Each of them corresponds to a spatial part of the keyframe: top, bottom, left, right, and middle. The five regions do not possess the same size to reflect the importance of the contained information based on its position. Moreover, regions overlap in order to introduce a dependence between them.

Afterwards, for each region we computed the associated histogram in the HSV space. Based on the importance of the region, the histograms of each region is valued in a more or less precise way (i.e. number of bins): 6x3x3 or 8x3x3.

At the end of this procedure, we obtain the what we called the Visual Information Descriptors, a set of numerical values (belonging to [0,1]) that characterizes every keyframe.

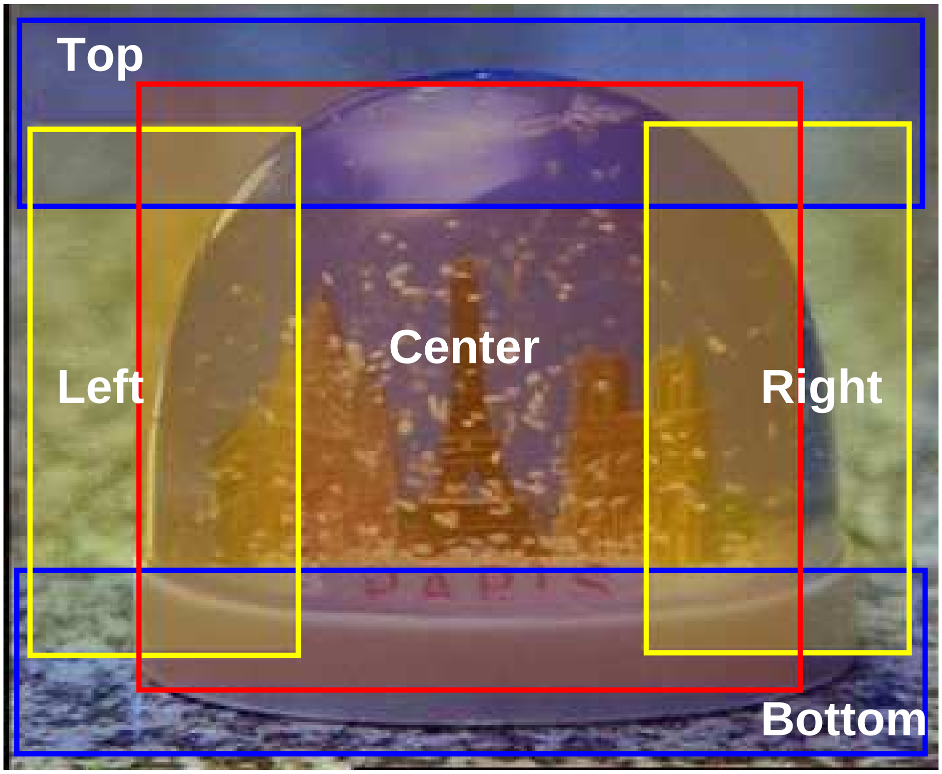

Figure 1: Spatial segmentation of a Keyframe

3.2 Video Information Descriptors

All the *Video Information Descriptors* we used, was extracted from the master shot reference XML file associated to the video, which is nothing else than the result of a shot detection process (here by [5]).

For a given keyframe, these descriptors were extracted by the parsing specific XML tags. In this way, for every shot we obtained the following information:

- the name of the keyframe (ident)
- the temporal position (timecode of the beginning) of the shot and of the keyframe itself,
- the duration of the shot containing the keyframe and the duration of the original shot (in case of a merging of subshots)

At the end, we obtain the Video Information Descriptors, a second set of numerical values that characterize keyframes.

3.3 Class Descriptor

The *Class Descriptor* is obtained from the human indexation of the video. It corresponds to the "correct" feature(s) to be detected on a shot. The Class Descriptor is extracted from the file obtained from the collaborative work of indexation of the devel video. Note that a keyframe can be associated with more than one class descriptor depending on the result of the indexation process.

4 Training with devel keyframes

In this section, we recall briefly how the training enables us to obtain a classifier (FFDT) that will be used afterwards to classify and rank the test keyframes (see Section 5). For more details on the method, see [4].

4.1 Building a training set

In order to use the Fuzzy Decision Trees (FDT) learning method, which is a supervised learning method, we must have a training set in which there are cases *with* the feature to be recognized and examples that do *not* possess that feature. In fact, decision trees construction methods are based on the hypothesis that the value for the class is equally distributed.

Thus, to have a valid training set for the construction of a FDT, we have to balance the number of keyframes of each class by (randomly) selecting a subset of the whole devel data set with an equal number of cases in each class.

4.2 Construction of a Forest of Fuzzy Decision Trees

4.2.1 Fuzzy Decision Trees

Inductive learning raises from the *particular* to the *general*. We build a tree from the root to the leaves, by successive partitioning the training set into subsets. Each partition is done by means of a test on an attribute and leads to the definition of a node of the tree. (for more details, see [3]).

The construction and the use of the FDT was done by means of the Salammbô software. This software was developed for building FDT efficiently and it enables us to test several kinds of parameters of the FDT [2]. Moreover, the automatic method to build a fuzzy partition on the set of values of the numerical attributes, mentioned above, was implemented enabling us to avoid the prior definition of fuzzy values of attributes. Various parameters (t-norms, t-conorms) can be set in the Salammbô software and have been tested in the process of classification on different kinds of databases.

4.2.2 Forests of Fuzzy Decision Trees

A forest of FDTs was constructed for each of the 36 features. A FFDT is composed of a given number n of Fuzzy Decision Trees. Each FDT F_i of the forest is constructed from a training set T_i. Each training set T_i beeing a random sample of the whole training set, as described in Section 4.1. For instance, for the Sports feature, a subset of keyframes with each class (with Sports, or without Sports) was ramdomly selected in order to build a training set.

5 Classification and ranking of test shots

5.1 Classifying keyframes with a Forest

The process of classification by means of a *single* Fuzzy Decision Tree has been explained in [3] and the process of classification by means of a forest of FDT has been explained in [4]. We recall here some basic steps of the method.

With a forest of n FDTs, corresponding to a single feature to be recognized, the classification of a keyframe k is done in two steps:

1. Classification of the keyframes k by means of the n FDT of the forest: each k is classified by means of each FDT F_i in order to obtain a degree $d_i(k) \in [0, 1]$ for the keyframe of having the feature. Thus, n degrees $d_i(k),\ i = 1...n$ are obtained from the forest for each k.

2. Aggregation of the $d_i(k),\ i = 1...n$ degrees for each k in order to obtained a single value $d(k)$, which corresponds to the degree in which the forest believe that the k contains the feature.

Two kinds of aggregating method has been used to compute the degree $d(k)$ that aggregates all the $d_i(k)$ degrees.

5.2 Simple votes

This basic aggregation has been done by summing the whole degrees: $d(k) = \sum_{i=1}^{n} d_i(k)$.

5.3 Weighted votes

This year, we introduce an aggregation done by a weighted sum of the whole degrees. The weight of each degree $d_i(k)$ has been set as the training accuracy of the corresponding FDT.

Thus, we computed $d(k) = \sum_{i=1}^{n} w_i d_i(k)$ with w_i, from $[0, 1]$, the accuracy of the corresponding FDT valued on the training set.

5.4 Ranking Test shots

The degrees of all the keyframes $d(k)$ of a shot are aggregated to obtain a global degree $D(S)$ for each shot to have the feature. The degree $D(S)$ for the shot S containing the feature is valued as $D(S) = \max_{k \in S}(d(k))$ (at least one keyframe of the shot should contain the feature).

As result, for every shot of the Test set, a degree is obtained from each FDT of the forest. The higher $D(S)$, the higher it is believed that S contains the corresponding feature.

From these degrees, two approaches can be used to obtain a ranking of shots.

5.4.1 Simple aggregation

The basic approach, used in [4], is to rank the shots by means of the degrees $D(S)$.

5.4.2 RankBoost algorithm

Another approach we introduced this year is to use the RankBoost Algorithm [1] for combining the FDT.

RankBoost is a supervised learning algorithm that learns a real-valued (scoring) function, by optimising a specific error measure suitable to ordering sets of objects. More precisely, RankBoost takes as input a set of examples (i.e. keyframes) represented as vectors, and finds a weight vector that optimises a convex approximation of the following error: order all the examples by decreasing values of their dot product with the weight vector, and compute the mean rank of the examples containing the feature.

Although this criterion is not simply related to Average Precision, it is still particularly appropriate to ranking tasks. Indeed, the algorithm takes into account, during learning, the relative scores between examples that contain/do not contain the high-level feature: it finds a weight vector that assigns higher scores to the examples that contain the feature than to the other examples.

In our setting, the examples were the keyframes, and, for the vectorial representation, we used the decision stumps technique described in [1], applied to the output of a set of FDTs. In the end, the algorithm learns a non-linear combination of the scores d_i of the various fuzzy decision trees.

6 Experiments

In this part, we present the results obtained by our approach.

6.1 Submitted runs

The five runs that have been submitted this year are the following:

Run #1: Forest of 35 fuzzy decision trees. The aggregation of values in each FDT is done by means of the Zadeh's t-norms (see [3]).

Run #2: Forest of 25 fuzzy decision trees and

- for the features 5, 6, 7, 8, 10, 12, 13, 15, 19, 20, 23, 24, 25, 26, 27, 28, 32, 35, 36, 37, 38, and 39: the aggregation of values in each FDT is done by means of the Zadeh's t-norms (see [3]). The RankBoost algorithm is used to optimize the ranking.
- for the features 1, 3, 4, 9, 11, 14, 16, 17, 18, 29, 30, 31, 33, and 34: the aggregation of values in each FDT is done by means of the Łukasiewicz's t-norms (see [3]).

Run #3: Forest of 25 fuzzy decision trees. The aggregation of values in each FDT is done by means of the Zadeh's t-norms (see [3]).

Run #4: Forest of 25 fuzzy decision trees. The aggregation of values in each FDT is done classically (see [3]).

Run #5: Forest of 25 fuzzy decision trees. The aggregation of values in each FDT is done by means of the Zadeh's t-norms (see [3]) and the weighted sum is used to optimize the ranking. Rankings have been submitted with this approach only for the features 5, 6, 7, 8, 10, 12, 13, 15, 19, 20, 23, 24, 25, 26, 27, 28, 32, 35, 36, 37, 38, and 39.

These runs enables us to compare various approaches:

- Runs #1 and #3: influence of the size of the forest,
- Runs #2 (part) and #3: influence of the RankBoost,
- Runs #3 and #5: influence of the weighted votes,
- Runs #2 (part) and #5: influence of the RankBoost vs the weighted votes,
- Runs #2 (part), #3, and #4: influence of the t-norms.

All these comparisons can be highlighted in Table 1 and will be commented in the following.

6.2 Results

The results in Inferred Average Precision for the five runs are presented in Table 1. In this table, the following labels are used (more details are given in the previous section):

- F35_Zad: results of Run #1, forest of 25 FDT with an aggregation by means of the Zadeh's t-norms.
- F25_RkBst: results of Run #2, forest of 25 FDT and ranking with the RankBoost algorithm (only on a restricted set of features).
- F25_Luk: results of Run #2, forest of 25 FDT with an aggregation by means of the Łukasiewicz's t-norms (only on a restricted set of features).
- F25_Zad: results of Run #3, forest of 25 FDT with an aggregation by means of the Zadeh's t-norms.
- F25_Class: results of Run #4, forest of 25 FDT with a classical aggregation.
- F25_Wght: results of Run #5, forest of 25 FDT with an aggregation by means of the Zadeh's t-norms and a weighted vote (only on a restricted set of features).
- Median: median of the results for the whole runs submitted at TRECVID 2007 as Type A approaches.

Concerning the results for run #2, we separate the results from the approach with the RankBoost algorithm, and the approach without it (and with the use of the Łukasiewicz's t-norms).

6.3 Discussion

As a global comment, we obtain the same kind of conclusion as in the previous year: our approach is generally better when finding good shots for a feature than when ranking them. It can be seen with the hits found within the 100, 1000, or 2000 shots of the ranking.

The use of a bigger forest (35 FDT rather than 25) seems to be a good improvement and should be studied better by increasing again the number of the FDT. The use of the weighted votes seems also interesting but it should be studied better by applying the weights for several sizes of forests.

The RankBoost algorithm seems also to be a good improvement of the ranking: the hits within the 100 first ranked shots are good and sometimes better than the corresponding hits for the FFDT without the RankBoost step. The RankBoost algorithm offers here a good improvement by re-ranking the hits and putting them in the first places of the ranking. However, it can be penalized by the descriptors and the fact that it has only be tested with the output of the FDT. Future works should be to experiment the RankBoost algorithm with the output classification degrees from various machine learning tool in order to better characterize the advantages of its use.

It should also be recalled that these results highly depend on the feature to recognize and a finest analysis of our results should also be done depending on that.

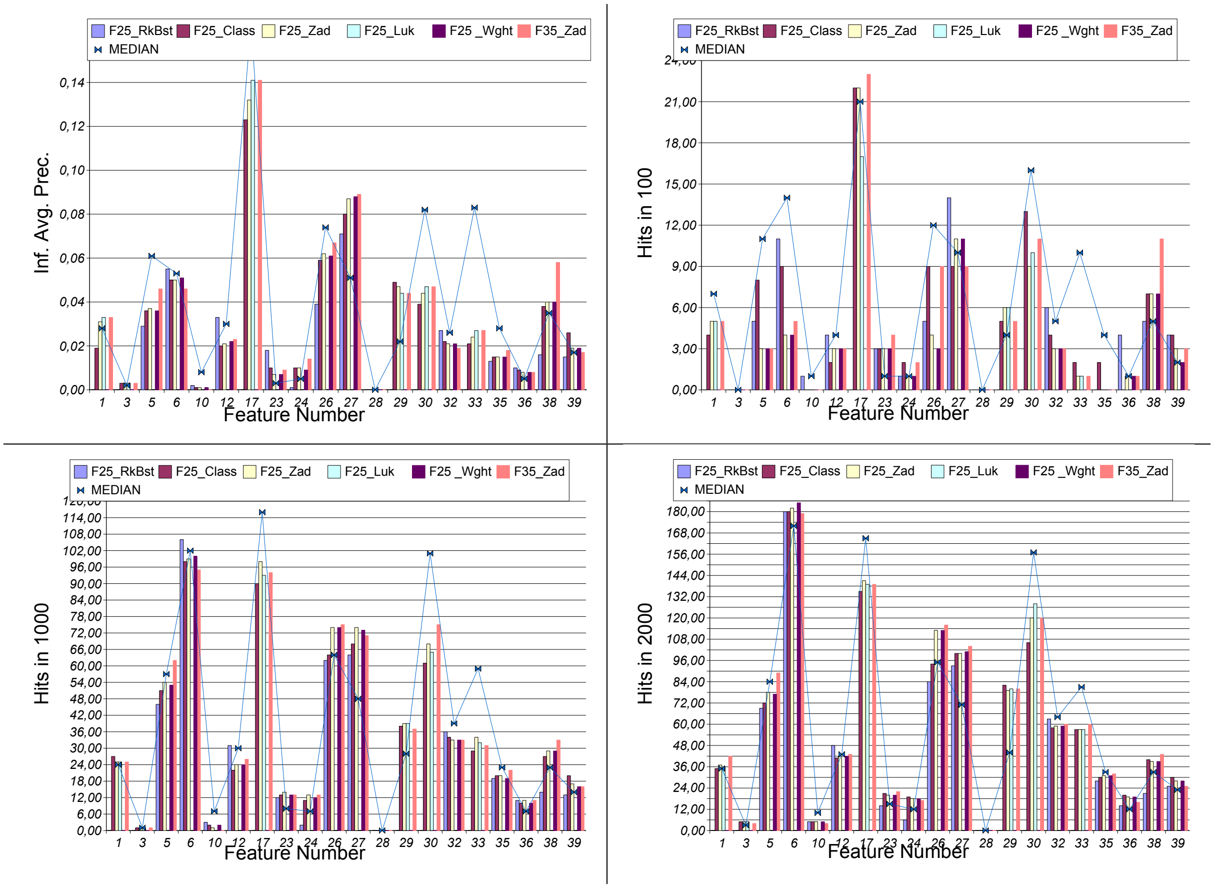

Table 1: Comparison of the submitted runs and median of the participants' runs at TRECVID'2007 (only type A systems)

7 Conclusion

We have introduced this year several new approaches to improve the results of our AI tools. Although, this work is at a preliminary stage, we obtained encouraging results.

As already stated in our previous participation, one of the main drawbacks of our method is that it is based on very simple and generic visual descriptions. The main reason for that constant in our approach is that we focus our approach on the step after the generation of these descriptors. In further works, we will try new kind of descriptors in order to study their influence on the final results.

Finally, if the results of Forest of Fuzzy Decision Trees seems promising when classifying shots, the introduction of a dedicated approach to optimize the ranking (the RankBoost algorithm or the weighted votes) seems to be interesting steps to improve the whole approach. It will be our main direction to study and adapt again the method in order to improve it for the high-level feature extraction process.

References

[1] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

[2] C. Marsala. *Apprentissage inductif en présence de données imprécises : construction et utilisation d'arbres de décision flous.* Thèse de doctorat, Université Pierre et Marie Curie, Paris, France, Janvier 1998. Rapport LIP6 n^o 1998/014.

[3] C. Marsala and M. Detyniecki. University of Paris 6 at TRECVID 2005: High-level feature extraction. In *TREC Video Retrieval Evaluation Online Proceedings*, November 2005. http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html.

[4] C. Marsala and M. Detyniecki. University of Paris 6 at TRECVID 2006: Forests of fuzzy decision trees for high-level feature extraction. In *TREC Video Retrieval Evaluation Online Proceedings*, November 2006. http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html.

[5] C. Petersohn. Fraunhofer HHI at TRECVID 2004: Shot boundary detection system. Technical report, TREC Video Retrieval Evaluation Online Proceedings, TRECVID, 2004. URL: www-nlpir.nist.gov/projects/tvpubs/tvpapers04/fraunhofer.pdf.

[6] 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.