High Scale Video Mining with Forests of Fuzzy Decision Trees

Christophe Marsala
Université Pierre et Marie Curie Paris6
CNRS UMR 7606, LIP6,
104 av. du Président Kennedy
Paris, F-75016, France
Christophe.Marsala@lip6.fr

Marcin Detyniecki
Université Pierre et Marie Curie Paris6
CNRS UMR 7606, LIP6,
104 av. du Président Kennedy
Paris, F-75016, France
Marcin.Detyniecki@lip6.fr

ABSTRACT

In this paper, a video mining method based on the use of Forests of Fuzzy Decision Trees (FFDT) is presented. We focus on the use of such a FFDT in a high scale video mining application and highlight the main advantages of using fuzzy set theory in such a process.

Categories and Subject Descriptors

H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems—*Video*; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Indexing methods*; H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems—*Evaluation/-Methodology*

General Terms

Algorithms, Experimentation, Performance.

Keywords

Video mining, Fuzzy Decision Trees, Forest of Fuzzy Decision Trees, TRECVid.

1. INTRODUCTION

Nowadays, the amount of recorded video is continually increasing leading to a growing need to find solution of how to handle it automatically. One of the main issues is to be able to index these data with high-level semantic concepts (or features) such as "indoor/outdoor", "people", "maps", "military personnel", etc.

Video indexing aims at analyzing a video, to find its seminal content, and to associate features to any of its part.

Most of video indexing is done manually, thanks to a human operator that associates himself features to parts of a video. However, due to the growth of recorded video, the introduction of automatic approaches as data-mining-based ones is a very promising perspective.

CSTST 2008 October 27-31, 2008, Cergy-Pontoise, France

Video mining is typically an inductive machine learning approach. It has as starting point a set of correctly labeled examples used to train or to build a model. Later, this model is used to perform an automatic classification of any forthcoming examples, even if they have not been already met before.

Inductive machine learning is a well-known research topic with a large set of methods, one of the most commonly used approaches being the decision tree approach (DT). However, robustness and threshold problems appear when considering classical DTs to handle numerical or imprecisely defined data. The introduction of fuzzy set theory, that leads to the fuzzy decision tree approach (FDT), enables us to smooth out these negative effects.

In the TRECVID competitions, we presented the use of Fuzzy Decision Trees for this kind of application [11]. The approach based on FFDT 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 [12], forests of FDT have been introduced to cover better the whole input space. The use of forests of decision trees is well-known in classical machine learning, see for instance [4]. In fuzzy machine learning, forests of fuzzy decision trees have been introduced some years ago and are becoming more popular nowadays [3, 6, 8, 9]. These approaches differ by the way the FDT are multiplied to grow the forest.

In this paper, we will show that this kind of approach is very useful for such a high scale challenge. First, the video is pre-processed in order to obtain a set of descriptors to feed the video mining algorithm (see Section 2). Afterwards, in the training step described in Section 2.4, the system learns how to recognize the presence of a high-level feature in a shot using manually indexed videos. Finally, in the classification step (Section 3), the system predicts the presence of a high-level feature in a shot for any additional video.

Before concluding, in Section 4 a set of experiments is described and discussed.

2. FROM VIDEO TO TRAINING SETS

From a video, a sequence of steps, such as the extraction of basic descriptors are necessary to feed the video mining

algorithm.

First of all, the video is automatically segmented into temporal shots. A shot is a sequence of the video with a more or less constant content. Then each shot is associated with a set representative images, called *frames*. The number of frames can vary from 1 to around 10 frames, depending on the complexity of its contents.

Secondly, a set of descriptors is extracted from each frame. The descriptors that are extracted can be of two kinds: *Visual Information Descriptors* and *Video Information Descriptors*. Moreover, frames from the video training set are also associated with a set of *Class Descriptors* obtained by means of a manual indexation of the videos.

2.1 Visual Information Descriptors

The *Visual Information Descriptors* are obtained directly and exclusively from the frames. In order to obtain spatial-related information, the frame is segmented into 5 overlapping regions (see Figure 1).

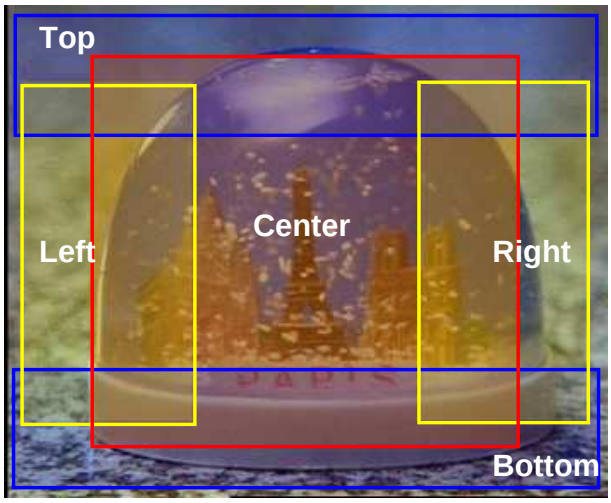

Figure 1: Spatial segmentation of a frame

Each of them corresponds to a spatial part of the frame: top, bottom, left, right, and middle. The five regions have not the same size to reflect the importance of the contained information based on its position. Moreover, regions overlap in order to introduce a dependency between them.

Afterwards, for each region the associated histogram in the well-known Hue-Saturation-Value (HSV) space is computed. The number of bins of the histogram follows the importance of the region by being valued in a more or less precise way: 6x3x3 or 8x3x3.

At the end of these steps, a set of Visual Information Descriptors, *ie.* a set of numerical values (belonging to [0,1]), characterizing each frame is provided.

2.2 Temporal Information Descriptors

The *Temporal Information Descriptors* are information related to the position of the frames, and of the shots, in the video. For every shot, we extract:

- the temporal position (timecode of the beginning) of the shot and of the frame itself,
- the duration of the shot containing the frame and the duration of the original shot if the shot results from a merging of smaller shots.

At the end of this step, the Temporal Information Descriptors, a second set of numerical values that characterize frames is obtained.

2.3 Class Descriptor

The *Class Descriptor* is from the result of a human indexation of the video. It corresponds to the *correct* high-level feature(s) or concept(s) to be detected on a given shot. The human indexation process is done by associating to each *frame* of the video of training set, a feature. Thus, the shots are described by the concepts appearing in at least one of its frames. Furthermore, a frame can be associated with more than one class descriptor.

2.4 Building a training set

In order to use the Fuzzy Decision Trees (FDT) learning method, we must have a balanced training set in which there are cases *with* the feature to be recognized and examples that do *not* possess that feature.

Since decision trees construction methods are based on the hypothesis that the value for the class is equally distributed, we have to balance the number of frames of each class by (randomly) selecting a subset of the whole development data set with an equal number of cases in each class.

3. DETECTING HIGH LEVEL FEATURES (CONCEPTS) ON A SHOTS

In the particular context of large data sets, as for instance for video indexing, we can focus our attention on the elements (here shots) that are classified with a high degree of confidence. In fact, it may be sufficient and more interesting to have some good examples rather than a average classification overall.

Thus, often in video indexing the classified shots are ranked based on the credibility on the fact that the shot contains the features or not.

FFDT can be easily used to provide a ranking of shots for a given feature. First, using them a classification of frames is done. Secondly, an aggregation the results leads to the classification of the shot. Finally, the shots are ranked based on the aggregated value, which corresponds to credibility that the features appear in it.

First of all, we recall briefly how the training enables us to obtain a classifier (FFDT) that will be used afterwards to classify and rank the test frames (see Section 3). For more technical details on this method, refer to [12].

3.1 Fuzzy Decision Trees

Inductive learning raises from the *particular* to the *general*. A tree is built, 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 [10]).

When mining numerical data, with an *Fuzzy* Decision Tree, a definition of fuzzy values of attributes is necessary. In the case of high scale mining, an automatic method is necessary. We build a fuzzy partition on the set of values of the numerical descriptors.

Moreover, in order to address high scale data sets the FDT have to be build efficiently. Thus, we used the Salammbô

software [10].

3.2 Forests of Fuzzy Decision Trees

A forest of FDTs was constructed for each high-level feature, to be detected. A FFDT is composed of n Fuzzy Decision Trees. Each FDT F_i of the forest is constructed based on a training set T_i, each training set T_i being a balanced random sample of the whole training set, as described in Section 2.4.

3.3 Classifying frames with a Forest

The process of classification by means of a *single* Fuzzy Decision Tree is explained in [11] and the process of classification by means of a forest of FDT is explained in [12]. We recall here some basic steps of the method.

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

1. Classification of the frames 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 frame of having the feature. Thus, n degrees $d_i(k),\ i = 1 \dots n$ are obtained from the forest for each k.
2. Aggregation of the $d_i(k),\ i = 1 \dots n$ degrees for each k in order to obtain 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 to compute the degree $d(k)$ that aggregates all the $d_i(k)$ degrees, were tested:

- Simple vote: This basic aggregation corresponds to the sum of all the degrees: $d(k) = \sum_{i=1}^{n} d_i(k)$.
- Weighted vote: Aggregation can also be weighted by the training accuracy of the FDT. Thus, the sum of the degrees becomes $d(k) = \sum_{i=1}^{n} w_i d_i(k)$, where w_i, from $[0,1]$ corresponds to the accuracy of the corresponding FDT F_i valued on the training set.

3.4 Ranking the shots

The degrees of all the frames $d(k)$ of a shot are aggregated to obtain a global degree $D(S)$ for each shot. Since it is sufficient that at least one frame of the shot contains the feature to be able to say that the shot contains the feature, the degree $D(S)$ for the shot S containing the feature is valued as $D(S) = \max_{k \in S}(d(k))$.

As result, for every shot, a degree is obtained. The higher $D(S)$, the higher it is believed that S contains the corresponding feature. Thus, the straightforward approach, used in [12], consists in ranking the shots by means of the degrees $D(S)$.

4. EXPERIMENTS

In order to study the behavior of FFDT as tool for high scale mining some experiments were conducted.

First, we study the influence of the number of FDT in the forest on a classical dataset. For this experiment we choose the well-known Waveform dataset [5], from the UCI repository [2]. We focus on the error rate (*ie.* the rate of bad-classified examples).

Secondly, we present the performance of our video mining tool on a rather large real video data set.

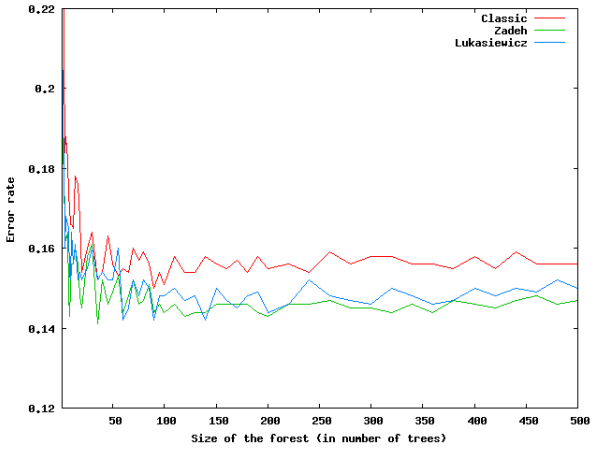

Figure 2: Influence of the size of the forest

4.1 Waveform dataset

For a first study on the influence of the size of the FFDT, we have chosen the well-known Waveform dataset [5]. This dataset is often used in the machine learning community and a lot of algorithms have been evaluated with it. For instance, in [4] or in [7], some results with this dataset can be found for algorithms combining decision trees (adaboost, random forests,...). This point is not discussed in this paper as our aim is not to compare FFDT with classical approaches but to show the influences of the size of the forest and of the choice of the aggregation operators.

Moreover, this dataset has the following interesting properties. There are 3 (symbolic) classes to recognize, and 21 real-valued attributes. Data can be noised (as in real-world problems). The dataset is composed of a total of 5000 instances. In this experiment, the dataset has been decomposed into 2 subsets: the training set is composed of 3500 examples, and the test set is composed of 1500 examples.

Similar to the video indexing application, FFDT are constructed following the protocol :

step 1 a class c is chosen from the set of classes

step 2 a sampling of the training set is done by taking all the examples associated with the class c, and in addition a random sampling of examples having one of the other classes. The idea being to have the same number of examples with the class c, than examples with another class.

step 3 from this sampling, a FDT is constructed by means of the Salammbô software [10].

This process is resumed for each of the 3 classes in order to obtain 3 FDT, each one enabling the classification of an example with regards to a given class. Thus, the process described in Section 3.2 can be applied.

An example from the test set is classified by each of the FDT and the individual classification results are aggregated in order to determine the final class of the example. The classification is done similarly than the one described in Section 3 (but without he terminal step of ranking that is useless here).

In Figure 2, we present the error rate variations when classifying the test set with various sizes of FFDT. We re-

call that the error rate is the ratio between the number of bad-classified examples and the total number of classified examples. Thus, the error rate ranges from 0 ("no bad classification") to 1 ("all the examples were badly classified"). In Figure 2, the average error rate is around 0.15 that corresponds to around 15% examples are badly classified. Forests with up to 500 FDT have been constructed, and each one of these forests has been used to classify all the examples from the test set.

From Figure 2, it can be shown that the error rate decreases with the size of the forest of fuzzy decision trees (in number of trees).

In Figure 2, the "Classic" graph corresponds to the error rate when using a FDT classically. That is, by considering that each fuzzy node of the FDT outputs a non fuzzy result. Here, the decision is done thanks to the use of the alpha-cut of degree 0.5 of the fuzzy values. In this case a FDT outputs a single class with a full membership, either "has the class" or "has not the class", for each example. The two other graphs show results in the complete fuzzy use of the FDT (in this case a FDT outputs a degree of membership (ranging from 0 to 1) of the example to each class). The "Zadeh" graph corresponds to the error rate obtained when using the Zadeh t-norms (the minimum and maximum operators) when classifying examples. The "Lukasiewicz" graph corresponds to the error rate obtained when using the Lukasiewicz t-norms when classifying examples. As the process of classification with a FDT is out of the scope of this paper, more details can be found in [10].

From Figure 2, it is clear that the use of the fuzzy set theory reduce the error rate for this problem. In this case, the error rate is always better in the cases of the fuzzy uses of the FFDT, no matter the size of the forest.

A final remark concern the complexity and the runtime of the whole process. The complexity for the construction of a FFDT is related to the number of the FDT it contains and, thus, is relatively low (if taken into account the fact that a small number of FDT is needed to obtain a small error rate). For instance, the runtime of the experiment described here, composed of the construction of 500 FDT, the classification of the test set by each of these FDT and with each of the presented operators (Classic, Zadeh, and Lukasiewicz), is around 7350 seconds on a multiprocessor computer (10 core 2.93 Ghz, 64 Go RAM, with GNU/Linux 2.6).

4.2 The TRECVid Challenge

In order to compare our approach to others in a real-world framework, we participated to the high-level feature extraction task, at the TRECVid 2007 Challenge [13, 14]. In this task, each team participant should propose, for each high-level feature from a given set, a ranking of at most 2000 video shots that contain it. The results presented here are the results obtained with our submission to the TRECVid 2007 Challenge and the evaluation process that has been conducted by the NIST institute.

The addressed features (and their identification number that is used in the forthcoming) are: sports (1), weather (3), office (5), meeting (6), desert (10), mountain (12), waterscape/waterfront (17), police security (23), military personnel (24), animal (26), computer TV screen (27), US flag (28), airplane (29), car (30), truck (32), boat or ship (33), walking or running (34), people marching (35), explosion fire (36), maps (38), and charts (39).

In this task of the TRECVid 2007 Challenge, the corpus of video is composed of 109 videos (around 30 minutes length each) and 18142 reference shots (shots have been provided thanks to [15]).

Several kinds of forests were studied and applied in order to submit several rankings to the challenge. We highlight here some of our results on the comparison of two kinds of FFDT: a forest composed of 35 FDT, and a forest composed of 25 FDT. For the FFDT with 25 FDT, results are presented in order to highlight better the advantages of using the fuzzy set theory in this kind of video mining task.

All the comparisons are given in the figures on Table 1 and will be commented in the following. Four approaches can be compared in this table: the results obtained by means of a forest of 25 FDT used classically ("F25_Classic"), the results of a forest of 25 FDT used with the Zadeh's t-norms ("F25_Zadeh"), the results of a forest of 35 FDT used with the Zadeh's t-norms ("F35_Zadeh"), and the median of the results for the whole results submitted at TRECVID 2007 by all the participating teams.

In Figure 3a, variations of the *Inferred Average Precision* [1] are presented. We recall that the *Inferred Average Precision* is the evaluation measure used in this TRECVid 2007 Challenge. It is based on the well-known precision of a classifier. The precision of a classifier with regards to a given class is valued as the ratio between the number of perfectly classified examples and the total number of test examples in the class. In the TRECVid 2007 Challenge, due to the high size of the test corpus, it is impossible to have the total number of examples in the class for each feature. Thus, the TRECVid evaluators have decided to infer a value of the precision by substituting the total number of examples in the class with the total number of examples in the class pertaining to the whole submitted results. This value is called the *inferred precision*. Evaluating methods by means of an inferred value is a well-known approach when the size of the corpus is too large to be fully evaluated.

In Figure 3a, it can be shown that the FFDT performance depends highly on the kind of features that should be recognize. It should be greatly linked to the descriptors (see Section 2) that are used to represent the shots. More complex features (for instance "car" (30) or "boat/ship" (33)) need more complex descriptors in order to enable the FFDT to perform better. For features such as "animal" (26), "computer TV screen" (27) or "maps" (38), FFDT obtain good results (in the first half of all the approaches that participated to the challenge).

In Figure 3b, Figure 3c, and Figure 3d, the number of hits, for each feature, is presented at several level of the submitted rankings. In Figure 3b, the number of perfectly classified examples that are ranked in the 100 first shots is presented. In Figure 3c, the number of perfectly classified examples that are ranked in the 1000 first shots is presented. In Figure 3d, the number of perfectly classified examples that are ranked in the 2000 first shots is presented. These 3 values, for each feature, are part of the evaluation of the methods that participated to the TRECVid Challenge.

As a general comment, it can be seen that the FFDT perform better when considering high values for the ranking. One main reason lies in the fact that the FFDT is a classification tool and not a ranking tool. Thus, if it can be easy for a FFDT to determine if a shot contains a feature or not, it is more difficult for it to set a ranking for all the shots

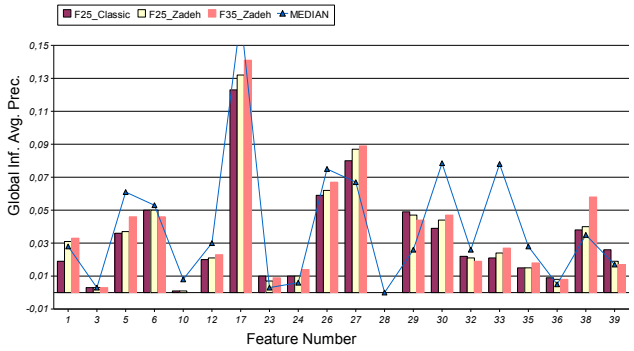

Figure 3a: Global Inf. Avg. Precision

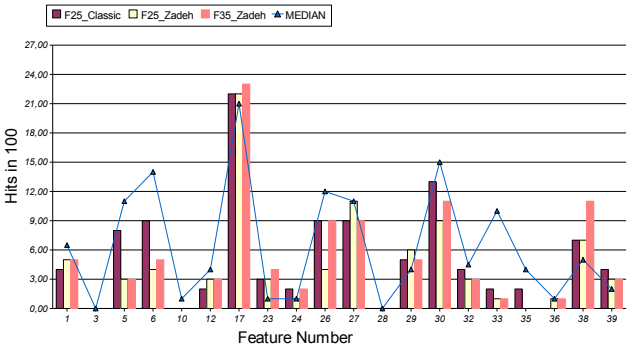

Figure 3b: Good Hits in the 100 firsts

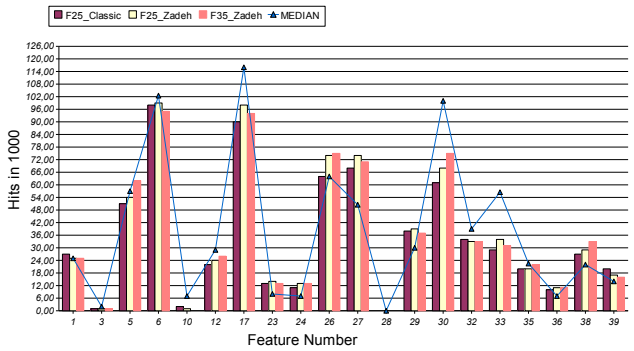

Figure 3c: Good Hits in the 1000 firsts

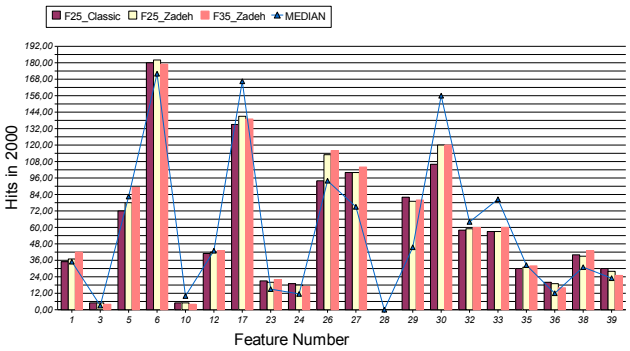

Figure 3d: Good Hits in the 2000 firsts

Table 1: Comparison of FFDT and median of the participants' runs at TRECVID'2007

that contains a feature.

Moreover, in these figures, it can be shown that the "F35_Zadeh" FFDT performs better than "F25_Zadeh" that highlights the importance of the size of the forest in this application too. It can be also shown that FFDT using full fuzzy FDT is globally better than "F25_Classic". A remark can be made considering Figure 3c where "F25_Classic" can generate good results. In this case, it seems connected to the poor performance of the FFDT in this level of ranking rather than a good performance of the "F25_Classic".

For the rest of the results (Figure 3a, Figure 3b, Figure 3c, and Figure 3d), the FFDT using the full power of the fuzzy set theory when using the FDT performs always better than a classic-approach based.

4.3 Discussion

As a global comment, FFDT can be considered as an interesting application of the fuzzy set theory to handle a high scale video mining.

Results obtained at the TRECVid challenge are promising and highlight that FFDT is generally better to find good shots (classification) for a feature than to rank them.

The size of the forest is a crucial parameter and should be studied better, but a high number of FDT should be considered as a good heuristic to obtain good results.

Moreover, as for the Waveform experiment presented previously, the fuzzy set theory when classifying with FDT offers better results than the classical use of FDT.

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.

5. CONCLUSION

In this paper, the forest of fuzzy decision trees are introduced and their application to high scale problems is shown. The results obtained by this approach in the real world application conducted, highlight the interest of using fuzzy set theory both for the construction of the fuzzy decision trees and for their use in classification. The participation to the TRECVid Challenge in order to confront the FFDT with a concrete and real video mining case of application offers a good way to ensure the scalability of this kind of fuzzy approach.

In the future, the study of other kinds of descriptors to encode the video shots, will be conducted in order to improve the results for other kind of high-level features.

6. REFERENCES

[1] Guidelines for the TRECVID 2006 evaluation - National Institute of Standards and Technology, 2006. http://www-nlpir.nist.gov/projects/tv2006/tv2006.html.

[2] A. Asuncion and D. Newman. UCI machine learning repository – University of California, Irvine, School of Information and Computer Sciences. http://www.ics.uci.edu/~mlearn/MLRepository.html, 2007.

[3] P. P. Bonissone, J. M. Cadenas, M. C. Garrido, and R. A. Díaz-Valladares. A fuzzy random forest: Fundamental for design and construction. In *Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'08)*,

pages 1231–1238, Malaga, Spain, July 2008.

[4] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[5] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification And Regression Trees*. Chapman and Hall, New York, 1984.

[6] K. Crockett, Z. Bandar, and D. Mclean. Growing a fuzzy decision forest. In *Proceedings of the 10th IEEE International Conference on Fuzzy Systems*, pages 614–617, December 2001.

[7] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, April 2006.

[8] C. Z. Janikow and M. Faifer. Fuzzy decision forest. In *Proceedings of the 19th International Conference of the North American Fuzzy Information Processing Society (NAFIPS'00)*, pages 218–221, July 2000.

[9] C. Marsala and B. Bouchon-Meunier. Forest of fuzzy decision trees. In M. Mareš, R. Mesiar, V. Novák, J. Ramik, and A. Stupňanová, editors, *Proceedings of the Seventh International Fuzzy Systems Association World Congress*, volume 1, pages 369–374, Prague, Czech Republic, June 1997.

[10] C. Marsala and B. Bouchon-Meunier. An adaptable system to construct fuzzy decision trees. In *Proc. of the NAFIPS'99 (North American Fuzzy Information Processing Society)*, pages 223–227, New York, USA, 1999.

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

[12] 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.

[13] C. Marsala, M. Detyniecki, N. Usunier, and M.-R. Amini. High-level feature detection with forests of fuzzy decision trees combined with the rankboost algorithm. In *TREC Video Retrieval Evaluation Online Proceedings*, 2007. http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html.

[14] P. Over, W. Kraaij, and A. F. Smeaton. Guidelines for the TRECVID 2007 evaluation - National Institute of Standards and Technology, 2007. http://www-nlpir.nist.gov/projects/tv2007/tv2007.html.

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