High Scale Fuzzy Video Mining

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

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

ABSTRACT

In this chapter, we focus on the use of Forests of Fuzzy Decision Trees (FFDT) in a video mining application. We discuss how to learn from a high scale video data sets and how to use the trained FFDTs to detect concepts in a high number of video shots. Moreover, we study the effect of the size of the forest on the performance; and of the use of fuzzy logic during the classification process. The experiments are performed on a well-know non-video dataset and on a real TV quality video benchmark.

KEYWORDS

Video indexing, video mining, fuzzy decision tree, forest of fuzzy decision trees.

INTRODUCTION

Nowadays, the amount of recorded video is continually increasing leading to a growing need to find a way 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 staff", etc.

Video indexing aims at analyzing a video, to find its seminal content, and to associate concepts to any of its part. Today effective video indexing is done manually, by a human operator, who associates concepts 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 promising perspective.

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, the model is used to perform an automatic classification of any of the forthcoming examples, even if they have not been met before. Video mining is becoming a very active domain today and several conferences take into account this domain in their topics (for instance, the workshop on Video Mining of last IEEE International Conference on Data Mining, or ACM Multimedia conferences, etc.). Some works related to video mining can be cited: (Pan, J.-Y., & Faloutsos, C., 2002), (Rosenfeld, A. et al.; 2003), (Zhu, X., et al. 2005), the proceedings of the TRECVid challenge organized by the US institute NIST.

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 construction of fuzzy decision trees (FDT) able to smooth out these negative effects.

In the 2005 TRECVID competition, we studied the use of Fuzzy Decision Trees for this kind of applications (Marsala, C., & Detyniecki, M., 2005). The approach, based on single FDTs (one per concept), provided as result a set of classification rules, which were in the one hand, human understandable, thus allowing further human development; but in the other hand, this first series of tests enables us to discover that, when addressing large, unbalanced, multiclass datasets, a single classifier is not sufficient for direct automatic exploitation. Thus, based on these observations, in (Marsala, C., & Detyniecki, M., 2006) forests of FDT were 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 (Breiman, L., 2001). In fuzzy machine learning, forests of fuzzy decision trees have been introduced some years ago and are becoming more popular nowadays (Bonissone, P.P. et al., 2008), (Crockett, K., et al. (2001), (Janikow, C. Z., & Faifer, M., 2000), (Marsala, C., & Bouchon-Meunier, B., 1997). These approaches differ by the way the FDT are multiplied to grow the forest.

In this chapter, we show that this kind of approach is very useful for high scale challenge. First, we present how the video is pre-processed in order to obtain a set of descriptors to feed a video mining algorithm. Afterwards, we explain how Forest of Fuzzy Decision Trees are built and consecutively used to detect concepts in video shots.

In the experimental part of the chapter, we first study on a well-studied dataset both the influence of the size of the forest (in terms of number of trees), and the influence of using the FDTs in a fuzzy manner or not. Afterward, the proposed approach is confronted to a real world video dataset. The performance of FFDTs with respect to other approaches is explored. And the observations obtained on the previous dataset are confronted.

FROM VIDEO TO TRAINING SETS

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

First of all, the video is automatically segmented into temporal shots. Here, a *shot* is a sequence of the video with a more or less constant content. The content of a shot is considered to have the same "meaning". Generally, all the frames that compose a shot are very similar visually and differ only slightly. A shot can be very short (less than 1 second), for instance in action sequences of a video, or it can be very long, for instance if the sequence in the video shows only a still host talking to the camera. A shot can be associated with a set of representative images, called *frames*. The number of frames can vary from at least 1 to more than 10 frames, depending on the complexity of its contents.

Secondly, two kinds of descriptors are extracted from each frame: *Visual Information Descriptors* and *Video Information Descriptors*. Moreover, frames from the video training set are also associated with a set of *Class Label* obtained through a manual indexation of the videos.

Visual Information Descriptors

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

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.

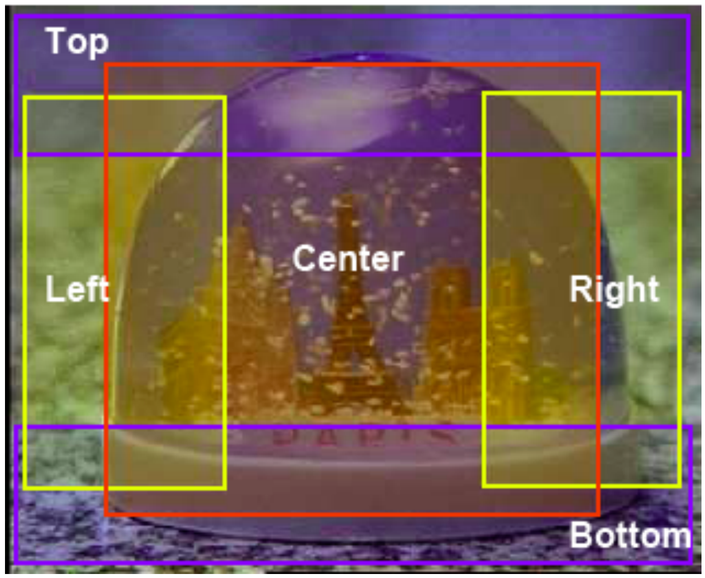

Figure 1: Spatial segmentation of a frame

For each region the associated histogram in 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 characterizing each frame is provided. This set is composed of values ranging from *0* to *1*. Each value represents the frequency of a color in the HSV space for the corresponding region it is associated with.

For instance, if the number of bins in the HSV space is 8x3x3 for the "Center" region, and if the number of bins is 6x3x3 for the four other regions, the Visual Information Descriptors of a frame is composed of 288 numerical values from [0,1].

The main interest in choosing overlapping regions in the frame is to create a link between the regions. This link is defined by the fact that their HSV histograms are valued on a same subset of pixels. The aim here is to alleviate a drawback of the attribute oriented inductive learning where attributes are usually considered as independent. Thus, creating a link between them will enable the learning model to take into account information about the color in the frame as a whole.

Here, the use of fuzzy histograms (histograms defined by means of a fuzzy definition of colors) can be a very interesting improvement of our approach and deserves further research.

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 (time code 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 a shot and its frames is obtained.

Class Label

The *Class Label* is the result of a human indexation of the video. It corresponds to the *correct* high-level concept(s) (features) to be detected on a given shot.
A concept is associated to each *frame* of the video through a human indexation process. 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.

Building a training set

In order to use the Fuzzy Decision Trees (FDT) learning method, we must have a training set in which there are cases *with* the concept to be recognized and examples that do *not* possess that concept. Moreover, the decision tree construction methods are based on the hypothesis that the value for the class is equally distributed. Thus, we have to balance the number of frames of each class by (randomly) selecting a subset of the whole development dataset. Each of such a subset must contain an equal number of cases in each class.

LEARNING AND DETECTING HIGH LEVEL CONCEPTS

In the particular context of large datasets, 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 an 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 concepts or not.

FFDT can be easily used to provide a ranking of shots for a given concept. First, a classification of frames is done by means of each tree of the FFDT. Secondly, an aggregation of 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 concepts appear in it.

First of all, we briefly recall how the training enables us to obtain a classifier (FFDT) that will be used later to classify and rank the test frames. For more technical details on this method, please refer to (Marsala, C., & Detyniecki, M., 2006).

Fuzzy Decision Trees

Inductive learning raises from the *particular* to the *general*. A tree is built, from the root to the leaves, by successively 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, please, refer to Marsala, C., & Bouchon-Meunier, B., 1999).

When mining numerical data, with a *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.

Finally, in order to address high scale datasets, the FDT has to be built efficiently and the use of the Salammbô software has been introduced in this step. This software has been introduced in (Marsala, C., & Bouchon-Meunier, B., 1999). It enables the construction and the use of fuzzy decision trees. A lot of parameters can be set (measure of discrimination, family of t-norms, parameters to build fuzzy partitions, etc.) in this software to build the FDT. Moreover, it has been written in C that enables it to handle very efficiently training sets with a very high number of examples.

Classifying frames with a Fuzzy Decision Tree

The process of a frame classification (*i.e.* detecting whether a concept is present), using a *single* Fuzzy Decision Tree is straightforward (Marsala, C., & Detyniecki, M., 2005). From each image-frame low-level features (in the same description space as for the training) are extracted. Based on this description, starting from the top of the tree, decisions are successively performed. The decisions can be made either in a *classical* or in a *fuzzy* manner as it is explained in the following.

When doing it classically, the decision is to follow one and only one of the branches. Technically the decision is done using the 0.5 alpha-cut degree of the fuzzy values. At the end, when a leaf is reached, the FDT outputs a single class with a full membership, either "has the class" or "has not the class", for each tested example.

When doing it in a fuzzy manner, if the decision is not crisp, for instance if the case to classify is close to the boundaries of the decision frontier, several branches can be followed. At the end the FDT's output is a degree of membership (ranging from 0 to 1) of the example observing the class. In order to compute these degrees, the trees are considered as a set of rules. All possible top-to-leaf paths are considered as a disjunctive set of rules and each individual path is considered as a conjunction of decisions. Based on this logical representation, the final degree can be computed using standard fuzzy logic operators. In this chapter, we consider Zadeh's family (maximum and minimum) and the Lukasiewicz one (bounded sum and its dual)[1].

These two families of t-norms have been chosen because their behavior is very different. However, any other family of t-norms could be used in this process and it deserves further research. For more details on the use of FDTs in fuzzy manner please refer to (Marsala, C., & Bouchon-Meunier, B., 1999).

Forests of Fuzzy Decision Trees

One way to address high scale datasets is to reduce the size of the problem. We propose to create, by sampling the large dataset, several smaller ones. Then we train one classifier on each of the size reduced sets. As a result we obtain a set of classifiers, which decisions have to be combined at the decision stage. An ensemble of decision tree classifiers is a so-called forest of decision trees.

This approach produces global classifiers that are not only robust, but having their score more reliable. Moreover, this technique allows to address another problem often observed in high

[1] We recall briefly that, given two values x and y from [0,1], the aggregation by means of the *Lukasiewicz t-norm* is valued as $\mathrm{T}(x,y)= max(x+y-1,0)$ and the aggregation by means of the *Lukasiewicz t-conorm* is valued as $\perp(x,y)=min(x+y,1)$.

scale-datasets: the balance of positive versus negative examples. In fact, even there is a lot of positive examples, the number of negative (or not labeled) examples is quickly overwhelming. If we sample several times asymmetrically, so that we obtain balanced smaller training sets, we not only solve the balance problem, but we also cover better the larger negative examples space.

A question remains the number of decision trees of need. Later in this chapter we study the influence, in terms of performance (error rate), of the number of Decision Trees used in a forest.

In the particular case of video mining, we construct a forest of FDTs for each high-level concept to be detected. A FFDT is composed of n Fuzzy Decision Trees. Each FDT Fi of the forest is constructed based on the training set Ti, each training set Ti being a balanced random sample of the whole training set, as described previously.

Classifying frames with a Forest of Decision Trees

The classification using a forest of n FDTs, is reduced to an aggregation problem. In fact, for a single concept, the classification of a frame k is carried out in two steps:

1. Classification of the frame by means of the n FDTs of the forest: each frame k is classified by means of each FDT Fi in order to obtain a degree $d_i(k) \in [0, 1]$ of having the concept. Thus, n degrees $d_i(k)$, $i=1\ldots n$ are obtained, from the forest, for each k.
2. Aggregation of the $d_i(k)$ ($i=1\ldots n$) degrees, into a single value $d(k)$, which corresponds to the degree in which the forest believes that the keyframe k contains the concept.

Two kinds of aggregating methods to compute the degree $d(k)$ were tested:

1. *Simple vote*: This basic aggregation corresponds to the sum of all the degrees:
$$d(k) = \sum_{i=1}^{n} d_i(k)$$
2. *Weighted vote*: Aggregation can also be weighted by taking into account 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 Fi valued on the training set.

Other aggregating methods could be used here and the choice of a convenient operator deserves further research. Moreover, a more complex aggregator could be used here in this step. For instance, a model could be tuned on the training data and a machine learning tool could be very useful to improve this aggregation.

Detecting a concept in a shot

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

Here, the choice of another aggregating operator (as the sum for instance) could also be done and it deserves further research in order to study whether it could improve the approach.

So, after this aggregation, for every shot, a degree is obtained. The higher *D(S)* is, the higher it is believed that the shot *S* contains the corresponding concept.

NUMBER OF DECISION TREES FOR HIGH SCALE MINING

As stated before, in order to cover high scale datasets it is suitable to sample the problem into several reduced sets of data and go from fuzzy decision trees (FDT) to forests of FDTs.
What is not clear is what is the precise effect, in terms of performance, on the number of trees that are used. The performance is measured by the error rate (*i.e.* the ratio of wrong classifications to the total number of classification evaluated). Thus, the error rate ranges from 0 ("no wrong classification") to 1 ("no correct classification").

Waveform datasets

In order to avoid any particularities of a video data set, we study the influences of the size of the forest and of the choice of the aggregation operators on the well-known Waveform dataset (Breiman, L. et al., 1984), from the UCI repository (Asuncion, A., & Newman, D., 2007). This dataset is often used in the machine learning community and a lot of algorithms have been evaluated on it. For instance, in (Breiman, L., 2001) or in (Geurts, P., et al. 2006), some results with this dataset can be found for algorithms combining decision trees (Adaboost, Random Forests, ...).

The Waveform 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 and the proportion of positive and negative examples is balanced. This dataset comes from an artificial problem where three different triangular functions (named either 1, 2, or 3) are defined by means of 21 real-valued attributes. For more detail on this dataset, please, refer to (Breiman, L. et al., 1984).

Experiments

In order to correctly measure the error rate, the dataset is decomposed into two subsets: the training set composed of 3500 examples, and the test set composed of 1500 examples.

Using the training set Forest of FDTs of different sizes (ranging from 1 to 500 FDTs) are built using a similar protocol as the one used for the video indexing application:

- *step 1*: a class *c* is chosen from the set of classes
- *step 2*: the training set is sampled by taking all the examples associated with the class *c*, and a random sample of examples of the other classes (i.e. negative examples). The idea here is to build a set of examples where there is the same number of examples of the class *c*, than examples of another class.
- *step 3*: from this sampling, a FDT is constructed using the Salammbô software (Marsala, C., & Bouchon-Meunier, B., 1999).

This process is repeated for each of the three classes in order to obtain three FDT, each one enabling the classification of an example with regards to a given class.

In the evaluation step, for each class, each example from the test set was classified by each of the FDTs. The classification was repeated three times, each time using the decision in a different manner: classical, fuzzy using the Zadeh operators and fuzzy using the Lukasiewicz ones. The individual tree classification degrees were then aggregated using a simple vote approach, to determine the final class of the example.

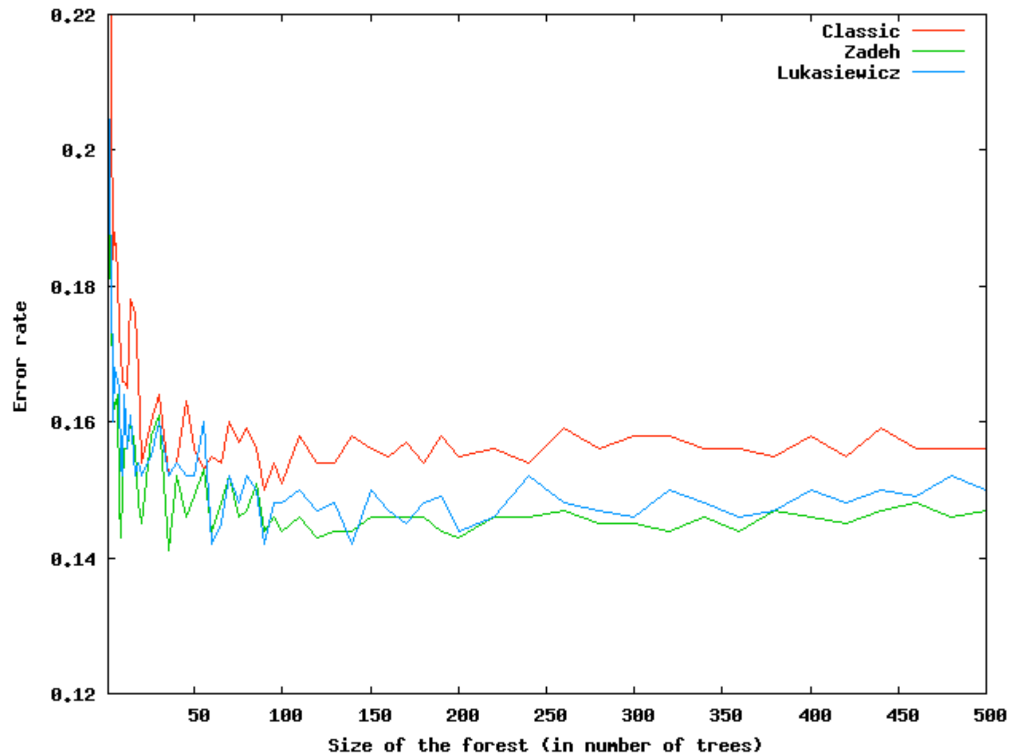

Figure 2: Influence of the size of the forest on the error rate

In Figure 2, we present the variation of the error rate when classifying the test set for various sizes of FFDT (in terms of number of trees). We notice that no matter how we use the FDTs (i.e. classically or fuzzy) the error rate decreases with the size of the forest.

Moreover, we notice that the error rate has great variations for small sized forests and stabilizes for larger ones. There seems to be a boundary error rate of around 0.15 (15% examples are badly classified). These results confirm the intuition: the more the number of trees is the lower the error rate. However, we remark that there is a limit to this approach. In fact, after a certain number of trees the results do not improve and we can even notice a slight worsening. It is also remarkable that a relatively small number of trees (for this problem around 100) is needed to get the limit performance.

Now, when comparing the classical use of the FDTs (curve labeled "classical) with the fuzzy-logic-based use of the same FDTs (curves labeled Zadeh and Lukasiewicz), it becomes clear that the use of the fuzzy set theory reduces the error rate for this problem, and this no matter the size of the forest. When comparing the fuzzy approaches we notice slight advantage for Zadeh's logic.

The complexity and runtime of the whole process is relatively low. In fact, the total runtime of all 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 Gb RAM, with GNU/Linux 2.6). This can be explained by the fact that the construction of a FDT was optimized in previous works and here the construction of the Forest of FDTs is just related to the number of trees built and, thus, is relatively low (taking into account the small number of trees needed to obtain a small error rate).

HIGH SCALE MINING ON TV VIDEO DATA

In order to compare our approach to others high scale approaches in a real-world framework, we participated to the high-level feature extraction task, at the TRECVid 2007 Challenge (Over, P., et al., 2007). Here, we only report the results obtained with our submission (Marsala, C., et al., 2007), the interested reader could refer to the proceedings of the TRECVid 2007 Challenge to have a good overview of the results of the whole participating teams.

The video corpus was composed of 109 videos (around 30 minutes length each) and 18142 reference shots (shots were provided by (Petersohn, C., 2004)). The challenge addressed 39 concepts: sports (1), weather (3), office (5), meeting (6), desert (10), mountain (12), waterscape-waterfront (17), police security (23), military staff (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).

The evaluation process was independently conducted by the NIST institute. Since TRECVid is information retrieval oriented, and given the size of the test set, each participating team had to propose, for each high-level concept, a ranking of at most 2000 video shots from the test set, that contain each of the concepts.

Due to the high size of the test corpus, it is impossible to manually annotate all examples for each concept. Thus, the TRECVid evaluators propose to evaluate a sample of the selected (by the submissions) shots and based on that infer the average precision. Thus, official metric (NIST, 2006) used to evaluate the runs was the *Inferred Average Precision*. Evaluating methods by means of an inferred value is a well-known approach whenever the size of the corpus is too large to be fully handled.

TRECVid Experiments

Several kinds and sizes of forests were studied (and submitted). Here, we focus on two sizes of forests (25 FDTs and 35 FDTs) and on the use of fuzzy logic in the classification step (classical use versus Zadeh and Lukasiewicz uses). More precisely, four approaches are compared: results obtained by means of a forest of 25 FDTs used classically (“F25_Classic”), results of a forest of 25 FDTs used with the Zadeh's t-norms (“F25_Zadeh”), results of a forest of 35 FDTs used with the Zadeh's t-norms (“F35_Zadeh”), and the median of the results for all the participating teams (to TRECVID 2007).

In Figure 3, variations of the *Inferred Average Precision* (Inf. AP.) are presented. The *average precision* combines the ideas behind both precision and recall by considering the precision at different depths of a list. It gives a paramount importance to the first shots returned, but also considers the total number of correct shots returned. It can be observed that the FFDT performance highly depends on the kind of concepts to be recognize. It is greatly linked to the low level descriptors used to represent the shots. Some concepts are simple to learn (not only for the FFDTs): waterscape-waterfront (17), animal (26), computer TV screen (27), US flag (28), airplane (29), car (30), boat or ship (33). However, concepts, such as weather (3), desert (10), US flag (28), people marching (35), need better (specialized) descriptors in order to allow the FFDT to perform better.

In average the FFDTs ranked among the first half of all the approaches that participated to the challenge. When compared to the median FFDTs perform for some concepts and less good for other and this independently of its “difficulty” to be learned. FFDTs outperform for the complex concepts: police security (23), military personnel (24), explosion fire (36); and for the simpler ones: TV screen (27), airplane (29).

As shown on the Waveform dataset, the increase of number of trees in the forest improves the results. Here the Inf. AP. (Inferred Average Precision) of forests of 35 fuzzy trees outperforms forests of 25 fuzzy trees, for almost all concepts.

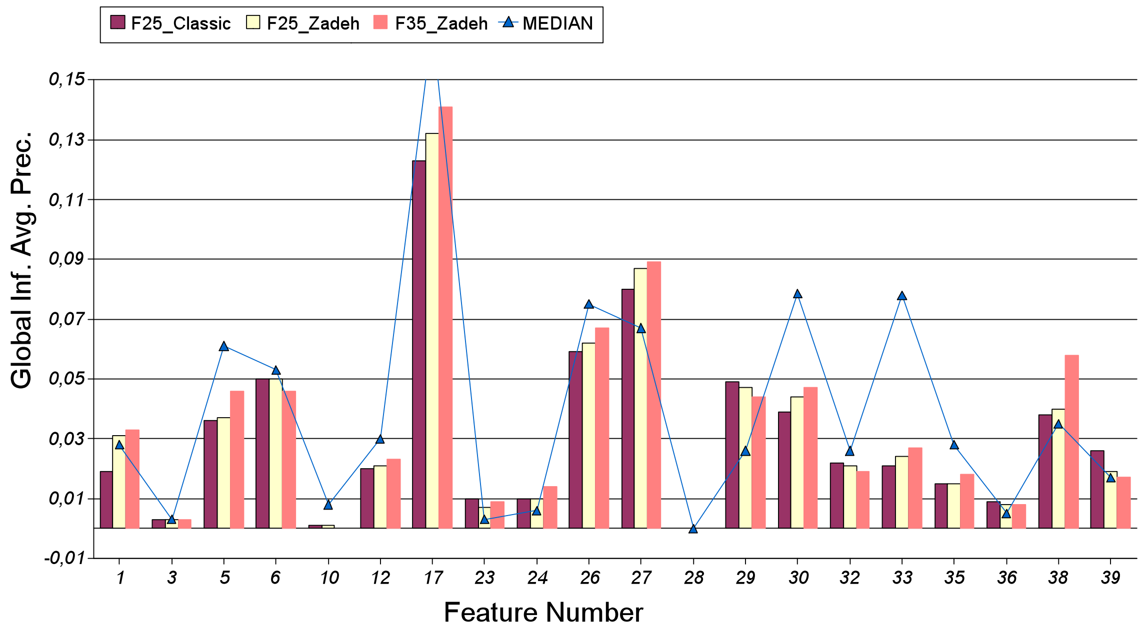

Figure 3: Global Inf. Avg. Precision

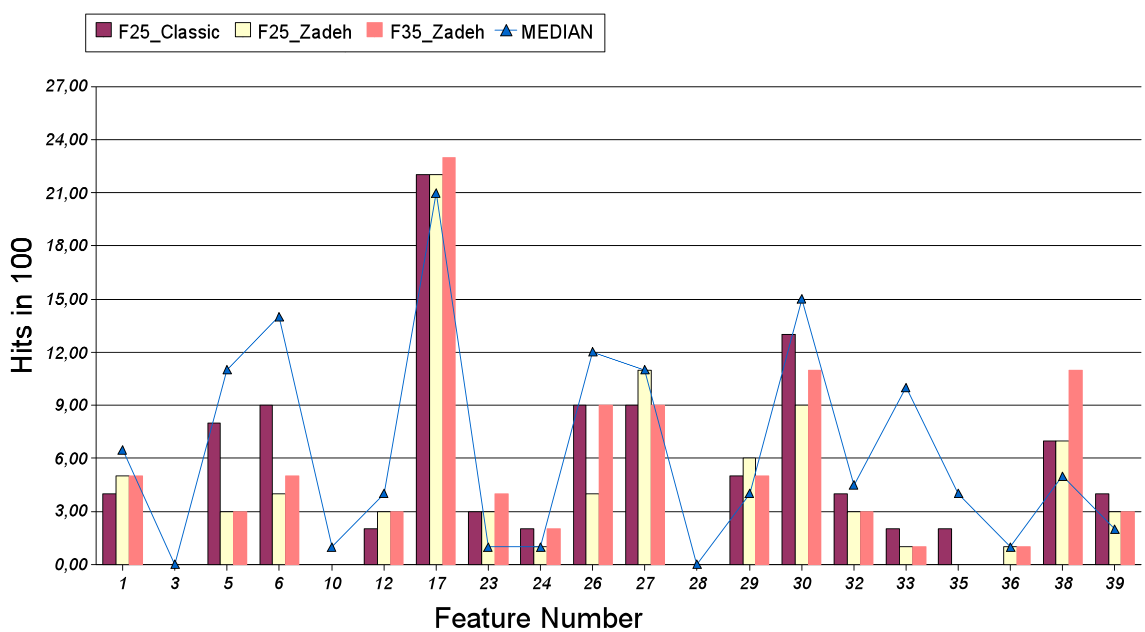

Figure 4: Good Hits in the 100 firsts

In Figure 4, Figure 5, and Figure 6, the number of correct classified shots (hits), for each concept, is presented when considering the first 100, 1000 and 2000 shots of the list, respectively. By considering the hits we do not take into account the order of the results. These 3 values, for each concept, are part of the evaluation metrics available at the TRECVid Challenge.

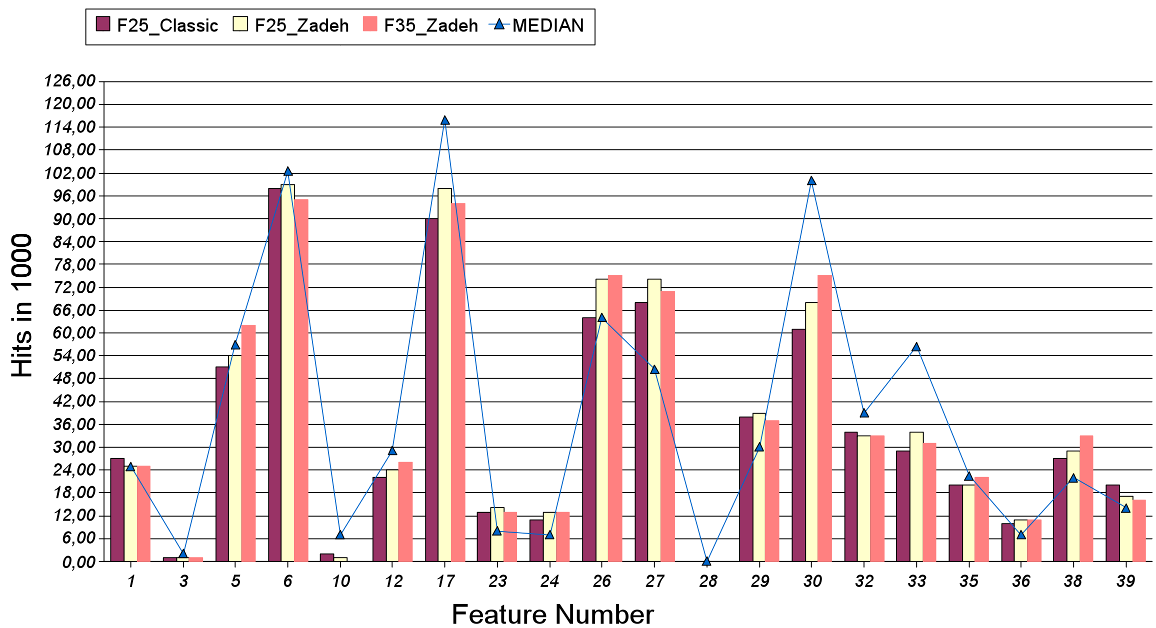

Figure 5: Good Hits in the 1000 firsts

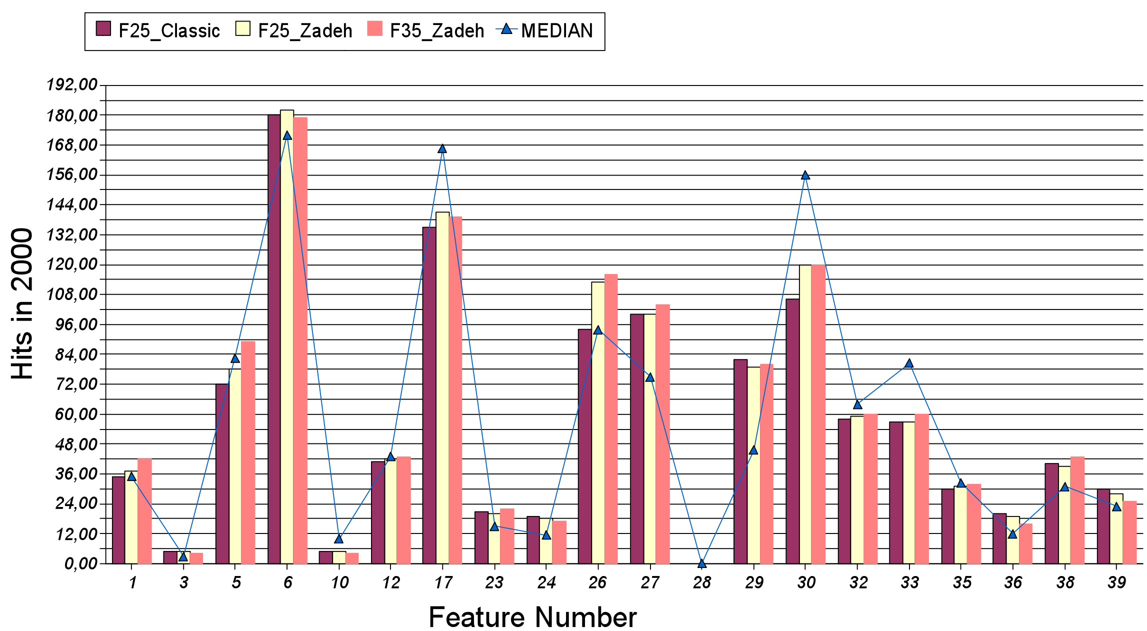

Figure 6: Good Hits in the 2000 firsts

When considering the number of hits in 100, 1000 and 2000 first shots for the FFDTs compared to the median (seen here as a reference point), we observe that the FFDT performs relatively better when considering most of the list. We claim that the main reason lies in the fact that the FFDT is a classification tool and not a ranking tool. In fact, the DTs optimize the decision boundary and the distance of the example to the boundary. If, during the learning stage, a shot is naturally put far away from the decision boundary (*i.e.* it is very easy to classify) then it will have a little influence on the selection of the boundary.

Based on the previous observation and in order to compare the different approaches it is better to choose the full returned list. On Figure 6, we observe that the use of the full power of the fuzzy set theory is always better than a classic-approach based. Moreover, we see again that the "F35_Zadeh" FFDT performs better than "F25_Zadeh" which highlights the importance of the size of the forest in this application too. Thus, confirming the results observed on the Waveform dataset. If we now focus our attention on the number of hits at 100 (Figure 4) and we compare the use of Fuzzy Logic ("F25_Zadeh") to the classical use ("F25_Classic"), it appears that in several times the latter outperforms the former. In other words, fuzzy logic is good for classification but it aggravates the ranking. This behavior can be explained by the fact that the use of degrees of truth on the one hand scrambles the strong decision (good for the top of the list), however, on the other hand it improves the overall decision (i.e. classification).

CONCLUSION

In this chapter, we presented the use of forests of fuzzy decision trees (FFDTs) for the high scale video mining problem. We showed that FFDTs can be considered as an interesting application of the fuzzy set theory to handle such a challenge.

In fact, we believe that, one effective way of addressing high scale data problems, with Fuzzy Decision Trees, is by splitting the problem by repeatedly sub-sampling the learning space and then for each sample train a classifier, leading to a Forest of Fuzzy Decision Trees.

Based on the carried out experiments performed on the well-studied Waveform data set and on the TRECVId real video data challenge, we advocate that a good heuristic leading to better results is to have as many FDT as possible. Moreover, we observed that the fuzzy of FFDTs outperforms the classical approach in a high scale classification problem.

The results on real world data (TV quality videos) from the TRECVid challenge highlight that this approach is already competitive with respect to others'. We show that FFDTs are good at detecting high-level concepts in shots (classification), but do not optimize the rank of the results.

In this real-world application that took place in a highly competitive context (the TRECVid Challenge that involved not only academic teams but also industrial teams) the tools from the fuzzy set theory have been proven to be a very sizeable and tractable approach. Moreover, the robustness of these tools when handling real-world measures enables the improvement of a classical data mining tools to construct fuzzy decision trees and build forests that benefit from the fuzzy degrees offer as output of the trees.

Several future works should be done in order to improve and to study better the proposed approach. For instance, the study of other kinds of descriptors to encode the video shots will be conducted in order to improve the results for other kinds of high-level concepts. Fuzzy descriptors could be introduced here (for instance, to build histograms defined on fuzzy colors, to define the boundaries of a shot, or to handle better the temporal measures related to the video) to take into account better this real-world data.

Moreover, several parameters that are used during the construction of the fuzzy decision trees, and the ones that are used to set the size of the forest deserve a deeper study. The study and the choice of the aggregation operators involved in various step of the use of the FFDT will also be studied deeper in order to be optimized for a given domain of application. The influence of the number of the FDTs to build a forest deserves also a deeper study that could brought out a better understanding of how to set a convenient size for such an ensemble of classifier.

REFERENCES

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

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

Bonissone, P.P., Cadenas, J. M., Garrido, M. C., & R. A. Diaz-Valladares, (2008). 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), (pp. 1231–1238), Malaga, Spain.

Breiman, L. (2001). Random forests. In Machine Learning, 45:5–32.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). Classification And Regression Trees. Chapman and Hall, New York.

Crockett, K., Bandar, Z., & McLean, D. (2001). Growing a fuzzy decision forest. In Proceedings of the 10th IEEE International Conference on Fuzzy Systems, (pp. 614–617) , Melbourne, Australia.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. In Machine Learning, 63(1):3–42.

Janikow, C. Z., & Faifer, M. (2000). Fuzzy decision forest. In Proceedings of the 19th International Conference of the North American Fuzzy Information Processing Society (NAFIPS'00), (pp. 218–221).

Marsala, C., & Bouchon-Meunier, B. (1997). Forest of fuzzy decision trees. In M. Mares, R. Mesiar, V. Novak, J. Ramik, & A. Stupnanova (Eds.), Proceedings of the Seventh International Fuzzy Systems Association World Congress, volume 1, (pp. 369–374), Prague, Czech Republic.

Marsala, C., & Bouchon-Meunier, B. (1999). An adaptable system to construct fuzzy decision trees. In Proc. Of the NAFIPS'99 (North American Fuzzy Information Processing Society), (pp. 223–227), New York, USA.

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

Marsala, C., & Detyniecki, M. (2006). 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://wwwnlpir.nist.gov/projects/tvpubs/tv.pubs.org.html.

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

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

Pan, J.-Y., & Faloutsos, C. (2002). VideoCube: A Novel Tool for Video Mining and Classification. In Lecture Notes in Computer Science, vol. 2555. (pp. 194-205).

Petersohn, C. (2004). Fraunhofer HHI at TRECVID 2004: Shot boundary detection system. Technical report, TR*EC Video Retrieval Evaluation Online Proceedings*, TRECVID. http://www-nlpir.nist.gov/projects/tvpubs/tvpapers04/fraunhofer.pdf.

Rosenfeld, C., Doerman, D., & DeMenthon, D. (2003). *Video Mining*. Kluwer Academic Publishers.

Zhu, X., Wu, X., Elmagarmid, A. K., Feng, Z., & Wu, L. (2005). Video Data Mining: Semantic Indexing and Event Detection form the Association Perspective. In IEEE Transactions on Knowledge and Data Engineering, vol. 17, nr. 5, (pp. 665-677).