Continuous Behaviour Knowledge Space For Semantic Indexing of Video Content

Fabrice Souvannavong and Benoit Huet
Département Communications Multimédia
Institut Eurécom
2229, route des Crêtes
06904 Sophia-Antipolis - France
(Fabrice.Souvannavong, Benoit.Huet)@eurecom.fr

Abstract - ***In this paper we introduce a new method for fusing classifier outputs. It is inspired from the behavior knowledge space model with the extra ability to work on continuous input values. This property allows to deal with heterogeneous classifiers and in particular it does not require to make any decision at the classifier level. We propose to build a set of units, defining a knowledge space, with respect to classifier output spaces. A new sample is then classified with respect to the unit it belongs to and some statistics computed on each unit. Several methods to create cells and make the final decision are proposed and compared to k-nearest neighbor and decision tree schemas. The evaluation is conducted on the task of video content retrieval which will reveal the efficiency of our approach.***

Keywords: classifier fusion, behavior knowledge space, video content indexing and retrieval

1 Introduction

Classification is a major task in many applications from multi-modal speech recognition to medical image analysis. It is a particularly important step for automatic semantic-based video content indexing and retrieval; especially to bridge the gap between low level features, like color or texture descriptors, and high level concepts, like mountain or people running. Unfortunately, the complexity of the task renders the choice of right descriptors and classification models very challenging. And the curse of dimensionality [1] renders the training of a single model inefficient. Mainly, two fusion strategies were set up to get around these. Either descriptors are first pre-processed to simplify their joint description [22] (descriptor fusion), either one or many classifiers are build per descriptor and fused later on in the process (classifier fusion).

Fusion is an old but still very active research field and it is nowadays receiving increasingly interest to improve classification performance. It has proved high potential in many applications from medical image analysis to audio-visual speech recognition. However, fusion is involved in all stages of the classification process and because of this, its mechanisms are difficult to understand. This challenging task is generally tackled on specific sub-problems such as descriptor fusion [22], classifier ensemble creation [5] or classifier fusion [14]. We focus our attention on classifier fusion that has the advantage to be very flexible despite the loss of correlation information between descriptors. Thus, we assume that an ensemble of classifiers is available. The fusion mechanism has to make a decision with respect to classifier outputs.

Classifier fusion methods can be divided into four families, namely classifier selection methods, probabilistic, possibilistic or belief methods, classification methods and statistical methods. First methods aim at making a static or dynamic selection of the best classifier with respect to the context [12, 2, 16]. A simple example of a static method is to divide the input space into cells to which are associated a reliable classifier. The second family of methods offer a set of operators to be applied on classifiers outputs. Operators include the sum, maximum, minimum, mean, majority voting, ... They were studied in [11, 13, 6]. In that case, classifier outputs are interpreted differently depending on the framework (probabilities [11], possibilities [6] or beliefs [9]). The difficulty is then, to find optimal operators. Third methods consist in solving another classification problem. However, the situation is particular since the distribution of classifier outputs is concentrated around two values, usually 0 *or* -1 *and* 1. The variance is, then, very low and some classification models are not appropriate [14]. Fourth methods rely on a knowledge space that drives the behavior of the classification. These methods are called Behavior Knowledge Space (BKS) methods [8] and they have the property to avoid making any assumption on classifier output distributions. In this paper, we propose to extend them to continuous classifier outputs.

BKS methods were originally designed to deal with a multi-class problem. A cell is defined as a combination of classifier decisions and a training set is used to compute the link between cells and the final decision. This approach requires a decision to be first taken by classifiers before the fusion. In order to avoid this early decision procedure that results in a loss of information, we introduce Continuous Behavior Knowledge Space that works directly on continuous classifier outputs.

The paper is organized as follows. We first present the Behavior Knowledge Space model. Then, we propose our extension to continuous classifier outputs and evaluate this new approach in the framework of semantic video content indexing and retrieval. Finally, the last section draws our conclusion.

2 Behavior Knowledge Space

2.1 Problem formulation

Let $\mathcal{X}$ be the descriptor space, $C = \{C_i, i = 1 : N_c\}$ be the output classes and $\{e_k, k = 1 : N_e\}$ the expert ensemble. e_k is a function $e_k : \mathcal{X} \rightarrow C$ such that if $x \in \mathcal{X}$ is a unknown pattern, then $e_k(x) = c_i$ means that x belongs to the class c_i. Let y_x denote the true class of x. The research focus of classifier fusion is then to find the function $F : C^{N_e} \rightarrow C$ that produces the best final decision.

2.2 BKS principle

The idea behind the Behavior Knowledge Space (BKS) model is to avoid making unjustified assumptions on the classifier ensemble such as classifier independency. For this purpose, the information is derived from a knowledge space which can concurrently record the decisions of all classifiers on each learned sample. BKS is a N_e dimensional space where each dimension corresponds to the decision of one classifier. A point in this space corresponds to the decision of all classifiers. Each point P contains three information, the total number of incoming samples P_i, the best representative class P_c and the total of samples from each class P_i^c. These statistics are computed on a training set, then they are used to make the final decision as follows:

$$F(P) = \begin{cases} P_c & , \textit{when } P_i > 0 \textit{ and } \frac{P_i^{P_c}}{P_i} \geq \lambda; \\ N_c + 1 & , \textit{otherwise}. \end{cases}$$

Where λ is a threshold that controls the reliability of the final decision, $P_i^{P_c}$ is the total of training samples from the class c in the unit and P_i is the total number of training samples in the unit.

3 Continuous Behavior Knowledge Space

3.1 Problem formulation

Let $\mathcal{X}$ be the descriptor space, $C = \{C_i, i = 1 : N_c\}$ be the output classes and $\{e'_k, k = 1 : N_e\}$ the expert ensemble. e'_k is a function $e'_k : \mathcal{X} \rightarrow \Re^{N_c}$ such that if $x \in \mathcal{X}$ is a unknown pattern, then $e'_k(x)$ measures the membership of x to each class c_i. The research focus of classifier fusion is then to find the function $F' : \Re^{N_e x N_c} \rightarrow C$ that produces the best final decision. Comparing to the original formulation any intermediate decision is taken and it is the role of F' to make the final decision with respect to a more accurate information.

3.2 CBKS principle

As for the BKS method, CBKS aims at making final decisions from an expert ensemble without making any assumption on this latter. Working directly with expert outputs has the advantage to avoid making preliminary decisions that remove the information on the decision quality. We propose a knowledge space where each dimension corresponds to the output of a given expert e_k for a given class c_i. The difficulty now is to find the right units. In this new space, units as described in the original method, are not well identified and they must be obtained using vector quantization techniques. We propose to compare two methods and some derivatives to get units: first uniform and non-uniform quantization by histogram computation, second k-means clustering. For each unit, the same statistics as BKS method are computed to make the final decision.

3.3 Histogram computation

We propose to determine unit boundaries by computing histograms. Uniform and non-uniform quantization strategies are studied.

The uniform quantization consists in splitting each dimension into s parts of the same length. It has the advantage to provide data independent units, therefore it is not sensitive to the training data set. However this approach is only optimal if data are uniformly distributed and it is rarely the case since we expect data to be distributed around some specific values (usually 0 or 1). Moreover, it is not scalable with respect to the number of dimension. Non-uniform quantization is then a better solution.

Unfortunately, it is a difficult problem on multiple dimensions. Non-uniform quantization is usually applied independently on each component (a given expert e_k for a given class c_i) thanks to histogram equalization technique. At the end, each bin has approximatively the same number of samples. Unfortunately, this simple approach discard the correlation information between dimension and it will simply fail on too many dimensions. We, thus, implement the well-known MHIST-p algorithm [15] that allows to efficiently build multi-dimensional histograms. It is an iterative process that loops over the two following steps until the desired number of units is obtained:

Step 1: We choose a unit U that contains a dimension d whose marginal distribution is the most in need of partitioning,

Step 2: Split U along d into a small number p of units.

The method has been proved efficient with $p = 2$ and the MaxDiff splitting method. It select the marginal distribution (U and d) with the largest difference in frequency values between adjacent values on a dimension. This approach does not fit to our problem since it approximates the probability density function of the data. Therefore, we propose a new splitting strategy that minimizes the squared error when elements of a unit are approximated by their mean value. Its selects the marginal distribution (U and d) with the highest squared error and splits the unit to minimize the quantification error:

Step 1: We choose a unit U that contains a dimension d where the squared error is the highest,

Step 2: Split U along d into a two units such that the quantification error is minimized.

This approach reminds the technique used to build decision trees and we find interesting to implement a goal oriented splitting strategy:

Step 1: We choose a unit U that contains a dimension d where the entropy reduction can be the highest,

Step 2: Split U along d into two units such that the entropy is minimized.

3.4 Clustering

Another intuitive way to determine units is given by clustering techniques. K-means clustering algorithm is applied on training data. In this case, data points are gathered by proximity. CBKS statistics are computed over each cluster. We also propose to apply a fuzzy approach to compute statistics and classify new samples. Units U_i are characterized by their center m_i. Training samples are clustered and we compute the probabilities $P_i(x)$ that the sample x belongs to units U_i. Statistics computed on units are then: the number of incoming samples $N_i = \sum_x P_i(x)$, the best representative class defined as $P_c = argmax_{c_i}(\sum_x P_i(x)|y_x = c_i)$, and the total number of samples from each class $P_i^c = \sum_x P_i(x)$ where $y_x = c_i$. Then, they are used to make the final decision as follows:

$$F'(P) = \begin{cases} P_c & , when\ P_i > 0\ and\ \frac{P_i^{P_c}}{P_i} \geq \lambda; \\ N_c + 1 & , otherwise. \end{cases}$$

Where λ is a threshold that controls the reliability of the final decision.

4 Video Content Retrieval

The continuous behavior knowledge space was designed for the task of video shot classification. The objective is to identify shot content for information retrieval purposes. The system is organized as depicted in figure 1. First the content of shot is captured by some descriptors presented in subsection 4.1. These descriptors are then processed as explained. From these descriptors and a reference database, classification models, presented in subsection 4.2, are built per semantic concept. Finally, the fusion mechanism is applied on classifier outputs to provide a detection score per concept. The evaluation is then presented in subsection 4.3.

4.1 Descriptors

It is far from trivial to identify the right features to extract for a general purpose application such as video content indexing. Many features were proposed in the literature during the last two decades. In some of our recent work on video content indexing [18], we proposed to use a region-based approach with color and texture information. At the end, an image vector space model (IVSM) is obtained to efficiently represent video shot content.

First key-frames of video shots, that are provided by TRECVID [20], are segmented thanks to the algorithm described in [7]. The algorithm is fast and provides visually acceptable segmentation. Its low computational requirement is an important criterion when we need to process a huge amount of data like the TRECVID database. Secondly, normalized HSV color histograms as well as mean and variance of 24 Gabor's filter response energies are computed for each region. Thirdly, the obtained vectors over the complete database are clustered to find the N most representative elements. The clustering algorithm used in our experiments is the well-known k-means. Representative elements are then used as visual keywords to describe video shot content. To do so, computed features on a single video shot are matched to their closest visual keyword with respect to the Euclidean distance or an other distance measure. Then, the occurrence vector of the visual keywords in the shot is build and this vector is called the raw signature of the shot.

The same process is applied on regions around salient points. They are detected thanks to the Haar wavelet transform as presented in [17]. The idea is to track and keep salient pixels at different scales. We then propose to build two rectangular regions around each salient point, one region on the left and the other on the right for vertical edges and one on the top and the other on the bottom for horizontal edges. The depth of rectangles is proportional to the scale level at which corresponding points were detected. We propose to have smaller rectangles for high frequencies. An illustration of both segmentation approaches is provided on figure 2.

(a) Region segmentation (b) Salient segmentation

Figure 2: Example of segmentation outputs.

Starting from an IVSM, one can build the occurrence matrix of visual keywords in training shots. The singular value decomposition of this matrix provides a new representation of video shot content where latent relationships can be emphasized. Image Latent Semantic Analysis (ILSA) is an adaptation of a method (Latent Semantic Indexing) used for text document indexing. It was originally introduced in [4] and it has now demonstrated its efficiency. In [19], the LSA is efficiently adapted into the so-called ILSA to deal with image content.

The number of singular values kept drives the ILSA performance. On one hand, if too many factors are kept, the noise will remain and the detection of synonyms and the polysemy of visual terms will fail. On the other hand, if too few factors are kept, important information will be lost, resulting in performance degradation. Unfortunately, no solution has yet been found and only numerous experiments allow to find the appropriate factor number.

Now, video shots have their visual content described by ILSA signatures on color, texture for both regions types (homogenious and salient). As far as the experiments reported in this paper are concerned, four signatures are used. The objective is then to deduce the semantic content of shots. Classification methods are appropriate tools for this task. They consist on automatically assigning labels to a given input vector. For this purpose a model is firstly created with respect to a training set. We propose to use support vector machines to solve our classification problem.

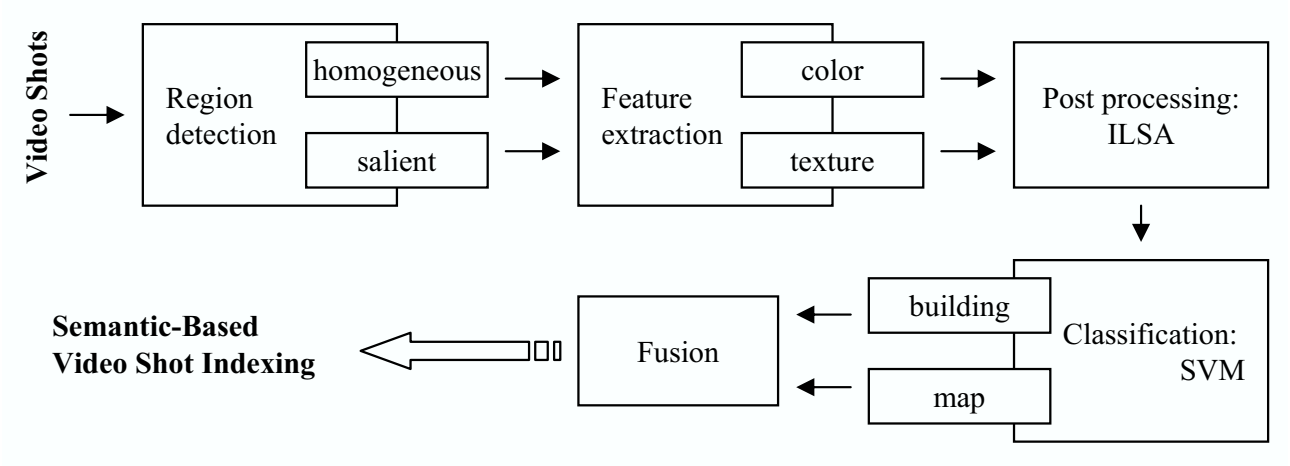

Figure 1: General framework of the application.

4.2 Classification

Support vector machines were widely used in the past ten years and they have been proved efficient in many classification applications. They have the property to allow a non linear separation of classes with very good generalization capacities. They were first introduced by Vapnik [21] for the text recognition task. The main idea is similar to the concept of a neuron: separate classes with a hyperplane. However, samples are indirectly mapped into a high dimensional space thanks to a kernel function that respects the Mercer's condition [3]. This allows one to lead the classification in a new space where samples are assumed to be linearly separable. To this end, we use the implementation SVMLight detailed in [10]. The selected kernel, denoted $K(.,.)$ is a radial basis function which normalization parameter σ is chosen depending on the performance obtained on a validation set. Let $\{sv_i\}, i = 1, ..., l$ be the support vectors and $\{\alpha_i\}, i = 1, ..., l$ corresponding weights. Then,

$$\mathcal{H}_{SVM}(x) = \sum_{k=1}^{k=l} \alpha_k K(x, sv_k)$$

We take advantage of the validation procedure to find both SVM parameters and the best number of factors to be kept by the ILSA.

4.3 Evaluation framework

In the framework of information retrieval, the system does not need to make a hard decision, i.e. to tell if a class is present or not, but only to provide a detection score that is useful to rank shots. For this purpose the CBKS method, as presented in the previous section, is slightly modified. First we assume that semantic concepts to be detected are independent. Classification and fusion are, then, conducted on binary problems, i.e. the concept is present in the shot or not. Second we don't really need to select an optimal value for λ that is required to make a decision. Thereby, we define the detection score per class c for a unit U_i as :

$$D_{i,c}(P) = \frac{P_i^{P_c}}{P_i}$$

Where $P_i^{P_c}$ is the total of training samples from the class c in the unit and P_i the total number of training samples in the unit. Finally, performance of the fusion is measured by the mean precision of retrieved shots, i.e. ordered shots with respect to detection scores.

5 Experiments

Experiments are conducted in the context of TRECVID 2005. Fusion algorithms are evaluated on the task of high-level feature extraction which aims at ordering shots with respect to their relevance to a semantic class. Proposed semantic classes in 2005 are building, car, fire/explosion, U.S. flag, map, mountain, sport, people walking/running and waterscape/waterfront. The quantitative evaluation is given by mean precision values of retrieval results limited to 2,000 retrieved shots. The training data set of TRECVID 2005 is composed of about 80 hours of news programs from American, Arabic and Chinese broadcasters. The set is split in three equal parts, chronologically by source, in order to train the SVM models, find the best fusion function and evaluate our system performance.

This section presents two types of experiments. First, CBKS method and its different quantization methods are studied. Then, CBKS performance is compared to k-nearest neighbor and decision tree performance.

5.1 CBKS study

In this section, we propose to compare the performance provided by the four quantization methods : uniform and MHIST-p presented in section 3.3 and the two version of k-means presented in section 3.4. Performance is presented with respect to the number of units in order to emphasize the effect of this parameter. Other quantization forms were also studied and their performance was not achieving as good as MHIST-p or k-means approaches.

Figure 3 shows the performance of CBKS when units are obtained by uniform quantization of the input space. We notice that only few bins are necessary : between 2 and 4 per input, i.e. between 16 and 256 units. Differences of performance with respect to the number of bins can be high and

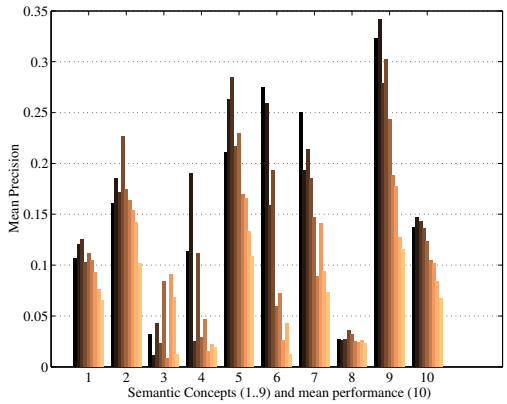

Figure 3: Uniform quantization, from 16 to 10.000 bins.

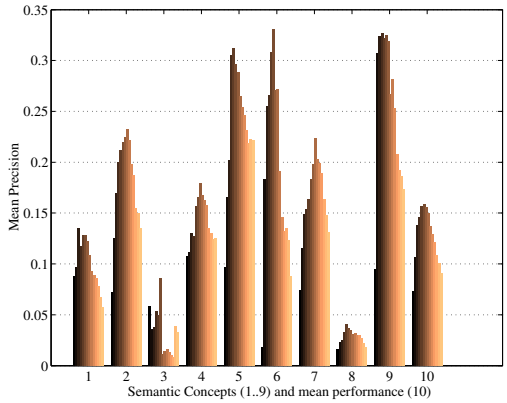

Figure 4: MHIST non-uniform histogram computation by minimizing squared error, from 5 to 3.000 bins.

localized as it is the case for semantic concepts two, four and seven. A validation procedure is highly recommended through it won't insure good generalization properties. The proposed MHIST quantization procedure allows to obtain more stable performance (see figure 4) with respect to the number of bins. It means that we can expect better generalization capacities contrary to the previous method. Furthermore, performance is very similar or higher if we have a look in details.

K-means algorithm is another way to create units. Figure 5 shows the performance when units are obtained thanks to it. As we can see, performance is highly improved for most of features. We also proposed to compute detection scores with respect to the distance to all units in order to soften the classification procedure. This method revealed efficient for most features (figure 6).

To conclude this part, CBKS is the most efficient when units are computed with k-means algorithm and the fuzzy approach is used to compute classification scores. However, this method does not lead to the best performance in some few cases (concepts three (car), four (US flag) and seven (sports)). It is mainly due to particular conditions that allows other methods to provide very good results. We think that these results are marginal since they correspond to isolated picks. In these particular cases, we do not expect good generalization properties.

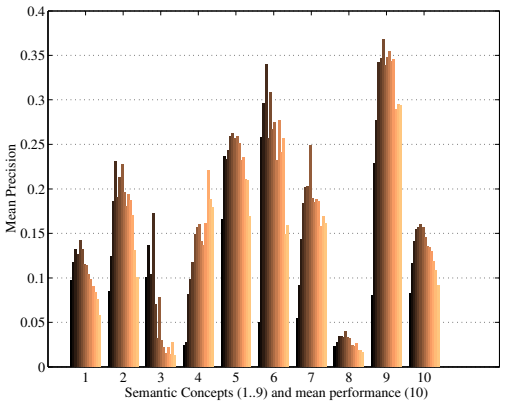

Figure 5: K-means clustering, from 5 to 3.000 clusters.

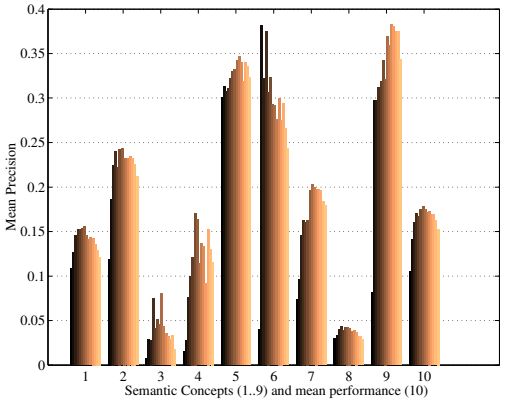

Figure 6: K-means clustering and fuzzy statistics, from 5 to 3.000 clusters.

5.2 Comparison of CBKS, K-NN and decision trees

This subsection compares our fusion method to related existing fusion techniques using the semantic ground-truth, namely k-nearest neighbors and decision trees. Figure 7 shows that many neighbors (around 500) are required for an optimal fusion, and still, it does not lead to the best fusion system. Results provided by decision tree are worst (see figure 8). We notice that the over-fitting phenomenon occurs quickly.

The difficulty of the task and the limited number of posi-

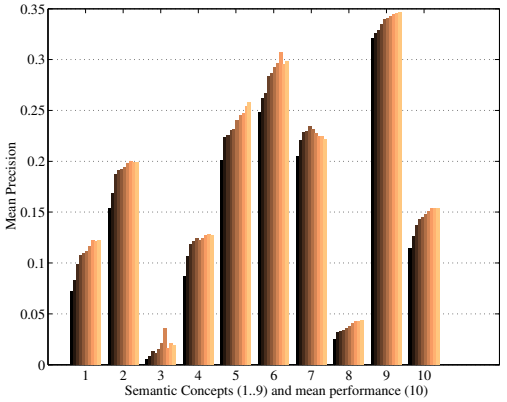

Figure 7: KNN fusion, from 25 to 800 neighbors.

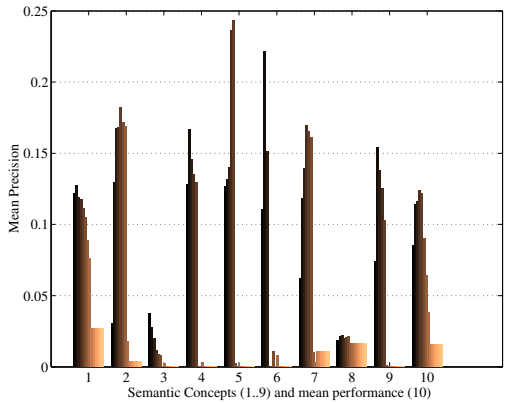

Figure 8: Decision Tree fusion, from 5 to 3.000 decisions.

tive samples make the fusion very challenging. Data driven quantization procedure are more efficient when the semantic ground-truth is not used. Indeed, it results in better generalization capacities due to the averaging operation applied on units.

6 Conclusion

We presented a new classifier fusion method, named continuous behavior knowledge space (CBKS). It is mainly designed for binary classification tasks (or semantic concept detection) while taking the original idea of behavior knowledge space. CBKS is an efficient fusion technique that builds a knowledge space, composed of units, with respect to the data distribution. The system automatically identifies units of interest on which classification scores are computed. The advantage is to avoid making any assumption on the classifier ensemble. Then, it does not required decision making by classification systems of the first level. Thus, it avoids to impose an early trade-off between false positives and true positives.

We proposed different strategies to compute optimal units and k-means algorithms, especially the fuzzy version, provides the best performance. Moreover, obtained performance is stable with respect to the number of bins and we expect good generalization properties. We further evaluated our system by comparing its performance to K-NN and decision tree methods and our system revealed its efficiency to deal with a complex problem where the number of positive samples is limited.

The next step is to address multi-class problems in order to study the capacity of CBKS to take advantage of inter-class relationships. Another direction is to look at the reliability of the method when many features are involved.

Acknowledgement

The work presented here was funded by France Telecom R&D, France.

References

[1] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is “nearest neighbor” meaningful? *Lecture Notes in Computer Science*, 1540:217–235, 1999.

[2] E. Brill and J. Wu. Classifier combination for improved lexical disambiguation. In *Proceedings of the 17th international conference on Computational linguistics*, pages 191–195, Morristown, NJ, USA, 1998. Association for Computational Linguistics.

[3] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*, chapter Kernel-Induced Feature Spaces. Cambridge University Press, 2000.

[4] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[5] T.G. Dietterich. Ensemble methods in machine learning. *Lecture Notes in Computer Science*, 1857:1–15, 2000.

[6] D. Dubois and H. Prade. Possibility theory and its applications: a retrospective and prospective view. In *Proceedings of IEEE ICFS*, volume 1, pages 5–11, 2003.

[7] P. Felzenszwalb and D. Huttenlocher. Efficiently computing a good segmentation. In *Proceedings of IEEE CVPR*, pages 98–104, 1998.

[8] Y.S. Huang and C.Y. Suen. A method of combining multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Trans. PAMI*, 17(1):90–94, 1995.

[9] R.A. Hummel and M.S. Landy. A statistical viewpoint on the theory of evidence. *IEEE Trans. PAMI*, 10(2):235–247, 1988.

[10] T. Joachims. *Advances in Kernel Methods - Support Vector Learning*, chapter 11 (Making large-Scale SVM Learning Practical). MIT Press, 1999.

[11] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas. On combining classifiers. *IEEE Trans. PAMI*, 20(3):226–239, 1998.

[12] L.I. Kuncheva. Switching between selection and fusion in combining classifiers: An experiment. *IEEE Transactions On Systems Man And Cybernetics, Part B-cybernetics*, 32(2):146–156, 2002.

[13] L.I. Kuncheva. A theoretical study on six classifier fusion strategies. *IEEE Trans. PAMI*, 24(2):281–286, february 2002.

[14] L.I. Kuncheva, J.C. Bezdek, and R.P.W. Duin. Decision templates for multiple classifier fusion. *Pattern Recognition*, 34(2):299–314, 2001.

[15] V. Poosala and Y.E. Ioannidis. Selectivity estimation without the attribute value independence assumption. In *23rd International Conference on Very Large Databases*, pages 486–495, 1997.

[16] F. Provost and T. Fawcett. Robust classification for imprecise environments. *Mach. Learn.*, 42(3):203–231, 2001.

[17] N. Sebe and M.S. Lew. Salient points for content-based retrieval. In *British Machine Vision Conference (BMVC'01)*, pages 401–410, 2001.

[18] F. Souvannavong, B. Merialdo, and B. Huet. Video content modeling with latent semantic analysis. In *3rd International Workshop on Content-Based Multimedia Indexing (CMBI'03)*, 2003.

[19] F. Souvannavong, B. Merialdo, and B. Huet. Latent semantic analysis for an effective region-based video shot retrieval system. In *6th ACM SIGMM International Workshop on Multimedia Information Retrieval, ACM Multimedia*, 2004.

[20] TRECVID. Digital video retrieval at NIST. http://www-nlpir.nist.gov/projects/trecvid/.

[21] V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[22] Y. Wu, E. Y. Chang, K. C.-C. Chang, and John R. Smith. Optimal multimodal fusion for multimedia data analysis. In *Proceedings of ACM MM*, pages 572–579, 2004.