Performance Analysis of Multiple Classifier Fusion for Semantic Video Content Indexing and Retrieval

Rachid Benmokhtar and Benoit Huet

Département Communications Multimédias
Institut Eurécom
2229, route des crêtes
06904 Sophia-Antipolis - France
(Rachid.Benmokhtar, Benoit.Huet)@eurecom.fr

Abstract. In this paper we compare a number of classifier fusion approaches within a complete and efficient framework for video shot indexing and retrieval[1]. The aim of the fusion stage of our sytem is to detect the semantic content of video shots based on classifiers output obtained from low level features. An overview of current research in classifier fusion is provided along with a comparative study of four combination methods. A novel training technique called Weighted Ten Folding based on Ten Folding principle is proposed for combining classifier. The experimental results conducted in the framework of the TrecVid'05 features extraction task report the efficiency of different combination methods and show the improvement provided by our proposed scheme.

1 Introduction

Multimedia digital documents are readily available, either through the Internet, private archives or digital video broadcast. Tools are required to efficiently index this huge amount of information and to allow effective retrieval operations. Unfortunately, most existing systems rely on the automatic description of the visual content through color, texture and shape features whereas users are more interested in the semantic multimedia content. In practice an important gap remains between the visual descriptors and the semantic content. New tools for automatic semantic video content indexing are highly awaited and an important effort is now conducted by the research community to automatically bridge the existing gap [1,2].

The retrieval of complex semantic concepts requires the analysis of many features per modalities. The task consisting of combining all these different parameters is far from trivial. The fusion mechanism can take place at different levels of the classification process. Generally, it is either applied on signatures (feature fusion) or on classifier outputs (classifier fusion). Unfortunately, complex signatures obtained from fusion of features are difficult to analyze and it results in classifiers that are not well trained despite of the recent advances in machine learning. Therefore, the fusion of classifier outputs remains an important step of the classification task.

[1] This work is funded by France Télécom R&D under CRE 46134752.

T.-J. Cham et al. (Eds.): MMM 2007, LNCS 4351, Part I, pp. 517–526, 2007.
© Springer-Verlag Berlin Heidelberg 2007

This paper starts with an overview of our semantic video content indexing and retrieval system. It is followed by a brief description of state of the art combination methods and classifiers, including Gaussian Mixture Model, Neural Network and Decision Template. In an effort to evaluate their classification and fusion ability, the previously mentioned approaches have been implemented within our system along with a number of training schemes. Among the training scheme evaluated here, we propose an alternative to the Ten Folding approach; the Weighted Ten Folding. This study reports the efficiency of different combination methods and shows the improvement provided by our proposed scheme on the TrecVid'05 dataset. Finally, we conclude with a summary of the most important results provided by this study.

2 System Architecture

This section describes the workflow of the semantic feature extraction process that aims to detect the presence of semantic classes in video shots, such as building, car, U.S. flag, water, map, etc ...

First, key-frames of video shots, provided by TrecVid'05, are segmented into homogeneous regions thanks to the algorithm described in [3]. Secondly, color and texture are extracted for each region obtained from the segmentation. 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.

Then, the occurrence vector of the visual keywords in the shot is build and this vector is called the Image Vector Space Model (IVSM) signature of the shot. Image latent semantic analysis (ILSA) is applied on these features to obtain an efficient and compact representation of video shot content. Finally, support vector machines (SVM) are used to obtain the first level classification which output will then be used by the fusion mechanism [4]. The overall chain is presented in figure 1.

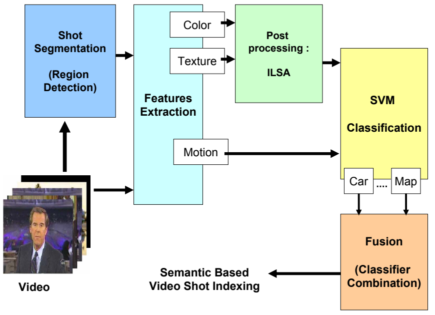

Fig. 1. General framework of the application

2.1 Visual Features Extraction

For the study presented in this paper we distinguish two types of visual modalities: HSV Histogram and Gabor filters features.Two visual features are selected for this purpose: Hue-Saturation-Value color histograms and energies of Gabor's filters [5]. In order to capture the local information in a way that reflects the human perception of the content, visual features are extracted on regions of segmented key-frames [6]. For the sake of computation complexity and storage requirements, region features are quantized and key-frames are represented by a count vector of quantization vectors.

2.2 ILSA

In [7], Latent Semantic Analysis was efficiently adapted from text document indexing to image content. The singular value decomposition of the occurrence matrix of visual keywords in some training shots provides a new representation of video shot content where latent relationships can be emphasized.

2.3 Classification

Classification consists in assigning classes to video shots given some description of its content. The visual content is extremely rich in semantic classes, but limited data is available to build classification models. Classification is therefore conducted on individual features in order to have enough training data with respect to input vector sizes. Allwein and al [8] showed that it was possible to transform a multi-classes classification problem into several binary classification problems. They propose a *one-against-all method*, which consists in building a system of binary classification by class. In our work, this method is adopted using the SVM classification.

Support Vector Machines are one of the most popular machine learning techniques, since they have shown very good generalization performance on many pattern classification problems. They have the property to allow a non linear separation of classes with very good generalization capacities. 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 [9]. This leads the classification in a new space where samples are assumed to be linearly separable. The selected kernel denoted $\mathcal{K}(.)$ is a radial basis function for which normalization parameter σ is chosen depending on the performance obtained on a validation set. The radial basis kernel is chosen for its good classification results comparing to Polynomial and Sigmoidal kernels [4].

3 Classifier Fusion

Combining classifier is an active research field [10,11]. There are generally two types of classifier combination: classifier selection and classifier fusion [10]. The classifier *selection* considers that each classifier is an expert in some local area of the feature space. The final decision is taken only by one classifier, as in [12], or more than one ”local expert”, as in [13]. Classifier *fusion* [14] assumes that all classifiers are trained over the whole feature space, and are considered as competitive as well as complementary. Duin and Tax [11] have distinguished the combination methods of different classifiers and the combination methods of weak classifiers.

The objective of the following section is to present an overview of classifier fusion methods and attempt to identify new trends that can be used in this area of research.

3.1 Non Trainable Combiners

Here, we detail the combiners that are ready to operate as soon as the classifiers are trained, i.e., they do not require any further training. The only methods to be applied to combine these results without learning are based on the principle of vote. They are commonly used in the context of handwritten text recognition [15]. All vote based methods can be derived from the majority rule E with threshold expressed by:

$$E = \begin{cases} C_i \text{ if } \max\left(\sum_i^K e_i\right) \geq \alpha K \\ \text{Rejection else} \end{cases} \tag{1}$$

where C_i is the i^{th} class, K is the number of classifiers to be combined and $e_i \in [0, 1]$ is the classifier output.

For $\alpha = 1$, the final class is assigned to the class label most represented among the classifier outputs else the final decision is rejected, this method is called **Majority Voting**. For $\alpha = 0.5$, it means that the final class is decided if more half of the classifiers proposed it, we are in **Absolute Majority**. For $\alpha = 0$, it is a **Simple Majority**, where the final decision is the class of the most proposed among K classifiers. In **Weighted Majority Voting**, the answer of every classifiers is weighted by a coefficient indicating their importance in the combination [16].

Soft label type classifiers combine measures which represent the confidence degree on the membership. In that case, the decision rule is given by the **Linear Methods** which consist in a linear combination of classifier outputs [17]:

$$E = \sum_{k=1}^{K} \beta_k m_i^k \tag{2}$$

where β_k is the coefficient which determines the attributed importance to k^{th} classifier in the combination and m_i^k is the answer for the class i.

3.2 Trainable Combiners

Contrary to the vote methods, many methods use a learning step to combine results. The training set can be used to adapt the combining classifiers to the classification problem. Now, we present four of the most effective methods of combination.

Neural Network (NN): Multilayer perceptron (MLP) networks trained by back propagation are among the most popular and versatile forms of neural network classifiers. In the work presented here, a multilayer perceptron networks with a single hidden layer and sigmoid activation function [18] is employed. The number of neurons contained in the hidden layer is calculated by heuristic.

Gaussian Mixture Models (GMM): The question with Gaussian Mixture Models is how to estimate the model parameter M. For a mixture of N components and a D dimensional random variable. In literature there exists two principal approaches for estimating the parameters: *Maximum Likelihood Estimation* and *Bayesian Estimation*. While there are strong theoretical and methodological arguments supporting Bayesian estimation, in this study the maximum likelihood estimation is selected for practical reasons. For each class, we trained a GMM with N components, using Expectation-Maximization (EM) algorithm.The number of components N corresponds to the model that best matches the training data. During the test, the class corresponding to the GMM that best fit the test data (according to the maximum likelihood criterion) is selected.

Decision Template (DT): The concepts of decision templates as a trainable aggregation rule was introduced by [10]. Decision Template DT_k for each class $k \in \Omega$ (where Ω is the number of classes) can be calculated by the average of the local classifier outputs $P_m^n(x)$.

$$DT_k(m,n) = \frac{\sum_{x \in T_k} P_m^n(x)}{Card(T_k)} \tag{3}$$

where T_k is a validation set different from the classifier training set. Decision Template is a matrix of size $[S, K]$ with S classifiers and K classes. To make the information fusion by arranging of K Decision Profiles (DP), it remains to determine which Decision Template is the most similar to the profile of the individual classification. Finally, the decision is taken by the maximum of the similarity difference.

Genetic Algorithm (GA): Genetic algorithms have been widely applied in many fields involving optimization problems. It is built on the principles of evolution via natural selection: an initial population (chromosomes encoding possible solutions) is created and by iterative application of genetic operators (selection, crossover, mutation) an optimal solution is reached, according to the defined fitness function [7].

3.3 Alternative Training Approaches

In the case of large sets of simple classifiers, the training is performed modified versions of the original dataset. Three heavily studied training alternatives are Adaboost (also known as boosting), Bagging (Bootstrapping), Random Subspaces and Ten Folding. In addition to the known methods, we propose an alternative to Ten Folding, which we call Weighted Ten Folding and is detailed at the end of this section.

Adaboost: The intuitive idea behind Adaboost is to train a series of classifiers and to iteratively focus on the hard training examples. The algorithm relies on continuously changing the weights of its training examples so that those that are frequently misclassified get higher and higher weights: this way, new classifiers that are added to the set are more likely to classify those hard examples correctly. In the end, Adaboost predicts one of the classes based on the sign of a linear combination of the weak classifiers trained at each step. The algorithm generates the coefficients that need to be used in this linear combination. The iteration number can be increased if we have time and with the overfitting risk [19].

Bagging: Bagging builds upon bootstrapping and adds the idea of aggregating concepts [20]. Bootstrapping is based on random sampling with replacement. Consequently, a classifier constructed on such a training set may have a better performance. Aggregating actually means combining classifiers. Often a combined classifier gives better results than individual base classifiers in the set, combining the advantages of the individual classifiers in the final classifier.

Ten Folding (TF): In front of the limitation (number of samples) of TrecVid'05 test set, *N-Fold Cross Validation* can be used to solve this problem. The principle of Ten Folding is to divide the data in $N = 10$ sets, where $N - 1$ sets are used for training data and the remaining to test data. Then, the next single set is chosen for test data and the remaining sets as training data, this selection process is repeated until all possible combination have been computed as shown in figure 2. The final decision is given by averaging the output of each model.

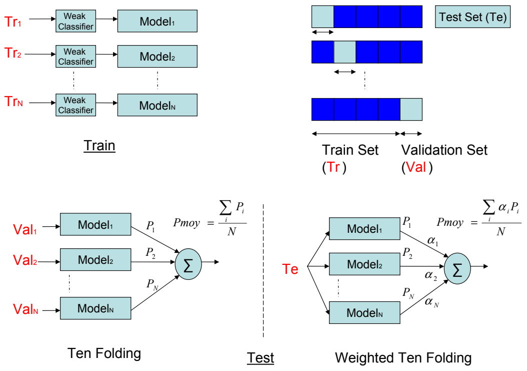

Fig. 2. The standard Ten Folding and Weighted Ten Folding combination classifier

Weighted Ten Folding (WTF): With TrecVid'05 test set limitation in mind, the well-known Bagging instability [20] (i.e. a small change in the training data produces a big change in the behavior of classifier) and the overfitting risk for Adaboost (i.e. when the iteration number is big [19]), we propose a new training method based on Ten Folding

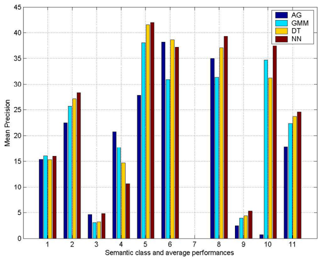

Fig. 3. Comparison of Genetic Algorithm, Decision Template method, GMM fusion method and Neural Network fusion method

that we call *Weighted Ten Folding*. We use the Ten Folding principle to train and obtain N models weighted by a coefficient indicating the importance in the combination. The weight α_i of each model is computed using the single set to obtain the training error ϵ_i. In this way, we obtain models with weak weight if the training error ϵ_i is high and models with high weight when ϵ_i is low.

$$\begin{cases} \epsilon_i = \sum_{j=1}^{N}(y(x_j) - f(x_j))^2 \\ \alpha_i = \frac{1}{2}\log(\frac{1-\epsilon_i}{\epsilon_i}) \end{cases} \tag{4}$$

The final decision combines measures which represent the confidence degree of each model. The weighted average decision in WTF improves the precision of Ten Folding by giving more importance for models with weak training error, contrary to the Ten Folding who takes the output average of each model with the same weight.

4 Experiments

Experiments are conducted on the TrecVid'05 databases [2]. It represents a total of over 85 hours of broadcast news videos from US, Chinese, and Arabic sources. About 60 hours are used to train the feature extraction system and the remaining for the evaluation purpose. The training set is divided into two subsets in order to train classifiers and subsequently the fusion parameters. The evaluation is realized in the context of TrecVid'05 and we use the common evaluation measure from the information retrieval community: the Average Precision.

The feature extraction task consists in retrieving shots expressing one of the following semantic concepts: *1:Building, 2:Car, 3:Explosion or Fire, 4:US flag, 5:Map, 6:Mountain, 7:Prisoner, 8:Sports, 9:People walking/running, 10:Waterscape*.

Figure 3 shows Mean Precision results for the trainable combiners. Of the four fusion scheme compared in this work, the Genetic Algorithm performs worst. This is clearly visible on the semantic concept (5, 10 and 11: Mean Average Precision), where the GA approach suffered from overfitting. The Decision Template and the Gaussian Mixture Model provide only marginally weaker performance than the Neural Network which performed best.

In the next experiment, Adaboost and Bagging principles are employed to increase the performances of GMM and Neural Network methods, considering them as weak classifier. As seen in figure 4, on average for all semantic concept the *Weighted Ten Folding* approach outperforms in turn boosting, bagging and Ten Folding technique in spite of the lack of datum. Significant improvement have been noticed for the following semantic concepts (4, 5, 6, 8 and 11:Mean Average Precision). This can be explained by the weight computation, which is computed on a validation set independently to training set. This allows to have more representative weights in the test for the whole classifier. So, we have best level-handedness of whole classifier contrary to boosting, where the weights computation is made by the training set.

Figure 5 consists in group of plots that represent the evolution of precision and recall values for 3 semantic concepts (Building, Car, Sports), using GMM and NN methods. We observe that the NN-based system has higher precision values for the ”Car” and ”Sports” concepts. These concepts present a rich motion information compared with ”Building” which have no motion. Similar poor results are obtained using ”Map” and ”Mountain” concepts. Therefore, the choice and the selection of features is very important and must be made by taking into account the behavior semantic concepts. In the same way, use audio features for ”Building, Map, US flag and Mountain” concepts will give no positive improvement, but it will be more beneficial for ”Explosion” and ”Sports” concept for example. A careful selection of the features is therefore necessary to improve our system such that it becomes more selective and less tolerant to changes. This question of features selection will be the object of our future works.

The table 1 presents the TrecVid'05 results submissions for [21], [22], [23] and our system. For this comparison task, we compute the Mean Average Precision (MAP) on the first 1000 retrieved shots as a measure of retrieval effectiveness. Our system presents very promising results, using SVMs classification and Weighted Ten Folding for NN Fusion. Models are trained per raw features and per concept. Looking at those results in some details, shows that the proposed system outperforms the top three systems for 6 of the 10 semantic concepts featured in TrecVid'05. Overall, the mean average precision is the best but only by a small (3%) improvement. We can explain this results by the system scheme classification, when we built a system of binary classification by class for each feature, it protects the correlation between the features. After, we fuse here response using neural network.

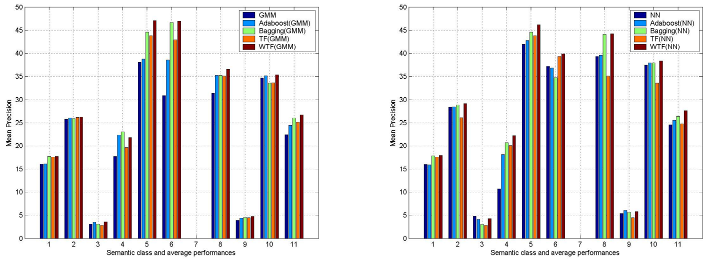

Fig. 4. Comparison of performance using Adaboost, Bagging, Ten Folding and Weighted Ten Folding for GMM and NN

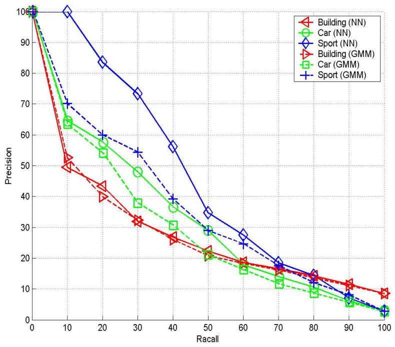

Fig. 5. Mean precision vs recall curves for three different objects (building, car, sports) using NN and GMM methods

Table 1. Mean Average Precision scores for TrecVid'05

Concepts	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	MAP
System A	39%	23.7%	2.8%	7.1%	15.1 %	18.4%	0%	31.2%	15.4 %	23.9%	17.66 %
System B	45%	27.9%	10.7%	24.6%	37.4%	37.8 %	2%	44.6%	27.5%	41.1%	29.86%
System C	47.6%	36.%	9.7%	18.7%	52.4%	45.4%	3%	40.1%	31.9%	47.6%	33.29%
Our System	**45.61%**	**48.49%**	**5.23%**	**38.49%**	**58.19%**	**50.43%**	**0%**	**38.08%**	**17.67%**	**58.89%**	**36.10%**

5 Conclusion

In this paper, we have presented an automatic semantic video content indexing and retrieval system where four different methods for combining classifiers are investigated in details. The Neural network based fusion approach managed all the features most effectively and appears therefore to be particularly well suited for the task of classifier fusion. Our newly proposed training scheme for combining weak classifiers, Weighted Ten Folding, achieved the best retrieval results. Adaboost and Bagging as they were originally proposed did not show a significant improvement, despite their special base model requirements for dynamic loss and prohibitive time complexity. It is due to the TrecVid'05 test set limitation and overfitting risk as the number of iteration increases. The later is solved by our proposed WTF which explains the performance improvement.

References

1. M. Naphade, T. Kristjansson, B. Frey, and T. Huang, "Probabilistic multimedia objects (multijets): a novel approach to video indexing and retrieval," *IEEE Trans. Image Process.*, vol. 3, pp. 536–540, 1998.
2. TRECVID, "Digital video retrieval at NIST," *http://www-nlpir.nist.gov/projects/trecvid/*.

3. P. Felzenszwalb and D. Huttenlocher, “Efficiently computing a good segmentation,” *Proceedings of IEEE CVPR*, pp. 98–104, 1998.
4. F. Souvannavong, “Indexation et recherche de plans video par contenu semantique,” Ph.D. dissertation, Phd thesis of Eurecom Institute, France, 2005.
5. W. Ma and H. Zhang, “Benchmarking of image features for content-based image retrieval,” *Thirtysecond Asilomar Conference on Signals, System and Computers*, pp. 253–257, 1998.
6. C. Carson, M. Thomas, and S. Belongie, “Blobworld: A system for region-based image indexing and retrieval,” *Third international conference on visual information systems*, 1999.
7. D. Souvannavong, B. Merialdo, and B. Huet, “Multi modal classifier fusion for video shot content retrieval,” *Proceedings of WIAMIS*, 2005.
8. E. Allwein, R. Schapire, and Y. Singer, “Reducing multiclass to binary : A unifying approach for margin classifiers.” *Journal of Machine Learning Research*, vol. 1, pp. 113–141, 2000.
9. N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge University Press, 2000, ch. Kernel-Induced Feature Spaces.
10. L. Kuncheva, J.C.Bezdek, and R. Duin, “Decision templates for multiple classifier fusion : an experiemental comparaison,” *Pattern Recognition*, vol. 34, pp. 299–314, 2001.
11. R. Duin and D. Tax, “Experiements with classifier combining rules,” *Proc. First Int. Workshop MCS 2000*, vol. 1857, pp. 16–29, 2000.
12. L. Rastrigin and R. Erenstein, “Method of collective recognition,” *Energoizdat*, 1982.
13. R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton, “Adaptive mixtures of local experts,” *Neural Computation*, vol. 3, pp. 1409–1431, 1991.
14. L. Xu, A. Krzyzak, and C. Suen, “Methods of combining multiple classifiers and their application to hardwriting recognition,” *IEEE Trans. Sys. Man. Cyb.*, vol. 22, pp. 418–435, 1992.
15. K. Chou, L. Tu, and I. Shyu, “Perfmrmances analysis of a multiple classifiers system for recognition of totally unconstrained handwritten numerals,” *4th International Workshop on Frontiers of Handwritten Recognition*, pp. 480–487, 1994.
16. B. Achermann and H. Bunke, “Combination of classifiers on the decision level for face recognition,” *technical repport of Bern University*, 1996.
17. T. Ho, “A theory of multiple classifier systems and its application to visual and word recognition,” Ph.D. dissertation, Phd thesis of New-York University, 1992.
18. G. Cybenko, “Approximations by superposition of a sigmoidal function,” *Mathematics of Control, Signal and Systems*, vol. 2, pp. 303–314, 1989.
19. Y. Freud and R. Schapire, “Experiments with a new boosting algorithms,” *Machine Learning: Proceedings of the 13th International Conference*, 1996.
20. M. Skurichina and R. Duin, “Bagging for linear classifiers,” *Pattern Recognition*, vol. 31, no. 7, pp. 909–930, 1998.
21. M. Cooper, J. Adcock, R. Chen, and H. Zhou, “Fxpal at trecvid 2005,” in *Proceedings of Trecvid*, 2005.
22. S.-F. Chang, W. Hsu, L. Kennedy, L. Xie, A. Yanagawa, E. Zavesky, and D. Zhang, “Video seach and high level feature extraction,” *Proceedings of Trecvid*, 2005.
23. A. Amir, J. Argillander, M. Campbell, A. Haubold, G. Iyengar, S. Ebadollahi, F. Kang, M. Naphade, A. Natsev, J. Smith, J. Tesic, and T. Volkmer, “Ibm research trecvid 2005 video retrieval system,” in *Proceedings of Trecvid*, 2005.