Classifier Fusion: Combination Methods For Semantic Indexing in Video Content

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. Classifier combination has been investigated as a new research field to improve recognition reliability by taking into account the complementarity between classifiers, in particular for automatic semantic-based video content indexing and retrieval. Many combination schemes have been proposed in the literature according to the type of information provided by each classifier as well as their training and adaptation abilities. This paper presents an overview of current research in classifier combination and a comparative study of a number of combination methods. A novel training technique called Weighted Ten Folding based on Ten Folding principle is proposed for combining classifier. Experiments are conducted in the framework of the TRECVID 2005 features extraction task that consists in ordering shots with respect to their relevance to a given class. Finally, we show the efficiency of different combination methods.

1 Introduction

With the development of multimedia devices, more and more videos are generated every day. Despite the fact that no tools are yet available to search and index multimedia data, many individual approaches have been proposed by the research community. Video content interpretation is a highly complex task which requires many features to be fused. However, it is not obvious how to fuse them. The fusion mechanism can be done at different levels of the classification. The fusion process may be applied either directly on signatures (feature fusion) or on classifier outputs (classifier fusion). The work presented in this paper focuses on the fusion of classifier outputs for semantic-based video content indexing.

Combination of multiple classifier decisions is a powerful method for increasing classification rates in difficult pattern recognition problems. To achieve better recognition rates, it has been found that in many applications, it is better to fuse multiple relatively simple classifiers than to build a single sophisticated classifier.

There are generally two types of classifier combination: classifier selection and classifier fusion [1]. The classifier *selection* considers that each classifier is an expert in some local area of the feature space. The final decision can be taken only by one classifier, as in [2], or more than one "local expert", as in [3]. Classifier *fusion* [4] assumes that all classifiers are trained over the whole feature space, and are considered as competitive as well as complementary. [5] has distinguished the combination methods of

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 65–74, 2006.

different classifiers and the combination methods of weak classifiers. Another kind of grouping using only the type of classifiers outputs (class, measure) is proposed in [4].

Jain [6] built a dichotomy according to two criteria of equal importance: the type of classifiers outputs and their capacity of learning. This last criteria is used by [1,7] for grouping the combination methods. The trainable combiners search and adapt the parameters in the combination. The non trainable combiners use the classifiers outputs without integrating another *a priori* information of each classifiers performances.

As shown in figure 1, information coming from the various classifiers are fused to obtain the final classification score. Gaussian mixture models, neural network and decision templates are implemented for this purpose and evaluated in the context of information retrieval.

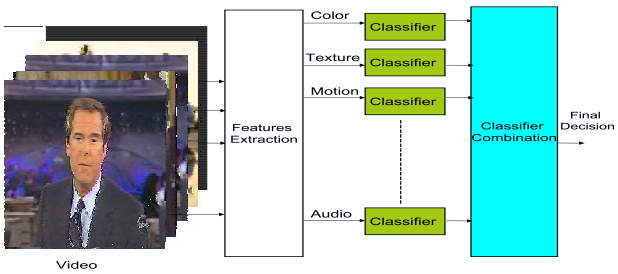

Fig. 1. General framework of the application

The paper presents the research we conducted toward a semantic video content indexing and retrieval system. It starts from a brief state of the art of existing combination methods and involved classifiers, including mixture of Gaussian, neural network and decision templates. All three methods are employed in turn to fuse and compared on the difficult task of semantic contents of video shots estimation . Then, we describe the visual and motion features that were selected. The results of our experiments in the framework of TRECVID 2005 are then presented and commented. Finally, we conclude with a summary of the most important results provided by this study along with some possible extension of work.

2 Combination of Different Classifiers

The classifiers may be of different nature, e.g. the combination of a neural network, a nearest neighbour classifier and a parametric decision rule, using the same feature space. This section starts by describing non-trainable combiners and continues with trainable ones.

2.1 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 [8]. All the methods of votes 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 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 there importance in the combination [9].

The classifiers of type soft label outputs combine measures which represent the confidence degree on the membership. In that case, the decision rule is given by the **Linear Methods** which consist simply in applying to the outputs classifiers a linear Combination [10]:

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

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.

2.2 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 [11] is employed. The number of neurons contained in the hidden layer is calculated by heuristic. A description of the feature vectors given to the input layer is given in section 4.

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 [12]. The number of components N corresponds to the model

that best matches the training data. The likelihood function of conditional density models is:

$$p(x; M) = \sum_{i=1}^{N} \alpha_i \mathcal{N}(\mu_i, \Sigma_i)(x) \tag{3}$$

where α_i is the weight of the i^{th} component and $\mathcal{N}(.)$ is the Gaussian probability density function with mean μ_i and covariance Σ_i.

$$\mathcal{N}(\mu_i, \Sigma_i)(x) = \frac{1}{\sqrt{(2\pi)^D |\Sigma_i|}} \exp\left(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)\right) \tag{4}$$

During the test, the class corresponding to the GMM that best fit the test data (according to the maximum likelihood criterion) is selected.

Decision Templates (DT). The concepts of decision templates as a trainable aggregation rule was introduced by [1,7]. 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 $P_m^n(x)$.

$$DT_k(m,n) = \frac{\sum_{x \in T_k} P_m^n(x)}{Card(T_k)} \tag{5}$$

where T_k is a validation set different from the classifier training set. Decision Templates 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.

Several similarity measures can be used, e.g., the Mahalanobis norm (equ.6) and Swain & Ballard (equ.7) or the Euclidian distance (equ.8).

$$Sim(DP(x_i), DT^k) = (\sum_{m,n=1}^{S,K} (DP(x_i)_{m,n} - DT_{m,n}^k))^T \Sigma^{-1} (\sum_{m,n=1}^{S,K} (DP(x_i)_{m,n} - DT_{m,n}^k)) \tag{6}$$

where: $m = 1, ..., S, n = 1, ..., K$ and Σ is the Covariance matrix.

$$Sim(DP(x_i), DT^k) = \frac{\sum_{m,n=1}^{S,K} \min(DP(x_i)_{m,n}, DT_{m,n}^k)}{\sum_{m,n=1}^{S,K} (DT_{m,n}^k)} \tag{7}$$

$$Sim(DP(x_i), DT^k) = 1 - \frac{1}{SK} \sum_{m=1}^{S} \sum_{n=1}^{K} (DP(x_i)_{m,n} - DT_{m,n}^k) \tag{8}$$

Finally, the decision is taken by the maximum of the similarity difference.

Genetic Algorithm (GA). Genetic algorithm have been widely applied in many fields involving optimization problems. It is built on the principles of evolution via natural selection: an initial population of individuals (chromosomes encoding the possible solutions) is created and by iterative application of the genetic operators (selection, crossover, mutation) an optimal solution is reached, according to the defined fitness function [13].

3 Combination of Weak Classifiers

In this case, large sets of simple classifiers are trained on modified versions of the original dataset. The three most heavily studied approaches are outlined here: reweighting the data (boosting-Adaboost), bootstrapping (bagging) and using random subspaces. Then, we introduce a new training method inspired from Ten Folding.

3.1 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 overfiting risk [14].

3.2 Bagging

Bagging builds upon bootstrapping and add the idea of aggregating concepts [15]. 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.

3.3 Random Subspace (RS)

The Random Subspace method consists to modify the learning data as in Bagging and Boosting. However, this modifications are realized on the features space. [15] showed that *RS* method allows to maintain a weak learning error and to improve the generalization error for the linear classifiers. It noticed that this method can outperform than the bagging and boosting if the number of features is big.

3.4 Ten Folding Training Approaches

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.

Weighted Ten Folding (WTF). With TrecVid'05 test set limitation in mind, the well-known Bagging instability [15] (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 [14]), we propose a new training method based on Ten Folding 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 of each model is computed using the single set. 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.

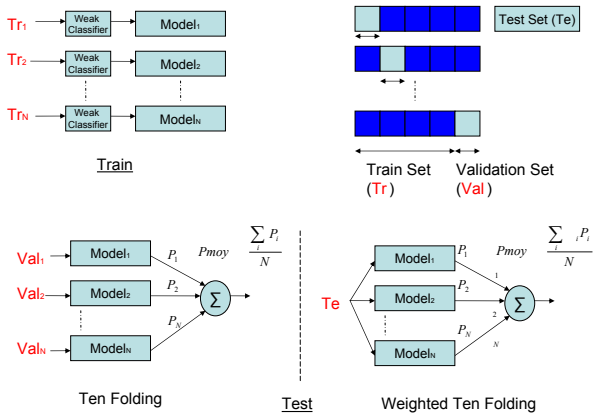

Fig. 2. The standard Ten Folding and Weighted Ten Folding combination classifier

4 Video Features

As far as this paper is concerned, we distinguish two types of modalities, visual and motion features, to represent video content.

4.1 Visual Features

To describe the visual content of a shot, features are extracted from key-frames. Two visual features are selected for this purpose: Hue-Saturation-Value color histograms and energies of Gabor's filters [16]. 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 [17]. Then, to have reasonable computation complexity and storage requirements, region features are quantized and key-frames are represented by a count vector of quantization vectors. At this stage, we introduce latent semantic indexing to obtain an efficient region based signature of shots. Finally, we combine the signature of the key-frame with the signatures of two extra frames in the shot, as it is described in [18], to get a more robust signature.

4.2 Motion Features

For some concepts like people walking/running, sport, it is useful to have an information about the motion activity present in the shot. Two features are selected for this purpose: the camera motion and the motion histogram of the shot. For sake of fastness, these features are extracted from MPEG motion vectors. The algorithm presented in [19] is used to estimate the camera motion. The average camera motion over the shot is computed and subtracted from macro-block motion vectors to compute the 64 bins motion histogram of moving objects in a frame. Then, the average histogram is computed over frames of the shot.

5 Experiments and Discussion

Experiments are conducted on the TRECVID 2005 databases [20]. 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 and we use the common evaluation measure from the information retrieval community: the mean 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 presented in section (2.2), the NN improves the precision result for all semantic concept when compared with results obtained by Genetic Algorithm [18]. This improvement is clearly visible on the semantic concept (5, 10, 11: Mean Average Precision), where the GA approach had an overfitting problem which damaged the average precision.

Figure 4 shows the variation of Mean Average Precision results for Decision Templates using different norms (Swain & Ballard, Euclidean and Mahalonibis) for similarity

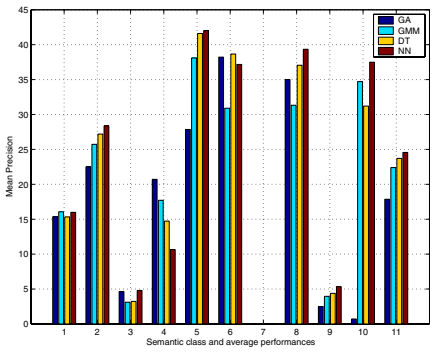

Fig. 3. Comparaison of Genetic Algorithm, Decision Templates method, GMM fusion method and Neural Network fusion method

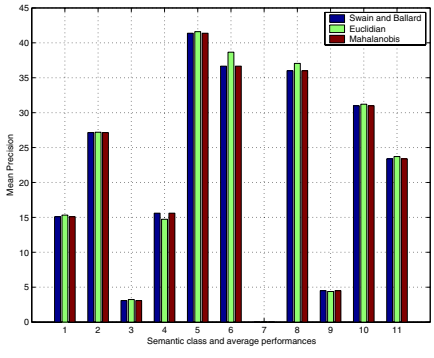

Fig. 4. Comparaison of Decision Templates performance with different norms (Swain & Ballard , Euclidean, Mahalanobis norm)

computation. Similar results are obtained for all three norms, which indicates that the Decision Templates method is more sensitive to data than to the chosen norm.

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 5, on average for all semantic concept the *WTF* 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, 8,11: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.

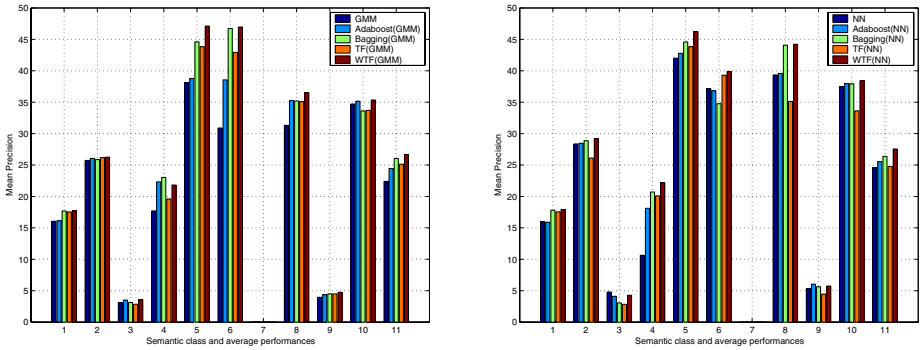

Fig. 5. Comparaison of performance using Adaboost, Bagging, Ten Folding and Weighted Ten Folding for GMM and NN

Fig. 6. Examples of first retrieved shots for waterscape, car and map classes

To conclude this section, figure 6 gives examples of first retrieved shots on TRECVID 2005 dataset for the classes waterscape, car and map to illustrate the efficiency of our classification method.

6 Conclusion

Fusion of classifiers is a promising research area, which allows the overall improvement of the system recognition performance. The work made on the combination also shows the multitude of combination methods which are different by their learning capacity and outputs classifier type.

Our experiments based on the TRECVID 2005 video database, show that Multilayer Neural Network and GMM approaches can improve the combination performance in comparison to the combination of multiple classifiers with averaging [21] and Genetic algorithm [13]. The results are very promising on the difficult problem of video shot content detection, using color, texture and motion features.

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 test set limitation and overfitting risk if the iteration number is big. The WTF resolves this last problem and improves Bagging and Adaboost results.

We have started to investigate the effect of the addition of many other visual features (Dominant Color, RGB, Canny edges features,...) as well as audio features (MFCC, PLP, FFT), to see their influence on the final result, and how the different approaches are able to deal with potentially irrelevant data. In parallel, we have initiated a program of work about descriptor fusion. We believe such an approach, which may be seen as normalization and dimensionality reduction [22], will have considerable effect on the overall performance of multimedia content analysis algorithms.

Acknowledgement

The work presented here is supported by the European Commission under contract FP6-027026-K-SPACE. This work is the view of the authors but not necessarily the view of the community.

References

1. 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.
2. L. Rastrigin and R. Erenstein, "Method of collective recognition," *Energoizdat*, 1982.
3. R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, pp. 1409–1431, 1991.
4. L. Xu, A. Krzyzak, and C. Suen, "Methods of combining multiple classifiers and their application to hardwriting recognition," *IEEE Trans.Sys.Man.Cyber*, vol. 22, pp. 418–435, 1992.
5. R. Duin and D. Tax, "Experiements with classifier combining rules," *Proc. First Int. Workshop MCS 2000*, vol. 1857, pp. 16–29, 2000.
6. A. Jain, R. Duin, and J. Mao, "Combination of weak classifiers," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, 2000.
7. L. Kuncheva, "Fuzzy versus nonfuzzy in combining classifiers designed by bossting," *IEEE Transactions on fuzzy systems*, vol. 11, no. 6, 2003.

8. K. Chou, L. Tu, and I. Shyu, "Performances 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.
9. B. Achermann and H. Bunke, "Combination of classifiers on the decision level for face recognition," *technical repport of Bern University*, 1996.
10. T. Ho, *A theory of multiple classifier systems and its application to visual and word recognition*. PhD thesis, Phd thesis of New-York University, 1992.
11. G. Cybenko, "Approximations by superposition of a sigmoidal function," *Mathematics of Control, Signal and Systems*, vol. 2, pp. 303–314, 1989.
12. P. Paalanen, J. Kamarainen, J. Ilonen, and H. Kalviainen, "Feature representation and discrimination based on gaussian mixture model probability densities," *Research Report, Lappeenranta University of Technology*, 1995.
13. F. Souvannavong, B. Merialdo, and B. Huet, "Multi modal classifier fusion for video shot content retrieval," *Proceedings of WIAMIS*, 2005.
14. Y. Freud and R. Schapire, "Experiments with a new boosting algorithms," *Machine Learning : Proceedings of the 13th International Conference*, 1996.
15. M. Skurichina and R. Duin, "Bagging for linear classifiers," *Pattern Recognition*, vol. 31, no. 7, pp. 909–930, 1998.
16. 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.
17. 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.
18. F. Souvannavong, B. Merialdo, and B. Huet, "Latent semantic analysis for an effective region based video shot retrieval system," *Proceedings of ACM MIR*, 2004.
19. R. Wang and T. Huang, "Fast camera motion analysis from mpeg domain," *Proceedings of IEEE ICIP*, pp. 691–694, 1999.
20. TRECVID, "Digital video retrieval at NIST," *http://www-nlpir.nist.gov/projects/trecvid/*.
21. S. 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.
22. Y. Y. C. Zheng, "Run time information fusion in speech recognition," *Proc. of ICSLP*, 2002.