Pattern Anal Applic (2014) 17:37–50
DOI 10.1007/s10044-013-0336-8

THEORETICAL ADVANCES

Performance evaluation of early and late fusion methods for generic semantics indexing

Yuan Dong · Shan Gao · Kun Tao · Jiqing Liu · Haila Wang

Received: 2 March 2011 / Accepted: 8 April 2013 / Published online: 17 April 2013

Abstract This paper focuses on the comparison between two fusion methods, namely early fusion and late fusion. The former fusion is carried out at kernel level, also known as multiple kernel learning, and in the latter, the modalities are fused through logistic regression at classifier score level. Two kinds of multilayer fusion structures, differing in the quantities of feature/kernel groups in a lower fusion layer, are constructed for early and late fusion systems, respectively. The goal of these fusion methods is to put each of various features into effect and mine redundant information of the combination of them, and then to develop a generic and robust semantic indexing system to bridge semantic gap between human concepts and these low-level visual features. Performance evaluated on both TRECVID2009 and TRECVID2010 datasets demonstrates that the systems with our proposed multilayer fusion methods at kernel level perform more stably to reach the goal than the classification-score-level fusion; the most effective and robust one with highest MAP score is constructed by early fusion with two-layer equally weighted composite kernel learning.

Y. Dong · S. Gao (✉) · J. Liu
Beijing University of Posts and Telecommunications, Beijing, People's Republic of China
e-mail: shannon.bupt@gmail.com

Y. Dong
e-mail: yuandong@bupt.edu.cn

K. Tao · H. Wang
France Telecom R&D Beijing Co., Ltd., Beijing, People's Republic of China
e-mail: kun.tao@orange-ftgroup.com

keywords Semantic indexing · Concept detection · Multiple kernel learning · Classifier-level fusion · Visual feature extraction

1 Introduction

Recently, the great increase of excellent multimedia products seems to have been the effects of the development of new Internet and TV services, as well as the multimedia-making, storage, and transmission capabilities. Picture albums and home video sharing services, e.g., *flickr*, *youtube*, and social networks, e.g., *facebook*, provide quantities of personal and personalized videos; *connected TV* makes a mass of TV products available over Internet. As these huge supply and video dissemination of unclassified multimedia information flooding us, the need for tools to filter, classify, search, and retrieve this content efficiently becomes more and more acute. Pushed by this demand, the field of multimedia indexing has witnessed a rapid growth, and powerful multimedia analysis techniques have emerged.

The task of semantic analysis and indexing aims to construct a mapping from machine computable low-level features to the high-level semantic interpretation. In an early stage, a variety of specific methods were developed over a limited number of concepts, like commercials [1], news anchor person [2], and baseball [3], which often sought to use specific low-level features directly to perform matching of video clips in one way or another and then ranked the retrieved clips in terms of some simple similarity measure or heuristic rules. These approaches were suited to low-level search, but had limitations when it comes to large-scale automated annotation of video archives. High-level similarity may not correspond to

low-level feature-based similarity if there is no attempt to understand the semantics of the query.

The addressed issues can be solved by a generic semantic analysis system [4, 5], which can learn a wider variety of semantic concepts by using multimodal analysis with a shared set of low-level features, and make use of the machine learning methods to learn the concept classification rules other than simple similarity measure. Enough empirical evidence states that multiple features make an exhaustive representation of the content of videos, and powerful learning algorithms and effective fusion methods of inference capability bridge the semantic gap to some extent and lead a significant improvement in detection performance. Generic approaches also avoid the instability when the number of concepts is large.

A generic semantic system is built by investigating various visual features to exploit the content of videos, and further to learn the information at semantic level by applying effective learning algorithm. Each feature has its own function. In this case, it is not an 'either-or', but 'and-and' approach to a robust semantic detection system, where fusion techniques effectively combine those features and classifiers and affect the performance of systems. How to effectively fuse different features and classification scores becomes crucial.

Generally, there are two typical fusion methods, namely early fusion and late fusion. As the name implies, early fusion is carried out at feature level, that is to say, various features are concatenated into one supervector for classification; while the late fusion methods fuse the modalities in classification scores level by using supervised/semi-supervised learners directly.

In [6, 7], the approaches to early fusion fed the concatenated supervector into a single-kernel SVM classifier to learn the semantic model. The weak points of this method, such as a sharp increase in feature dimension, the inflexibility of further analysis of different feature effects, seriously limit the widespread use. *Multiple Kernel Learning* (MKL) method, which can get accurate classification results and identify relevant and meaningful features, was introduced to overcome the defects of traditional single-classifier learning for high-dimensional concatenated feature at early fusion stage. One-layer MKL has achieved excellent performances in [8–10]. Considering four types of features will be used in the system, we propose a two-layer MKL scheme to make an exhaustive use of the descriptions of intra/inter feature groups and prevent the combination from the sparsity of combining weights.

By contrast, late fusion is popularly used for information fusion due to easily computed and flexibly analyzed. Simple late fusion methods achieved satisfying performances, by obtaining average maximum, minimum, sum, product, and geometric mean [7, 11]. EuroCom [12] proposed a hierarchical fusion structure on these generic algorithms. Borda counting was also used for late fusion by predicting the combining weights of each classification results on the test set [7, 13, 15]. What is more, supervised learners are used to train and learn fusion model, such as Bayesian posterior probability model [14], probability model support rank aggregation (PMSRA) [15], and ordered weighted average (OWA) [16, 17]. RankBoost learning algorithm was carried out to optimize the fusion model by reducing the error rate; RelayBoost was proposed to solve the issue of imbalanced positive/negative samples [18]. In [19, 9], logistic regression was used to learn the fusion model and showed the discrimination and generalization ability over large numbers of concepts in a generic semantic analysis system. Consequently, our proposed late fusion applies two-layer fusion methods based on logistic regression.

In this paper, two kinds of fusion schemes are presented for a generic semantic indexing system, and the system implementation is given out by using a variety of low-level visual features and making exhaustive use of machine learning methods. Various visual descriptions are deployed, like histograms in the RGB color space, local binary pattern (LBP), and local information by interest points descriptor, in order to meet the different characteristics of kinds of concepts and develop a generic approach. Then, kernel-based learning methods are utilized to learn the rich low-level feature information and combine them to enable the machine a better understanding capability, where two kinds of notable kernels are tried, i.e., χ^2 kernel and Histogram Intersection kernel. Finally, five two-layer structures for early fusion and late fusion are outlined, where multiple kernel learning and logistic regression are applied, respectively. The experiments on TRECVID2009 [9] and TRECVID2010 [10] show that the notable fusion schemes improve our semantic indexing engine by developing a generic model, and excellent performances are gained on both object and scene type concepts.

The rest of the paper is arranged as follows. Section 2 presents our proposed fusion schemes for a generic semantic indexing system and the details of the methodology. The performances are evaluated and discussed in Sect. 3. Finally, in Sect. 4, some conclusions are drawn and future work is given.

2 Methodology

2.1 Overview

A video clip is usually perceived as a document, which can be decomposed into paragraphs, sentences, and words. Similarly, for faster and more accurate access, video is

segmented into scenes and shots, and key sequences or key frames are extracted as analysis and indexing units. A typical structure imposed on the videos for efficient browsing is shown in Fig. 1. Processing for segmentation can be done by shot boundary detection technology [20]. Key-frames can be extracted from shots for further analysis. In our scheme, the visual features are extracted from the key-frames image, and so is the evaluation of performance, so the analysis of semantic concepts can be given on the level of shots.

Semantic indexing is treated as a combined computer vision and machine learning problem. Various visual features are extracted to represent the content of key frames, and then fed into the supervised classifiers to learn the relation between visual features and semantics. After that, a fusion step integrates the representation capability of various features and different modeling information from different learning machines, which can be carried out either on the feature level or on the classifier level. The generic structure is illustrated in Fig. 2.

In the following parts of this section, the details of implementation will be dived into, including kinds of currently useful visual feature extraction methods in Sect. 2.2, kernel-based machine learning methods and SVM in Sect. 2.3, and the two-layer fusion methods which combine the multiple feature information to get better understanding of semantic concepts in Sect. 2.4.

2.2 Features extraction

Feature extraction techniques have a crucial impact on the performance of pattern recognition systems as an important part of such system. Many variations exist in a specific semantic concept, for example, variations of the viewpoint, lighting, and rotations. Hence, visual features are expected to be minimally affected by the variations, while they are still able to distinguish concepts with different semantics.

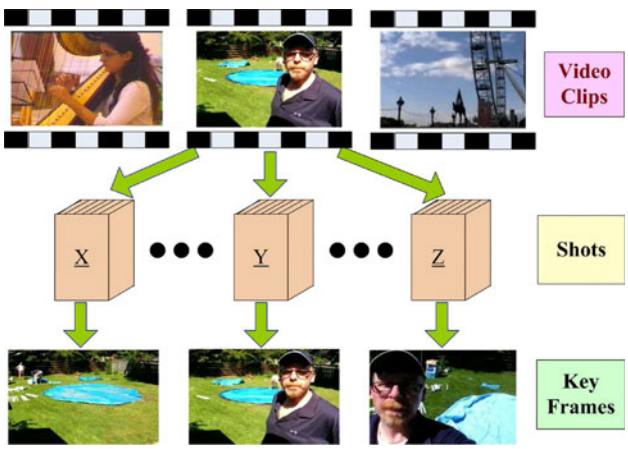

Fig. 1 A typical structure imposed on the videos for efficient browsing

Recent work has shown that local features are powerful representations, like scale-invariant feature transform (SIFT) [22]. SIFT descriptors in a manner of Bag-of-Words model [25] and pyramid multi-resolution represents. They are, to some extent, invariant to intensity, color, scale, and rotation.

Compared with the local features which mainly focus on interest points or interest regions, the traditional features extract the visual information from the whole or large blocks of images, so-called global features. They were used widely to represent the information of entire image in the field of scene classification and recognition [21]. There are three kinds of global features involved in our system, which are color, texture, and edge features. Most commonly, they are computed easily and efficiently, and most importantly be invariant to slight translation, viewing angle, and robust to low noise, to some extent.

A summary of the low-level visual features used in the paper are presented in Fig. 3.

2.2.1 *Global features*

1. *Color descriptors* For some semantic concepts, the color is specific. Color features are the most intuitive ones, for example, they are able to classify the green grass and blue sky easily. The color statistics can be gathered per pixel, either in standard RGB space or HSV space.
   There are three types of color feature descriptors in our systems, including Grid color moments (GCM), Color auto-correlograms (CAC) [24], and Color coherence vector (CCV) [23]. These features offer the information on spatial configurations of the pixels, which is also distinctive in many situations, and also are invariant to viewing angle, scale, and local image transformations.
2. *Texture descriptors* Texture is of importance in classifying different materials like the line-like pattern in a brick wall, or the dot-like pattern of sand. It is robust to shadow and light changes, and rotation.
   Two typical texture features are adopted, which are Local Binary Patterns (LBP) and Gabor filters. Gabor filters are designed in four scales and six orientations, and use mean and variance statistics of each filter as the feature. Local binary patterns (LBP) are adapted to find the patterns in images. The statistics of textures are calculated in regions.
3. *Edge descriptors* Edges are kind of measures like texture considering the local patterns. Edge features detect the changes of edges of the regions, while texture features focus on the characteristics of the region.

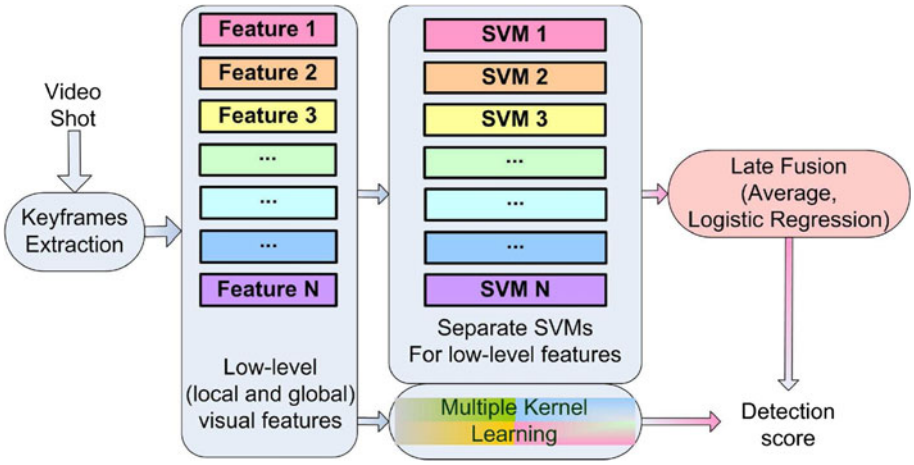

Fig. 2 Framework of a generic semantic analysis structure

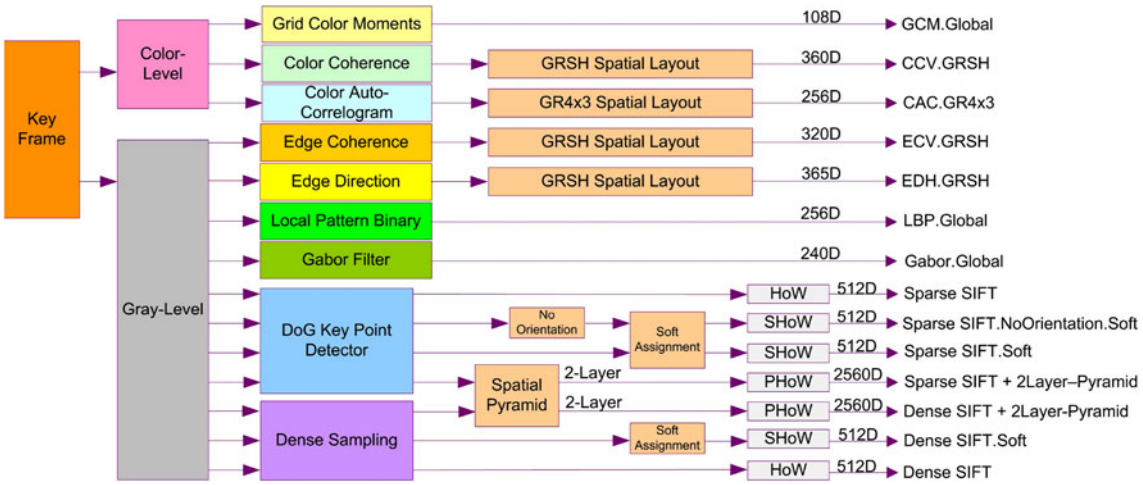

Fig. 3 Summary of low-level visual features

This paper uses two types of edge features, edge direction histogram (EDH) and edge coherence vector (ECV). Edge coherence vector (ECV) also considers the spatial information of the edges of image, as does CCV for a color image.

Spatial layout partition is applied to incorporate spatial information, and the main partition methods include GRI5, GRSH, GR4x3 [26]. Figure 4 gives an intuition of these spatial layouts.

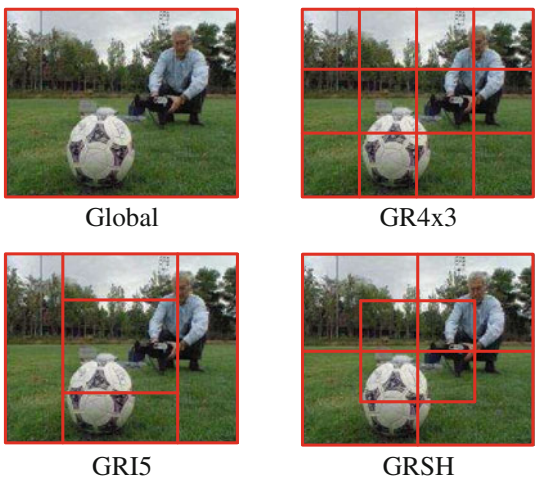

Fig. 4 Spatial layouts of global features [26]

2.2.2 *Local features*

For each key frame, SIFT descriptors are extracted at some difference of Gaussian (DoG) interest points, so-called sparse SIFT [22], as well as densely sampled at points on a grid with spacing of 6 pixels, so-called dense SIFT [27].

The SIFT descriptors are Gaussian derivatives computed at 8 orientation planes over a 4×4 grid of spatial locations, giving a 128-dimensional vector. A codebook of 512 words is generated by K-means for sparse and dense SIFT descriptors, respectively. One descriptor is either assigned to one visual word, or softly assigned to three words, so-called soft assignment (SA) [10]. Either in training or

testing phase, the descriptors are accumulated to a histogram representation, so-called histogram of words (HoW) [25].

Similar to Sect. 2.2.1, the image is partitioned into sub-blocks (as shown in Fig. 5) and histograms of SIFT visual words are computed in these sub-blocks. Histograms of visual words for each sub-region were then concatenated into one super vector, so-called pyramid histogram of words (PHOW). Similar to the method used in [28], two-layer PHOW representations are built for both sparse and dense SIFT, respectively.

2.3 Kernel-based classification

Kernel-based learning algorithms are known to represent complex decision boundaries very efficiently and generalize well to unseen data [29]. The use of kernels overcomes the problem of non-linearly separable data sets by mapping the initial problem into a higher dimensional space.

Support vector machine (SVM), as one of the popular kernel-based learning method, has shown its capacity for supervised learning of high-level semantic concepts over low-level visual features.

Given l N-dimensional training vectors (x_i, y_i), the feature vector $x_i \in \mathbf{R}^N$, the corresponding class label $y_i \in \{-1, 1\}$ in two classes, $i = 1, \cdots, l$. SVM solves the following primal problem:

$$\min_{w,b,\xi} \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{l}\xi_i \tag{1}$$

$$\text{s.t.} \quad y_i\left(\omega^T\phi(x_i) + b \geq 1 - \xi_i\right), \tag{2}$$

$$\xi_i \geq 0, i = 1, \ldots, l \tag{3}$$

In a two-class case, the decision function for a test sample x has the following form:

$$y(x) = \sum_{i=1}^{N} y_i\alpha_i k(x_i, x) + b \tag{4}$$

where α_i the learned weight of the training sample x_i, b is a learned threshold parameter, and $k(\cdot,\cdot)$ is the kernel function.

Fig. 5 Spatial layouts of PHOW

Kernel function maps the initial problem into a higher dimensional space and computes the similarities of the sample vectors. Conventional kernels, such as radial basis function (RBF) kernel, have shown great capacity of discrimination and generalization. Recent researchers in multimedia retrieval show that Histogram Intersection kernel (henceforth, *HI*) and heavy-tailed χ^2 kernel (so-called ChiSquare, *CS*) can outperform RBF kernels on difficult image classification problems where the features are high-dimensional histograms [32].

In our work, different suitable kernel types are set for different features to achieve better performance, and also leave the space for the potential of multiple kernel learning, which is introduced in the following sections.

For all the features as described in the previous section, the χ^2 kernels are adopted to model the similarity between histograms. The exponential χ^2 kernel for two image x and y is defined to be

$$k_{\text{CS}}(x, y) = \exp\left(\chi^2(H_x, H_y)/A\right) \tag{5}$$

where H_x is the histogram of image x; the χ^2 distance is evaluated as

$$\chi^2(H_x, H_y) = \frac{1}{2}\sum_{i=1}^{N}\frac{\left(H_x(i) - H_y(i)\right)^2}{H_x(i) + H_y(i)} \tag{6}$$

and, the scaling parameter A, a parameter of χ^2 kernel, can be tuned to get good performance over development data.

In particular, *HI* kernel is also chosen to map the high-dimensional features in manner of histogram into a high *Hilbert* space. This kind of kernel is easy to compute and also time-saving. The exponential *HI* is formulated as

$$k_{HI}(x, y) = \exp\left(-HI(H_x, H_y)\right) \tag{7}$$

where the *HI* is defined as

$$HI(H_x, H_y) = \sum_{i=1}^{N}\min(H_x(i), H_y(i)) \tag{8}$$

and, *HI* kernel does not have kernel parameters while χ^2 kernel does. Hence, *HI* kernel saves the time of searching for a optimal value for a kernel parameter.

The features of spatial pyramid representation also use the two kernel types in our experiments. Take the pyramid χ^2 exponential kernel as an example, the formulation is defined as [32]:

$$k(x, y) = \sum_{l=1}^{1}\sum_{i=0}^{2^{2l}-1}\exp\left(\chi^2(H_x^{l,i}, H_y^{l,i})/A^{l,i}\right) \tag{9}$$

where $H_x^{l,i}$ stands for the histogram of the i-th sub-region at the l-th level. In our work, all sub-regions are weighted equally.

The scheme of SVM's training and testing process is illustrated in Fig. 6, and the toolkit SVMTorch [30] is used to learn the support vectors and corresponding weights. Grid search is used to select the parameters of the kernels. Similar to [31], the cost parameter C is set by the ratio of negative and positive samples' numbers, in order to handle imbalance problem in the number of positive and negative samples for each high-level concept. In the training step, the positive samples are from one topic class and the negative data consist of the samples of other topic classes.

2.4 Fusion strategy

To combine the representative capabilities of the low-level features, a combined analysis or fusion is carried out to gain insight into the role of various analysis approaches on concept detection performance. Two fusion strategies are tried for the fourteen features in our system. One is carried out on kernel-level combination, so-called early fusion. The other is based on classifier-level combination, so-called late fusion.

2.4.1 Early fusion: kernel-level combination

Different from conventional early fusion methods which simply concatenate low-level features into one super vector, our proposed early fusion strategy on kernel level makes use of the technique called multiple kernel learning (MKL) [33].

In this strategy, a composite kernel is constructed by weighted linear combination of multiple kernels for various low-level features:

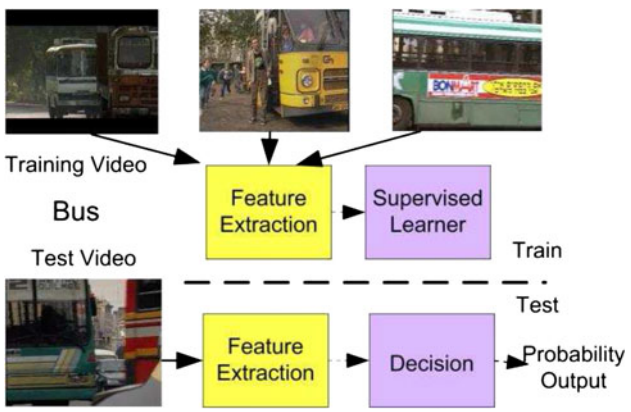

Fig. 6 SVM's training and testing. Take the concept "Bus", for example, "Bus" model is learned with the training samples as shown in the upper block; the lower block shows an unknown sample is tested with a probability output by being compared with the "Bus" model after being fed into the same feature extraction module in "train" part

$$K(x,y) = \sum_{\mathrm{f}} \beta_{\mathrm{f}} K_{\mathrm{f}}\left(L_x^f, L_y^{\mathrm{f}}\right) \tag{10}$$

where L_x^{f} is f feature vector extracted from the image x, K_{f} is corresponding kernel, and β_f is the kernel weight. It is expected that such composite kernel could measure input similarities from various aspects (e.g., local gradient information, color, edge, texture, etc.) by integrating similarity measures with respect to various low-level features. An SVM is then trained with this composite kernel to detect high-level semantic concept.

The kernel combination weights could be learned through MKL, and also be set before learning for each high-level concept. The MKL approach is adopted as proposed in [33]. To prevent MKL from deriving two sparse solutions on the kernel weights, a regularization term $\mathrm{R} = \varphi \Sigma_{\mathrm{f}} \beta_{\mathrm{f}}^2$ is added to the MKL objective function [34]:

$$\min_{\beta_{\mathrm{f}},\omega_{\mathrm{f}},b,\xi_i} \frac{1}{2}\sum_{f=1}^{F}\frac{1}{\beta_{\mathrm{f}}}\|\omega_{\mathrm{f}}\|_2^2 + C\sum_{i=1}^{N}\xi_i + \varphi\sum_{f=1}^{F}\beta_{\mathrm{f}}^2 \tag{11}$$

$$\text{s.t. } y_i\left(\sum_{f=1}^{F}\langle\beta_{\mathrm{f}}, L_i^{\mathrm{f}}\rangle + b\right) \geq 1-\xi_i, \quad \forall i \tag{12}$$

$$\xi_i \geq 0 \tag{13}$$

$$\sum_{f=1}^{F}\beta_{\mathrm{f}} = 1, \beta_{\mathrm{f}} \geq 0, \quad \forall f \tag{14}$$

where b, ξ_i, and C are the standard SVM bias, slack variables, and regularization term; ω_f is the primal SVM weight associated kernel K_{f} (with underlying feature transformation function ϕ_f, i.e.):

$$K_{\mathrm{f}}(L_x^{\mathrm{f}}, L_y^{\mathrm{f}}) = \left\langle \phi_{\mathrm{f}}(L_x^{\mathrm{f}}), \phi_{\mathrm{f}}(L_y^{\mathrm{f}}) \right\rangle \tag{15}$$

$$\omega_{\mathrm{f}} = \beta f \sum_{i=1}^{N} \alpha_i y_i \phi_{\mathrm{f}}(L_i^{\mathrm{f}}) \tag{16}$$

The parameter, φ, in Eq. 11, controls the level of sparsity in MKL solution. Larger values of φ would drive MKL towards a more uniform set of weights. In our experiments, φ was set to 10.

2.4.2 Schemes of early fusion

Although different features extract different visual information or differ in organizing the information, those features in the same group share some same or similar visual information intuitively and also provide their patent of information to the group. For color feature group *C3*, it contains three color features, i.e., GCM, CCV, and CAC. They describe color characters of the image by considering

the moments, correlograms, and coherence information, respectively. So, combination of them will enrich the color information.

A second combining is applied to the first combinations. This process reaches some sort of equilibrium that each group plays a strong role with adequate information in describing the contents of images, and intuitively, the combining weights of feature groups are not sparse.

So, two early fusion schemes are built for kernel-level combination with two-layer structure of feature combination, shown as block (A) and (B) of Fig. 7. This feature grouping methods have been proven effective and achieved the third best performance in the task high-level feature extraction (HFE) of TRECVID2009 [35] and Semantic INdexing (SIN) of TRECVID2010 [36]. One-layer fusion structure is also applied for further comparison and illustrated in block (C).

Each scheme considers two ways of achieving combination weights in the multiple kernel. One way is that the weights are learned by SVMs, at the same time when they learn the support vectors, named as *Multiple Kernel Learning*; another is that the weights are predefined equally, named as *Equal Composite Learning* in Fig. 7, where SVMs only learn the support vectors.

Block (A) and (B) apply two different feature-grouping metrics for the first layer combination of multiple kernel on basis of the features introduction in Sect. 2.2. Block (A) shows that the features are mainly grouped into two collections, one contains global features and another is local. While Block (B) partitions the global feature group

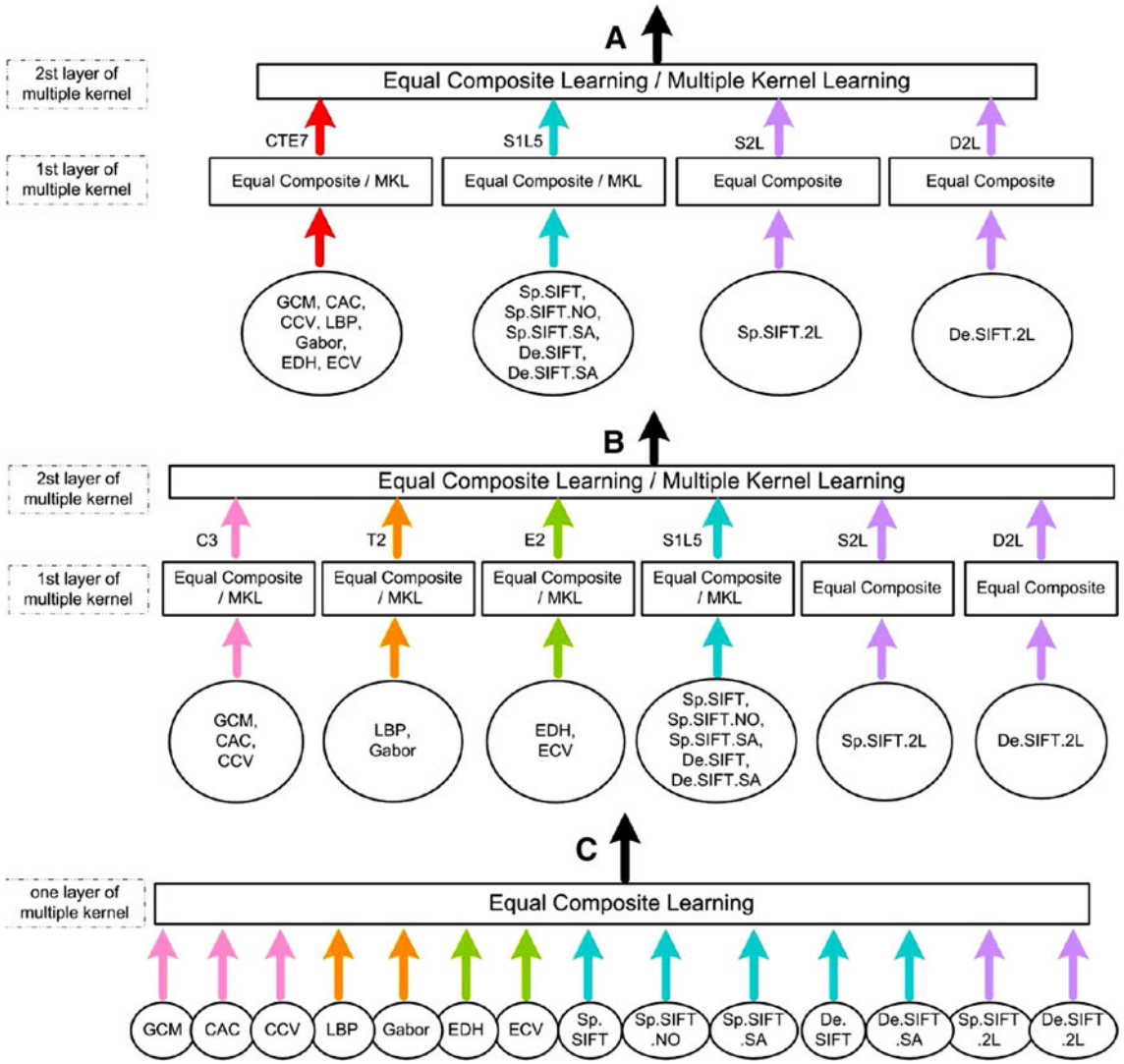

Fig. 7 Schemes of early fusion (EF). The output multiple kernels on second layer in **a** and **b** is composed of 4/6 multiple kernels on the first layer, and the kernels on first layer are combined with the intra-group features, respectively. In contrast to the two-layer approaches of **a** and **b**, a basic structure of one-layer multiple kernel learning is presented, designed as the block diagram **c**

into 3 smaller ones according to traditionally defined visual property.

The multiple kernels combined by features from the first layer are treated as the inputs of multiple kernel on the second layer. The output multiple kernel generated by Block (A) combines 4 multiple kernels on the first layer, i.e., *CTE7*, *S1L5*, *S2L*, and *D2L*, while kernel of Block (B) is composed of six multiple kernels, labeled with the tags *C3*, *T2*, *E2*, *S1L5*, *S2L*, and *D2L*, respectively. For clarity, *C3* stands for the multiple kernel used for the concatenated super vector of 3 color features, and its formulation is defined as follows. Similarly, *T2* represents the one for 2 texture features; *E2* represents the one for 2 edge features; *S1L5* represents the one for 5 features related to one-layer SIFT; and *CTE7* represents the one for the combination of 7 color, texture and edge features.

For example, the formulation of output kernel of Block (A) is represented by

$$K_A(x,y) = \sum_{f=CTE7,S1L5,S2L,D2L} \beta_f K_f\left(L_x^f, L_y^f\right) \tag{17}$$

where the formulations of *CTE7*, *S1L5*, *S2L*, and *D2L* are the same as Eq. (17), like

$$K_{CTE7}(x,y) = \sum_{\mathrm{f}} \beta_{\mathrm{f}} K_{\mathrm{f}}\left(L_x^{\mathrm{f}}, L_y^{\mathrm{f}}\right) \tag{18}$$

and L_x^{f} is f feature vector extracted from the image x, K_{f} is corresponding kernel and β_f is the kernel weight. In Eq. (18), $f \in \{\mathrm{GCM}, \mathrm{CAC}, \mathrm{CCV}, \mathrm{LBP}, \mathrm{Gabor}, \mathrm{EDH}, ECV\}$.

To be mentioned, the kernels (S2L and D2L) used for the features of 2-layer pyramid histogram, i.e., Sparse SIFT + 2LayerPyramid (represented by Sp.SIFT.2L in Fig. 7) and Dense SIFT + 2LayerPyramid (De.SIFT.2L), are nature multiple kernels, which combine one kernel for the whole image and 4 kernels for subregions of Fig. 5. Take S2L for an example, the formulation is described as

$$K_{S2L}(x,y) = \sum_{f} \beta_f K_f\left(L_x^f, L_y^f\right) \tag{19}$$

where $f \in \{whole, UL, UR, DL, DR\}$. Considering the growing complexity of kernel matrix and the computing time, the two-layer local features De.SIFT.2L and Sp.SIFT.2L are separated from those one-layer ones as one single group.

2.4.3 Late fusion: classifier-level combination

Corresponding to early fusion, there exists a fusion strategy called late fusion, wherein a SVM corresponding to a single low-level visual feature and semantic concept will classify the semantic concept and give a ranked list according to the decision value. The ranked lists from the various features are aggregated to get an optimum combined ranked list. This method was also recognized as classifier-level combination.

Late fusion needs two learning stages. At the first stage, various supervised learners learn unimodal features to separately learn concept scores, SVM learners were used on this stage. And the second learning stage is carried out on semantic level. Late fusion focuses on the individual strength of modalities. Unimodal concept detection scores are fused into a multimodal semantic representation.

For our baseline results, the average fusion strategy is simply utilized, which usually generates robust performance from the empirical study of many previous works. In the simplest condition, the weight for each classifier is equal, which is called average fusion.

To reach a stable and robust performance, logistic regression (LR) [37] is also used to learn the combination weights. Similar to the classification problem introduced in Sect. 2.3, logistic regression applies linear kernel, and the following primal problem of logistic regression is

$$\min_{\omega} \frac{1}{2}\omega^T\omega + C\sum \log\left(1 + \exp(-y_i\omega^T x_i)\right) \tag{20}$$

where ω is a matrix/vector with the model weights. $\omega^T x$, where x ranges from 1 to the number of dimensions.

In the following section, the toolkit Liblinear [37] is used to implement logistic regression and to obtain the weights of linear combination, and the training and testing flowchart of logistic regression is similar to the one of SVM illustrated in Fig. 6.

2.4.4 Schemes of late fusion

Yuan [9, 10] has proven that it was not robust to carry out one-layer fusion structure where the combining weights were set over all low-level feature SVMs. These features are grouped into several collections, and a two-layer fusion structure is built to combine the results of different features and groups (as shown in Fig. 8). Similar to Sect. 2.4.2, the combining weights of the two layers are set equally, and are also optimized through logistic regression.

In Fig. 8, the block diagrams (A) and (B) are similar, except for the granularity of the category setting on the first layer of fusion, where two collections are grouped in terms of whether the feature belongs to SIFT group, and five are grouped according to the perceivable attributes for (A).

For clarity, *C3* represents the combination of 3 SVM scores related to the color features; similarly, *T2* represents the combination of 2 texture features' SVM scores; *E2* represents the combination of 2 edge features' SVM scores; *S1L5* represents the combination of 5 SVM scores related to the one-layer SIFT features; *S2L2* represents the

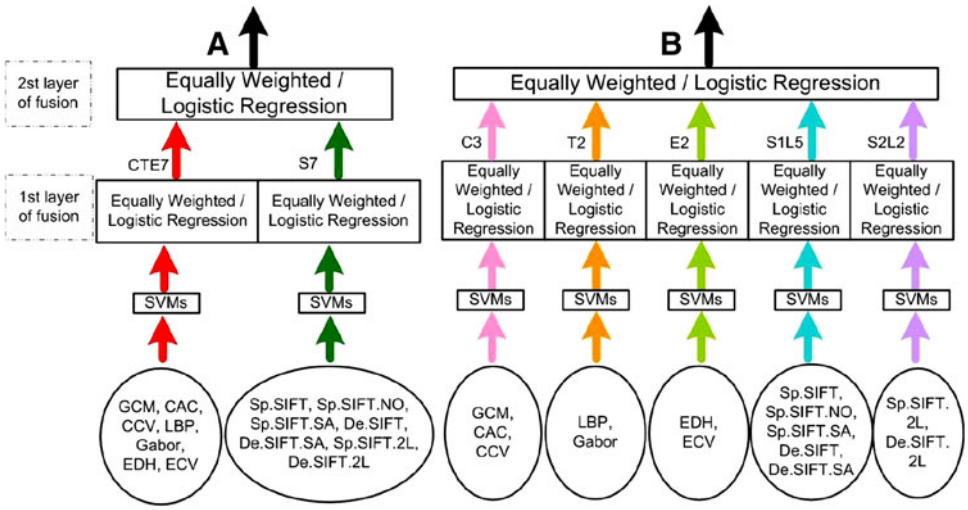

Fig. 8 Schemes of late fusion (LF)

combination of 2 SVM scores related to the two-layer SIFTs; *CTE7* represents the combination of 7 color, texture and edge features's SVM scores; and *S7* represents the combination of 7 local features' SVM scores.

3 Experiments and discussions

3.1 Data sets

The multilayer fusion schemes are evaluated on the archives of TRECVID09 and TRECVID10 benchmarks. TRECVID has been of great importance in assessing complete multimedia indexing methods. The video archive of the TRECVID 2010 is from internet videos of IACC in MPEG-4/H.264, while TRECVID 2009 videos are archived from various channels covering science news, documentaries, and news reports. The general quality of TREC09 videos are better than that of TREC10, which means the higher performance can be achieved in TRECVID 2009 benchmark. As listed in Table 1, lexicon is compiled for experiment with the semantic concepts including some typical concepts of scenes, objects, programs, and events which appear, respectively, in TREC09 and TREC10 concept lexicons. The published data which contain these concepts are split up into two non-overlapping training (making up 60% shots) and evaluation set (the remaining 40% shots) respectively for our experiments. This process is repeated for 5 times, and the average accuracy of these scores for each system is given finally.

3.2 Evaluation criteria

We use average precision (AP) to determine the accuracy of ranked concept detection results on our experiments, following the standard in TRECVID evaluations. Average precision is a single valued measure that is proportional to the area under a recall-precision curve. This value is the average of the precision over all relevant judged shots. Hence, it combines precision and recall into one performance value. Let $L^k = \{l_1, l_2, \cdots, l_k\}$ be a ranked version of the answer set A. At any given rank k, let $R \bigcap L^k$ be the number of relevant shots in the top k of L,where R is the total number of relevant shots. Then, average precision is defined as

$$\mathrm{AP}(L) = \frac{1}{R}\sum_{k=1}^{A}\frac{R\bigcap L^k}{k}\psi(l_k) \tag{21}$$

where indicator function $\psi(l_k) = 1$ if $l_k \in R$ and 0 otherwise. As the denominator k and the value of $\psi(l_k)$ are dominant in determining average precision, it can be understood that this metric favors highly ranked relevant shots. Mean average precision (MAP) is defined as the mean of the concepts' AP scores, in order to compare multiple concept detection results.

3.3 Results

3.3.1 Evaluation on early fusion

Three types of fusion schemes stated in Sect. 2.4.2 are implemented, where two multiple kernel learning methods, i.e., *MKL* and *ECL*, and two kinds of kernel configurations are tested. The details of performances and configurations for different early fusion schemes are listed in Table 2, and the performance comparison between three early fusion schemes is visualized in Fig. 9.

Generally speaking, from Fig. 9 we can see that the schemes *EarlyFusion.A* and *EarlyFusion.B* based on two-layer structures perform better and more robustly than the single layer *EarlyFusion.C* to obtain the generic semantic

Table 1 Concepts of TRECVID 09 and 10 in our Lexicon

Concepts of TREC09 HFE Task				
Classroom	Person_riding_bicycle	Chair	Telephone	Infant
Person_eating	Traffic_intersection	Demonstration_or_protest	Doorway	Hand
Airplane_flying	People_dancing	Person_playing_instrument	Female_close up	Bus
Boat_ship	Person_playing_soccer	Nighttime	Cityscape	Singing
Concepts Of TREC10 SIN Task				
Airplane_flying	Anchorperson	Animal	Beach	Bicycles
Boat_ship	Bus	Cats	Chair	Charts
Cityscape	Classroom	Construction_vehicles	Crowd	Dark-skinned_people
Demonstration_or_protest	Female_person	Flowers	Hand	House_of_workshop
Instrumental_musician	Laboratory	Nighttime	Roadway_junction	Running
Shopping_mall	Sing	Sitting_Down	Sports	Telephone

Table 2 MAPs and corresponding kernel-level configurations of different early fusion schemes

RunID	Scheme	Combination	Kernels	TREC09 MAP/SD	TREC10 MAP/SD
EF.A.E.CS	EF.A	Equal CompositeL	all: χ^2 Kernels	0.17145/0.00451	0.07119/0.00071
EF.A.E.CH	EF.A	Equal CompositeL	CTE7: χ^2 Kernels, SIFTs: *HI* Kernels	0.16494/0.00246	0.06813/0.00025
EF.A.M.CS	EF.A	MKL	all: χ^2 Kernels	0.07348/0.00067	0.17546/0.00012
EF.A.M.CH	EF.A	MKL	CTE7: χ^2 Kernels, SIFTs: *HI* Kernels	0.16941/0.00109	0.07018/0.00044
EF.B.E.CS	EF.B	Equal CompositeL	all: χ^2 Kernels	0.16585/0.00128	0.07014/0.00027
EF.B.E.CH	EF.B	Equal Composite	CTE7: χ^2 Kernels, SIFTs: *HI* Kernels	0.15943/0.00274	0.06754/0.00038
EF.B.M.CS	EF.B	MKL	all: χ^2 Kernels	0.16528/0.00249	0.07092/0.00007
EF.B.M.CH	EF.B	MKL	CTE7: χ^2 Kernels, SIFTs: *HI* Kernels	0.15887/0.00142	0.06762/0.00025
EF.C.E.CS	EF.C	Equal Composite	all: χ^2 Kernels	0.16582/0.00420	0.07107/0.00023
EF.C.E.CH	EF.C	Equal Composite	CTE7: χ^2 Kernels, SIFTs: *HI* Kernels	0.14138/0.00842	0.06616/0.00081
EF.C.M.CS	EF.C	MKL	all: χ^2 Kernels	0.14598/0.00454	0.07011/0.00015
EF.C.M.CH	EF.C	MKL	CTE7: χ^2 Kernels, SIFTs: *HI* Kernels	0.13826/0.00244	0.06649/0.00009

indexing system where a shared set of low-level features are used to describe the diverse semantic concepts. Two-layer structure enhances the descriptive capability of these features of similar attributes and make all categories of features play a strong role. At the same time, it prevents multiple kernel learning from deriving sparse solutions on the kernel weights especially in case of large scatter within the classes of some semantic concepts, although it is much more complicated to compute and to solve the quadratic programming (QP) problem for classification than the one-layer.

Comparing performances of different two-layer fusion schemes, the systems with structure *EarlyFusion.A* achieve better results than the ones with *EarlyFusion.B*. They differ in the predetermined quantity of groups in the first layer of multiple kernel. The results demonstrate that the more low-level features which have similar attribute and descriptive ability grouped together, the better results they will achieve. That is because although the color, texture, and edge features have the poor description and discrimination, the combination of them can make a big contribution to semantic analysis, which has been proved in the experiments of late fusion.

MKL-based approach achieves more robust system with stable performances than the equally weighted composite kernels when the kernels in the first level of multiple kernel change. *MKL* shows the effects in controlling sparsity of the combining weights for each sub-kernels, makes an exhaustive use of all the involved features, and consequently makes semantic models more general on the evaluation data set. But, many issues, such as the limited number of positive samples for some concept and large intra-concept variability, affect the generalization capability of multiple kernel learning.

The first-level kernels directly affect how well the features exert their discriminative capability, and consequently play an essential role in building a powerful multiple kernel. Comparing to *HI* kernel, χ^2 kernel obtains much higher mean average precision. We can see that χ^2 kernel is effective and could derive more robust semantic models from these high-dimension features in manner of histogram, without regard to the computation cost.

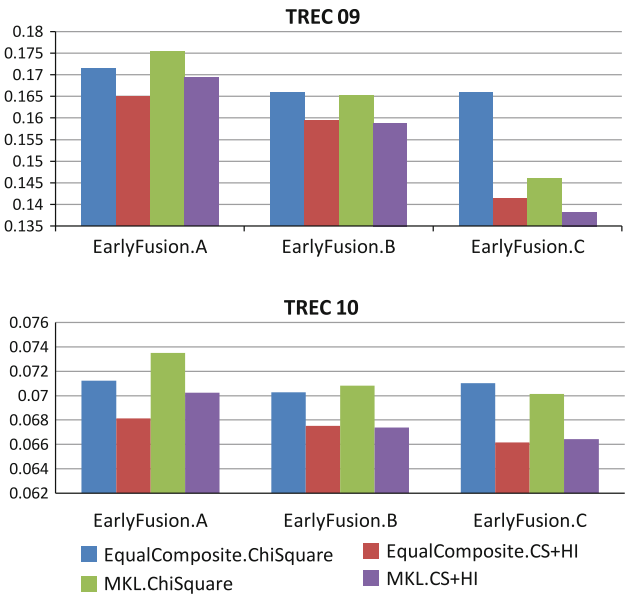

Fig. 9 Performance comparison of different early fusion schemes. Each *bar* means the MAP Value of one early fusion scheme, e.g., the *blue bars* named *EqualComposite.ChiSquare* in *EarlyFusion.A* parts refer to RunID *EF.A.E.CS* in Table 2

3.3.2 Evaluation on late fusion

Before the discussion about the performances of late fusion methods, the unimodal features' indexing performances are analyzed, and the MAP scores and the SD (Standard deviation) are listed in Table 3. Fig 10 gives a more intuitive comparison among all the low-level visual features.

Table 3 demonstrates that the features related to SIFT give a better performance in a large scale than all the other visual features, which prove that local features indeed represent more information of video content. Of all the SIFT features, dense SIFT descriptors perform better than the sparse ones. As TRECVID data are low-resolution video to some extent, the sparse SIFT detects fewer interest points and gets less information than the dense ones. The two-level spatial pyramid representations of SIFT descriptors effectively understand the spatial context and make an impression with the highest MAP scores.

Comparing the results of single-feature systems in Table 3 and classification-score-level combination system in Table 4, any of the former single systems gives poor performance, while with system fusion strategies stated in Fig. 8, combination of them can make big contribution to a better semantic indexing system.

Besides, the *CTE7* can be perceived as the combination of *C3*, *T2*, and *E2*. The results indicate that *CTE7* outperforms any of the sub-combinations *C3*, *T2*, and *E2*, and *S7* also performs better than both *S1L5* and *S2L2*.

In addition, for color, edge or texture features, their combination, i.e., the *CTE7* system, achieves comparable performance with SIFT-based systems, i.e., *S1L5*, *S2L2*, and *S7*, as stated in Table 4, although the performance of any of these systems with traditional features is inferior to those based on SIFT local descriptors. Combination of the features who are of similarly poor description is able to perform as well as the powerful descriptors do.

The results visualized in Fig. 11 show that the systems adopting late fusion scheme (A) outperform the ones using scheme (B). Performance of scheme *CTE7* is comparable to the one of scheme *S7*, while schemes *S1L5* and *S2L2* outperform *C3*, *T2*, and *E2*, by a substantial degree. Combining the classification scores with the same magnitude can prevent the combination from the sparsity of combining weights and implement semantic analysis robustly by making an exhaustive use of all the involved

Table 3 Performances and configuration of 14 low-level visual features' SVMs

Feature	Kernel	TREC09 MAP	TREC09 SD	TREC10 MAP	TREC10 SD
GCM	χ^2	0.05204	0.00018	0.04214	0.00010
CAC	χ^2	0.04202	0.00084	0.03753	0.00016
CCV	χ^2	0.05065	0.00165	0.04529	0.00021
LBP	χ^2	0.06201	0.00046	0.03927	0.00004
Gabor	χ^2	0.04326	0.00119	0.03619	0.00054
EDH	χ^2	0.05909	0.00094	0.04235	0.00007
ECV	χ^2	0.07283	0.00082	0.02932	0.00062
SpSIFT	χ^2	0.06107	0.00013	0.04346	0.00042
SpSIFT.NoOri	χ^2	0.06190	0.00020	0.05539	0.00006
SpSIFT.Soft	χ^2	0.06709	0.00017	0.05114	0.00037
SpSIFT.2L.PHOW	Pyramid χ^2	0.06840	0.00033	0.04876	0.00022
DenSIFT	χ^2	0.09971	0.00073	0.05453	0.00026
DenSIFT.Soft	χ^2	0.10039	0.00008	0.05525	0.00014
DenSIFT.2L.PHOW	Pyramid χ^2	0.12250	0.00020	0.06143	0.00010

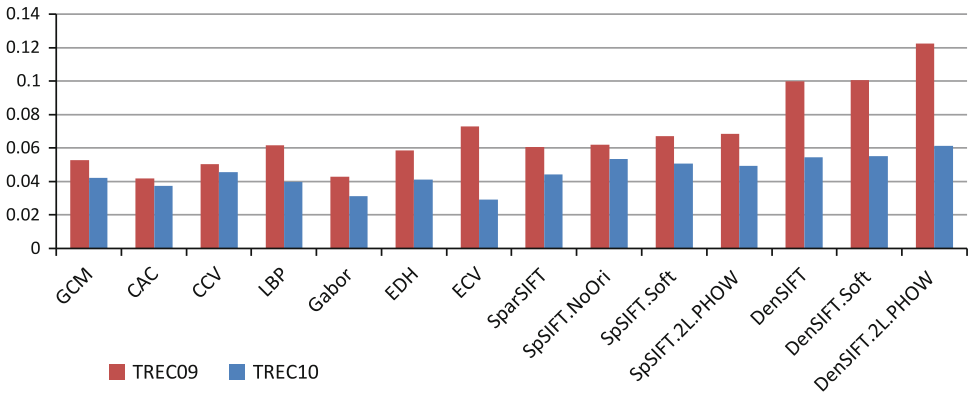

Fig. 10 Performance comparison of 14 low-level visual features. MAP values in TREC 09 and 10 for various features is listed. Sp- and Den- are the abbreviations for sparse and dense, PHOW is short for pyramid histogram

Table 4 Performances and configurations of classifier-level combination

RunID	Scheme	Combination	TREC09 MAP	TREC09 SD	TREC10 MAP	TREC10 SD
CTE7.E	LF.A	EqualWeighted	0.12262	0.00027	0.06329	0.00027
C3.E	LF.B	EqualWeighted	0.05965	0.00048	0.04870	0.00051
T2.E	LF.B	EqualWeighted	0.06840	0.00062	0.04260	0.00012
E2.E	LF.B	EqualWeighted	0.07252	0.00067	0.04341	0.00004
S7.E	LF.A	EqualWeighted	0.14598	0.00012	0.06530	0.00016
S1L5.E	LF.B	EqualWeighted	0.14932	0.00072	0.06152	0.00026
S2L2.E	LF.B	EqualWeighted	0.13662	0.00071	0.06379	0.00002
LF.A.E	LF.A	EqualWeighted	0.15612	0.00025	0.06888	0.00013
LF.B.E	LF.B	EqualWeighted	0.15966	0.00011	0.07033	0.00003
CTE7.L	LF.A	Logistic-regress	0.15452	0.00019	0.06402	0.00004
C3.L	LF.B	Logistic-regress	0.12393	0.00055	0.04932	0.00021
T2.L	LF.B	Logistic-regress	0.13323	0.00034	0.04189	0.00005
E2.L	LF.B	Logistic-regress	0.12177	0.00041	0.04331	0.00004
S7.L	LF.A	Logistic-regress	0.14875	0.00037	0.06532	0.00003
S1L5.L	LF.B	Logistic-regress	0.15719	0.00033	0.06257	0.00035
S2L2.L	LF.B	Logistic-regress	0.14692	0.00041	0.06386	0.00004
LF.A.L	LF.A	Logistic-regress	0.16341	0.00015	0.07112	0.00013
LF.B.L	LF.B	Logistic-regress	0.15483	0.00046	0.06995	0.00017

features, which also has been drawn from the experiments of comparison between different early fusion schemes.

Among the late fusion schemes in Table 4, the scheme LF.A.L wins with the highest mean average precision of 0.16341 (TREC09) and 0.07112 (TREC10), which adopted fusion structure (A) and applied logistic regression to derive combination weights through learning effects of corresponding features. It is hard to make the decision which kind of late fusion method is the best choice for semantic indexing. However, the supervised learning method achieves a stable performance with a smaller variance, and shows its capability to gain the optimal combining weights and to control the level of weighting sparsity.

3.3.3 Comparison between early fusion and late fusion

After analysis and comparison of combination structures for early fusion and late fusion, respectively, we conclude that systems with two-layer early fusion outperform the ones using late fusion schemes to a great extent in terms of building a robust, generic semantic analysis and indexing system which contains manifold concepts as illustrated in Fig. 12.

In particular, two-layer multiple kernel learning methods used for early fusion are of significance in statistics in spite of consuming much time, which do not only consider the description capability of the low-level features and the capability of corresponding bottom-level kernels modeling the concepts, but also take account of the mutual information of the lower-level kernels within the upper-level multiple kernel to try to make an effective combination of them.

In contrast to early fusion, late fusion schemes only utilize the classification scores of SVMs fed with single features, which lose the mutual information of low-level features and kernels and are unable to achieve the desired effect obtained by early fusion.

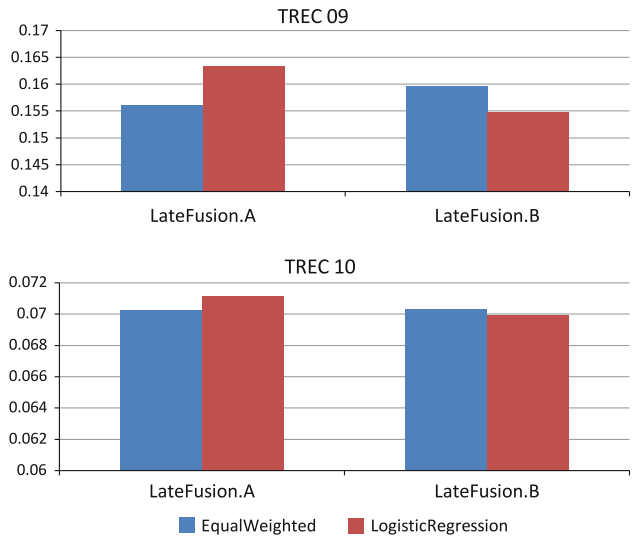

Fig. 11 Performance comparison between two late fusion schemes. In scheme A, logistic regression has a better performance, while for scheme B, equally weighting achieves the highest score for both TREC 09 and 10

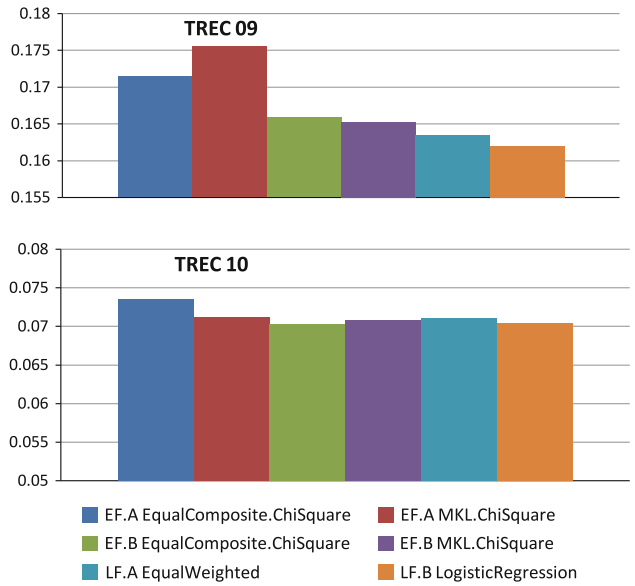

Fig. 12 Performance comparison between two late fusion schemes. EF and LF are the fusion strategies meaning early fusion and late fusion, respectively. In both TREC 09 and 10, Early Fusion with multi-kernel learning is proved to be a robust fusion strategy with satisfied performance

4 Conclusions and future work

In this paper, two-layer fusion schemes are presented to enhance the efficiency of video semantic concepts analysis. An exhaustive use of machine learning is made to gain the better fusion performance: support vector machines are used to learn the semantic information from various low-level visual features, and multiple kernel learning and logistic regression are used to combine kinds of sub-system on classifier level and on kernel level to achieve a robust semantic indexing system.

Experimental results evaluated by the dataset TRECVID 09 and 10 show that the proposed multilayer fusion schemes, especially the early fusion with two-layer equally weighted composite kernel learning, are more effective and robust to integrate different aspects of knowledge about concepts and achieve better indexing performance, than any of single low-level features. Comparing the early and late fusion approaches, early fusion based on multiple kernel learning shows more significance in statistic and is more generic for use; however, it is not up to much better performance than other combination methods, which might be related to limited number of positive samples for some concept and large intra-concept variability. The performance of late fusion is subject to the classification results achieved by using any single low-level features. The descriptive and discriminative features will improve the late fusion to some extent. In future, we plan to expand our set of low-level features, and moreover add the context into kernel learning stage and to try more effective learning and fusion methods.

Acknowledgments This work is sponsored by collaborative Research Project *SEV*01100474 between Beijing University of Posts and Telecommunications and France Telecom R&D Beijing, and National Natural Science Foundation of China 90920001.

References

1. Lienhart R, Kuhmunch C, Effelsberg W (1997) On the detection and recognition of television commercials. In: Proceeding of the IEEE conference on multimedia computing and systems, pp 509–516
2. Zhang H, Tan SY, Smoliar SW, Yihong G (1995) Automatic parsing and indexing of news video. Multimed Syst 2:256–266
3. Rui Y, Gupta A, Acero A (2000) Automatically extracting highlights for TV baseball programs. In: Proceedings of the eighth ACM international conference on multimedia, pp 105–115
4. Snoek G, Worring M et al (2006) The semantic pathfinder: using an authoring metaphor for generic multimedia indexing. IEEE Trans Pattern Anal Mach Intell 28:1678–1689
5. Cees G.M. Snoek, Koen E.A. van de Sande et al (2010) The MediaMill TRECVID 2010 Semantic Video Search Engine TRECVID Workshop
6. Cees G.M. Snoek et al (2005) Early versus late fusion in semantic video analysis. In: ACM MM'05
7. Kieran Mc Donald, Alan F. Smeaton (2005) A comparison of score, rank and probability-based fusion methods for video shot retrieval
8. Ayache S, Gensel J, Qu'enot GM (2006) Clips-lsr experiments at trecvid 2006—draft. In:TREC Video Retrieval Workshop, NIST
9. Dong Y et al (2009) The france telecom orange labs (beijing) video high-level feature extraction systems—trecvid 2009 notebook paper. TRECVID Workshop
10. Dong Y, Tao K et al (2010) The france telecom orange labs (beijing) video semantic indexing systems—trecvid 2010 notebook paper. TRECVID Workshop

11. Amir A, Argillander J, Campbell M et al (2005) IBM research trecvid-2005 video retrieval system. NIST TRECVID-2005 Workshop
12. Souvannavong F, Huet B (2005) Hierarchical genetic fusion of possibilities. In: Proceedings of the European workshop on the integration of knowledge. Semantic and Digital Media Technologies
13. Xue X, Lu H, Wu L et al (2005) Fudan university at trecvid 2005. In: TREC Video Retrieval Workshop, NIST
14. Liu J, Zhai Y, Basharat A et al (2006) University of central florida at trecvid 2006 high-level feature extraction and video search. In: TREC Video Retrieval Workshop, NIST
15. Yuan J, Guo Z, Lv L et al (2007) Thu and icrc at trecvid 2007. In: TREC Video Retrieval Workshop, NIST
16. Tang S, Zhang YD, Li JT et al (2007) Trecvid 2007 high-level feature extraction by mcg-ict-cas. In: Proceedings of the TREC-VID, NIST
17. M. Li, Y. T. Zheng, SX Lin et al (2009) Multimedia evidence fusion for video concept detection via owa operator. In: MMM'09, pp 208–216
18. Yuan J, Wang H, Xiao L et al (2005) Tsinghua university at trecvid 2005. In: TREC Video Retrieval Workshop, NIST
19. Cooper M, Adcock J, Chen R et al (2005) Fxpal at trecvid 2005. In: TREC Video Retrieval Workshop, NIST
20. Naphade MR, Mehrotra R et al (1998) A high performance algorithm for shot boundary detection using multiple cues. In: Proceedings of the IEEE International Conference on Image Processing, pp 884–887
21. Hadjidemetriou E, Grossberg MD, Nayar SK (2004) Multiresolution histograms and their use for recognition. IEEE Trans Pattern Anal Mach Intell 26:831–847
22. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput V 60:91–110
23. Pass G, Zabih R, Miller J (1997) Comparing images using color coherence vectors. In: Proceedings of the fourth ACM international conference on Multimedia, pp 65–73
24. Huang J, Ravi Kumar S, Mitra M, Zhu W, Zabih R (1999) Spatial color indexing and applications. Int J Comput V 35:245–268
25. Willamowski J, Arregui D, Csurka G, Dance CR, Fan L Categorizing nine visual classes using local appearance descriptors. illumination, vol 17
26. Liang Y, Liu X, Wang Z et al (2008) THU and ICRC at trecvid
27. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: Proceedings of the 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). IEEE Computer Society 1:886–893
28. Bosch A, Zisserman A, Muoz X (2008) Scene classification using a hybrid generative/discriminative approach. IEEE Trans Pattern Anal Mach Intell 30:712–727
29. Muller KR, Mika S, Ratsch G et al (2001) An introduction to kernel-based learning algorithms. IEEE trans neural netw 12:181–201
30. Collobert R, Bengio S (2001) Svmtorch: support vector machines for large-scale regression problems. J Mach Learn Res 1:143–160
31. Akbani R., Kwek S, Japkowicz N (2004) Applying support vector machines to imbalanced datasets. In: Proceedings of the 15th European conference on machine learning, pp 39–50
32. Zhang J, Marszaek M, et al (2007) Local features and kernels for classification of texture and object categories: a comprehensive study. Int J Comput Vision 73:213–238
33. Rakotomamonjy A, Bach F et al (2007) More efficiency in multiple kernel learning.In: Proceedings of the 24th international conference on machine learning. ACM, Corvalis, Oregon, pp 775–782
34. Longworth C, Gales M (2009) Combining derivative and parametric kernels for speaker verification. IEEE Trans Audio Speech Lang Process 17:748–757
35. Kraaij W, Awad G (2009) TRECVID 2009 High-Level Feature Task: Overview. http://www-nlpir.nist.gov/projects/tvpubs/tv9.slides/tv9.sin.slides.pdf, NIST
36. Quenot G, Awad G (2010) TRECVID 2010 Semantic Indextion Task. http://www-nlpir.nist.gov/projects/tvpubs/tv10.slides/tv10.hlf.slides.pdf, NIST
37. Fan RE et al (2009) LIBLINEAR: A library for large linear classification journal of Machine Learning Research, pp 1871–1874