Double Fusion for Multimedia Event Detection

Zhen-zhong Lan, Lei Bao, Shoou-I Yu, Wei Liu, and Alexander G. Hauptmann

School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA
{lanzhzh,iyu,alex}@cs.cmu.edu.com, {lei.bao.cn,lwbiosoft}@gmail.com

Abstract. Multimedia Event Detection is a multimedia retrieval task with the goal of finding videos of a particular event in an internet video archive, given example videos and descriptions. We focus here on mining features of example videos to learn the most characteristic features, which requires a combination of multiple complementary types of features. Generally, early fusion and late fusion are two popular combination strategies. The former one fuses features before performing classification and the latter one combines output of classifiers from different features. In this paper, we introduce a fusion scheme named double fusion, which combines early fusion and late fusion together to incorporate their advantages. Results are reported on TRECVID MED 2010 and 2011 data sets. For MED 2010, we get a mean minimal normalized detection cost (MNDC) of 0.49, which exceeds the state of the art performance by more than 12 percent.

Keywords: Feature Combination, Early Fusion, Later Fusion, Double Fusion, Multimedia Event Detection.

1 Introduction

In recent years, due to its great potential for many applications, the explosive growth of the user generated online videos and the prevailing online communities such as YouTube, Hulu etc., automatic detection of complex events in unconstrained videos has received a lot of interest from the research community [1] [2] [3]. However, most current tools only focus on single modality such as automatic transcription of speech from audio signal, scene recognition using color features or action detection based on time-related features. How to combine these state-of-the art approaches to build an accurate, fast and robust multimedia system to help users to study these overwhelming video data is still an open question. Many research in progress during the past few years still focus on the following two tracks: the design of highly discriminative and robust features [4] and the combination of multiple complementary features based on different modalities such as visual, audio and text [5] [6] [7] [9] [10]. For example, in 2010, NIST held the first Multimedia Event Detection (MED) evaluation [7] [10], which emphasis the importance of combining multiple modalities for event detection. As shown in Fig. 1, the task is: given an Event Kit (including an description of the concepts and some example videos), find videos that belong to the event defined by the Event Kit. In this paper, we will deal with the same task.

K. Schoeffmann et al. (Eds.): MMM 2012, LNCS 7131, pp. 173–185, 2012.

Event Name: Batting a run in
Definition: Within a single play during a baseball-type game, a batter hits a ball and one or more runners (possibly including the batter) scores a run.
Evidential Description:
scene: outdoor or indoor ball fields (official or ad hoc), during the day or night
objects/people: baseball, bat, glove, crowd in background, fence, pitchers mound, bases, other players, officials
activities: pitching, swinging a bat, running, throwing a ball, cheering or clapping, making a call, crossing home plate
Exemplars:

Fig. 1. The illustration MED task

Many research papers [5] [7] [8] [9] [10] state that a multimodal approach helps to obtain an effective retrieval/classification performance on image and video. In general, there are two types of combination strategies, namely early fusion and late fusion [9]. Early fusion combines feature before performing classification, such as multi-kernel learning [11] [12]. Late fusion combines output of classifiers from different features, such as average fusion, committee voting [13] and co-regularized least squared regression [14]. There is no universal conclusion of which strategy is the preferred method for multimedia content analysis and retrieval. [9] found that early fusion is better than late fusion in semantic indexing based on their results on TRECVID 2004 benchmark. While studying data on TRECVID 2006, [15] found that early fusion gets better results on most of concepts while late fusion is more robust and can tackle some harder concepts. To incorporate the advantages of both methods, we introduce a simple yet efficient fusion strategy called double fusion. In double fusion, we first perform early fusion to generate different combinations of features from subsets on the single features pool. After that, we train classifiers on each feature or feature combination and carry out late fusion on the output of these classifiers. For example, as shown in Fig. 2, we first extract three kinds of features (visual, audio and text) from three training and three testing videos. After that, pairwise early fusion (visual+audio, visual+text) are carried out in these three features based on their kernel matrice. In the training step, five classifiers are trained based on five features and feature combinations (visual, audio, text, visual+audio, visual+text). For each video, there are thus five output scores indicating how likely it is that

this video belongs to the event. In the last step, late fusion is used to fuse five output score vectors into one score vector, on which the final interpretation can be executed. Experimental results on the TRECVID MED 2010 and MED 2011 data sets with about 484 hours' video clips for 18 events show the effectiveness of double fusion. For MED 2010 we get a mean minimal normalized detection cost (MNDC) of 0.49, which exceeds the state of the art performance [7] by more than 12 percent.

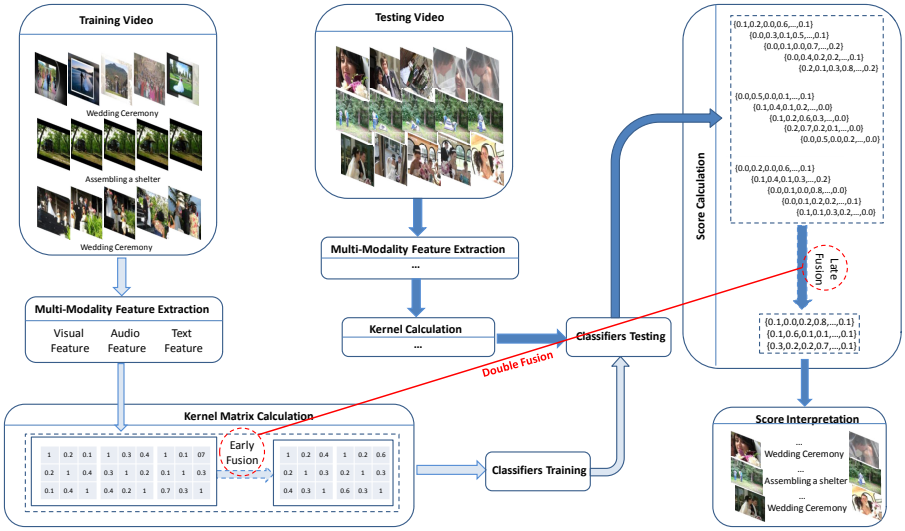

Fig. 2. The illustration of our MED system

The remainder of the paper is organized as follows. Section 2 briefly introduces different fusion strategies. Section 3 presents the details of our implementation, including feature representation, bag-of-words scheme, classifiers and fusion schemes. Section 4 demonstrates and analyzes experimental results on MED 2010 and MED 2011. Finally, section 5 concludes the paper and outlines our future work.

2 Fusion Scheme

Early Fusion [9] is a combination scheme that runs before classification. Both feature fusion and kernel space fusion are example of early fusion. The main advantage of early fusion is that only one learning phase is required. However, it is hard to combine features into a common representation [9]. Multiple kernel learning is one of the most popular early fusion technologies. Its drawback is the curse of high dimensionality, usually accompanied by limited training data.

In contrast to early fusion, late fusion [9] happens after classification. While late fusion is easier to perform, in general, it needs more computational effort and

has potential to lose the correlation in mixed feature space. Normally, another learning procedure is needed to combine these outputs, but in general, because of the overfitting problem, simply averaging the output scores together yields better or at least comparable results than training another classifier for fusion.

As shown in paper [9], there is no conclusion about which fusion scheme will get better performance. For some concepts such as stock quotes, early fusion get better result, for other concepts such as road, late fusion get better performance. Could we come up a solution to combine the strengths of both early and late fusion? In this paper, we introduce a method called double fusion, which combines early fusion and late fusion together. Specifically, for early fusion, we fuse multiple subsets of single features by using standard early fusion technologies; for late fusion, we combine output of classifiers trained from single and combined features. By using this scheme, we can freely combine different early fusion and late fusion techniques, and get benefits of both.

Two early fusion strategies, i.e., rule-based combination and multiple kernel learning [12], are used to combine kernels from different features. For rule-based combination, we use the average of the kernel matrix. Multiple kernel learning [12] is a natural extension of average combination. It aims to automatically learn the weights for different kernel matrix. Our experimental results show that the performance of multiple kernel learning is slightly better than average combination. However, because of the explosive number (the number of combination is $2^n - 1$, n is the number of features) of combination, it is time consuming to use all possible feature combination when the feature space becomes large. To address this problem, our first possible solution is by combing features belonging the same categories. For each category or single feature, we train one classifier. The number of classifiers for late fusion will be n+c, in which c is the number of category. Our second solution is to combine all features together in early fusion and perform late fusion with all single feature classifiers that results in n+1 classifiers need to be fused in later fusion. In this paper, we use both approaches and train n+c+1 classifiers for late fusion, in which there are c early fusion classifiers built on category-based features, n single feature classifiers and one early fusion classifier trained on the combination of all features. This allows us to exploit the advantages of single feature classifier, category-based classifier and complete-feature classifiers.

3 Implementation

As shown in Fig. 2, there are four key steps in our system. In step one, we perform feature extraction on visual, textual and audio modality. After modality specific data processing, bag-of-words representation is used to aggregate the point features into whole video features. Early fusion is applied in step two after calculating the kernel matrix. In step three, classifiers are trained to perform the classification. The outputs of different classifiers are combined by using late fusion strategies in step four.

Feature Extraction and Feature Representation. Feature representation is critical for video content understanding. In TRECVID MED System, we explore three feature modalities including visual features, audio features and text features.

Visual Feature. We use five visual features, namely SIFT [20], CSIFT [16], MoSIFT [17], STIP [18] and GIST [4] .

For SIFT feature and CSIFT, the harris-laplace key point detector is used to detect key points. As processing all MED video frames will be computationally expensive, we only extract features from key frames extracted by a shot boundary detection algorithm. Specifically, the algorithm calculates the color histogram for every five frames and subtracts the histogram with the histogram of the previous frame, if the subtracted value is larger than a certain threshold, which is empirically setted, the key frame will be a shot boundary. After detecting the shot, we use the frame in the middle of the shot to represent that shot. By using this algorithm, we extracted 114992 key frames from MED 2010 and 364747 key frames from MED 2011 development data.

While SIFT and CSIFT describe 2D local structure in images, space-time interest points (STIP) and MoSIFT capture space time volumes where the image values have significant local variations in both space and time. STIP and MoSIFT are different in both key points detector and descriptor. STIP uses 3D Harris corner detectors and its key points are represented in two parts: the first part is HOG (Histograms of Oriented Gradients; 72 dimensions) which indicates the spatial appearance and the second part is HOF (Histograms of Optical Flow; 90 dimensions) describing the motion information. MoSIFT uses a Difference of Gaussian (DoG) based detector and represents by another descriptor which is also concatenated from two parts: the first part is SIFT (128 dimensions) which indicates the spatial appearance and the second part is also HOF (128 dimensions).

For the GIST feature, we follow the suggestion from [4] and set the dimension of feature points to 960.

Audio Feature. For the audio feature, we used Automatic Speech Recognition (ASR) feature, which is extracted as described in [10].

Textual Feature. Following the work of [10], we use Optical Character Recognition (OCR) feature extracted by the Informedia system to represent the text feature.

Bag-of-words Representation. After extracting above features from given videos, a formal Bag-of-words representation is adopted to cast features of key frames into fixed length feature vector. First, vector quantization (VQ) technique is used to cluster feature descriptors into a large number of clusters (i.e. 'words') using k-means clustering algorithm. For visual features, the code book size is 4096 except for GIST, which has 960 dimensions. Second, by mapping these features into their cluster centroid, we can get a feature representation for each key frame. Here, we adopt a soft-weight strategy in which we choose the ten

nearest clusters and assigned a rank weight for them. For using these words to represent the videos, we need to cast image feature into video feature. For SIFT, CSIFT and GIST, we first normalize feature vectors of each key frame in a video and then sum them together to represent the video. For STIP and MoSIFT, we just sum all the feature points in a video together and normalize it. As for ASR and OCR, we simply count the number of words or tokens found in videos. There are a total of 11618 unique words and 180228 unique tokens extracted for ASR and OCR, respectively.

Spatial Pyramid Matching. Since the classic bag-of-words method loses all information about the spatial layout of features,[19] adopt the pyramid matching scheme by repeatedly subdividing the image and computing histograms of local features for each sub-regions. Specifically, besides the bag-of-word representation for the whole image, we divided the keyframe into 2x2 and 1x3 sub-regions, and computed the bag-of-word representation for each sub-region. Thus, the feature dimension for the spatial pyramid matching is 8x4096=32768. We applied this simple yet effective method for SIFT and CSIFT features.

Classifier. A large variety of classifiers exist for mapping the feature space into score space. In this paper, we adopt two classifiers, i.e. non-linear support vector machine (SVM) [21] and kernel regression (KR) [14]. SVM is one of the most commonly used classifier due to its simple implementation, low computational cost, relatively mature theory and high performance. In TRECVID MED 2010, most of the teams [7] [8] use SVM as their classifiers. Compared to SVM, KR is a simpler but less used algorithm. However, our experiment shows that the performance of KR is consistently better than the performance of SVM.

Fusion. In our feature set, only visual feature set has multiple features, while all other features represent each category by its own. By performing visual feature (SIFT, CSIFT, MoSIFT, STIP, GIST) combination and all-feature (SIFT, CSIFT, MoSIFT, STIP, ASR, OCR, GIST) combination, we have two feature combination and seven single features (SIFT, CSIFT, MoSIFT, STIP, ASR, OCR, GIST). For late fusion, we use two rule-based fusion methods to combine the output of above 9 classifiers. One is average combination, another one is weighted combination using weight learned from cross-validation. The detail of the weight calculation will be given in the experimental part.

4 Experiment

4.1 Data

For TRECVID MED 2010, we used both the annotated training and testing data, which consists of 114 hours of video clips and three event kits, i.e., "Making a cake","Batting a run" and "Assembling a shelter". For MED 2011, currently, we only have the annotated development data of MED 2011, which consists of about 370 hours of video clips and 15 events including 5 training events (Attempting a board trick, Feeding an animal, Landing a fish, Wedding ceremony

and Working on a woodworking project) and 10 testing events (Birthday party, Changing a vehicle tire, Flash mob gathering, Getting a vehicle unstuck, Grooming an animal, Making a sandwich, Parade, Parkour, Repairing an appliance and Working on a sewing project). To test the performance of our system on MED 2011 dataset, we manually split the 10 testing events into same size of training and testing data. After the splitting, we have 3135 video clips for training and a 6687 video set for testing on MED 2011.

We ran our program on the Carnegie Mellon University Parallel Data Lab cluster, which contains 300 cores and it took us about 57000 CPU hours to extract features and perform the bag-of-words mapping.

4.2 Evaluation

For performance comparison, two evaluation schemes are adopted: the first one is the MNDC, which, as indicated in formula 1, is an evaluation criteria for NIST to evaluate MED 2010 and MED 2011. Lower MNDC indicates better performance. For better understanding, we also use maximum F1 Score by using test label to search the best threshold. Considering that we have 100 times negative sample than positive samples for each event, MNDC is still a better criteria for evaluation since it gives more weight on the cost of false alarm. However, both of above two criteria are highly depended on threshold and are not stable for evaluation.

$$NDC(S,E) = \frac{C_M * P_M(S,E) * P_T + C_{FA} * P_{FA}(S,E) * (1 - P_{FA}(S,E))}{MINUMUM(C_M * P_T, C_M * (1 - P_T))} \quad (1)$$

where $P_M(S,E)$ is the missed detection probability for system S, event E while $P_{FA}(S,E)$ is the false alarm probability for system S, event E. $C_M = 80$ is the cost for missed detection, $C_{FA} = 1$ is the cost for false alarm and $P_T = 0.001$.

4.3 Parameter Selection

For both SVM and KR, we used a χ^2 kernel [22] since all of our features are histogram features and the χ^2 kernel has been extensively used for histogram features. A parameter γ is needed for χ^2 kernel. For SVM, we have one additional regularization parameter C. To optimize these parameters, we ran two-folded cross-validation 10 times by randomly spliting the training data into two folds. Then, the average MNDC of two folds are used to choose the best parameters. We also use the average MNDC to generate weights to perform weight averaging for late fusion. The search ranges for both C and γ are 10^{-3} to 10^3, in multiples of 10. We did try small step size search for parameter selection suggested by [23], but didn't find much difference.

4.4 Results

To get a statistically meaningful experiment, for each setting, we run 10 times and calculate the mean and standard deviation for that setting. Because running

Table 1. Comparison of single features on TRECVID MED2010. Two evaluation criteria including MMNDC and MMF1 are adopted. For MMNDC, lower score indicates better performance; for MMF1, higher score means better performance.

Feature	MMNDC %± STD	MMF1%± STD
CSIFT	60.6 ± 0.7	52.5 ± 0.6
SIFT	**60.5 ± 1.4**	**53.3 ± 1.1**
MoSIFT	63.9 ± 1.4	50.6 ± 0.9
STIP	69.1 ± 0.5	48.2 ± 1.8
GIST	82.9 ± 1.5	33.7 ± 0.7
ASR	89.1 ±4.7	22.5 ± 4.1
OCR	85.7 ± 0.1	28.8 ± 0.8

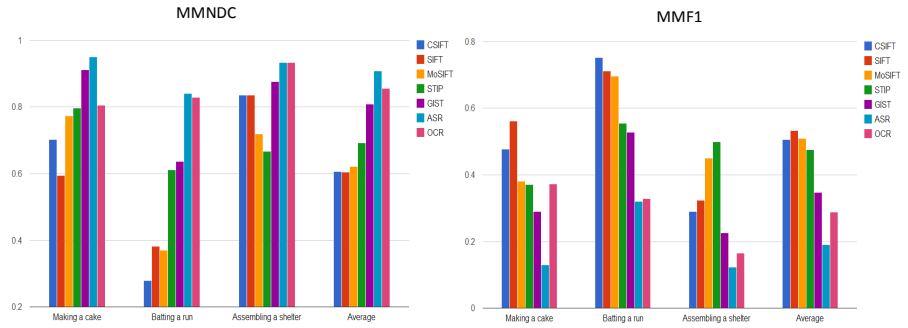

Fig. 3. Comparison of single feature on TRECVID MED2010. Two evaluation criteria including MMNDC and MMF1 are adopted. For MMNDC, lower score indicates better performance; for MMF1, higher score means better performance.

all the combination of fusion strategies and classifiers will be computational expensive and meaningless to our concern, we first compare all the classifiers, early fusion and late fusion strategies on MED 2010 and choose the best strategy for each step to perform further experiments on MED 2011.

Single Feature Comparison. First, we compare the mean MNDC (MMNDC) (lower MMNDC indicates better performance) and mean MF1 (MMF1) (higher MMF1 indicates better performance) of single features on MED 2010. As shown in Table 1 and Fig. 3, the performance of different features vary dramatically from event to event. Generally, four local features including CSIFT, SIFT, MOSIFT and STIP consistently outperform other three features. In these four features, motion based features including MOSIFT and STIP get much better results than static features including SIFT and CSIFT in "Assembling a shelter " event, which has a lot of motion. Contradictorily, static features are obviously superior to other features in "Batting a run" event and "Making a cake", because of their relatively monotonous background. Different matched situation for different features shows that above features are complementary to each other.

Table 2. Comparison of classifiers, early fusion and late fusion strategies on TRECVID MED 2010. Two evaluation criteria including MMNDC and MMF1 are used. For MMNDC, lower score indicates better performance; for MMF1, higher score means better performance.

	Classifiers		Early Fusion		Late Fusion	
	KR	SVM	MKL	Average Fusion	Weighted Fusion	Average Fusion
MMNDC% ± STD	**60.5 ± 1.4**	62.3 ± 1.1	**50.6 ± 0.8**	50.7 ± 0.6	**52.5 ± 1.5**	57.6 ± 1.9
MMF1% ± STD	**53.3 ± 1.1**	50.7 ± 2.9	**61.4 ± 0.1**	61.2 ± 0.6	**59.7 ± 1.1**	54.4 ± 1.6

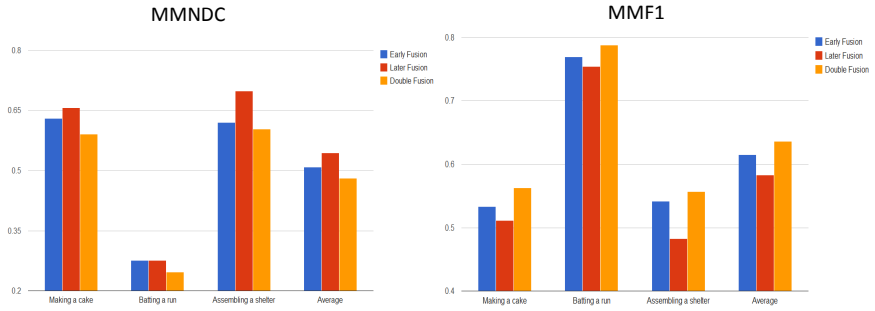

Fig. 4. Comparison of double fusion with early fusion and late fusion on MED 2010. Two evaluation criteria including MMNDC and MMF1 are adopted. For MMNDC, lower score indicates better performance; for MMF1, higher score means better performance.

Also, the performances of ASR and OCR features are much worse than those visual feature. All of these indicate that giving different weights for different features is a promising fusion strategy.

KR versus SVM. We further compared the performance of different classifiers by simply using the best single feature, which is SIFT. From Table 2, we can see that, compared to SVM, KR has lower MMNDC and higher MMF1, which indicate that KR is a better classifier for TRECVID MED task. From now on, we will use KR as our classifier for further experiments in this paper.

Early Fusion Strategies Comparison. For early fusion, we choose either multiple kernel learning oraverage fusion. As indicated in Table 2, we can see that MKL only gets comparable results to simple average fusion, this is consistent with what was suggested by [12]. Considering that the performances of some features are much worse than other features, it is quite unreasonable to give them equal weight. However, finding a better weight strategy is still an open question.

Late Fusion Strategies Comparison. Table 2 shows the results of late fusion using weighted fusion and average fusion. The result of weighted late fusion is

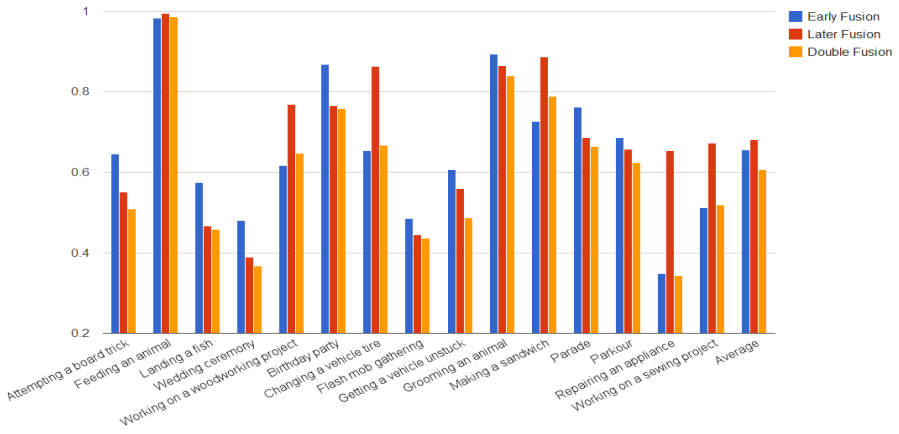

Fig. 5. Comparison of double fusion with early fusion and late fusion on MED 2011 by suing MMNDC criteria. Lower MMNDC indicates better performances.

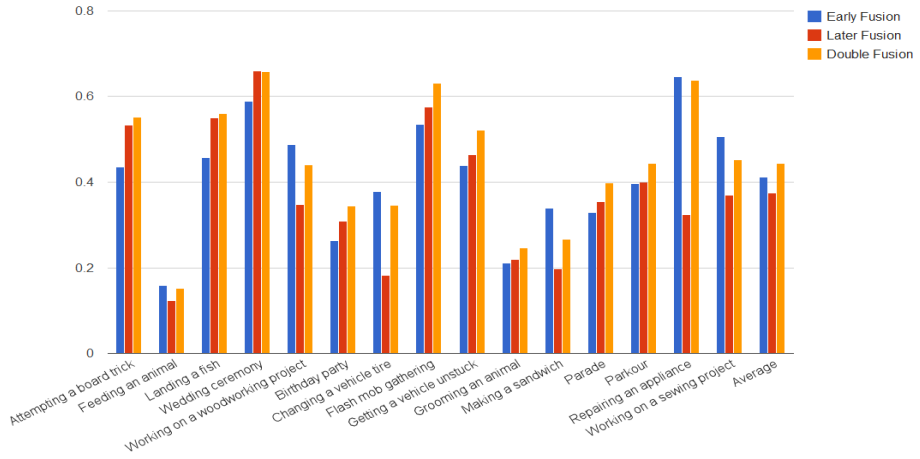

Fig. 6. Comparison of double fusion with early fusion and late fusion on MED 2011 by using MMF1 criteria. Higher MMF1 indicates better performances.

much better than the result of average late fusion. This indicates that different features have different contributions to the final results, especially when the performance varies dramatically between features. We will only use the weighted combination for late fusion for further comparison.

Double Fusion Versus Early Fusion and Late Fusion. The result of double fusion is shown in Table 3. From the table, we can see that double fusion gives much better results than both early and late fusion. The current best result

Table 3. Comparison of double fusion with early fusion and late fusion on MED2010. Two evaluation criteria including MMNDC and MMF1 are adopted. For MMNDC, lower score indicates better performance; for MMF1, higher score means better performance.

	MED 2010			MED 2011		
	Early Fusion	Late Fusion	Double Fusion	Early Fusion	Late Fusion	Double Fusion
MMNDC% ± STD	50.6 ± 0.8	52.5 ± 1.5	**48.9 ± 0.7**	65.6 ± 0.7	68.2 ± 1.3	**60.6 ± 0.8**
MMF1% ± STD	61.4 ± 0.1	59.7 ± 1.1	**62.9 ± 0.6**	41.1 ± 0.5	37.4 ± 3.8	**44.3 ± 0.9**

on TRECVID MED 2010 was achieved [7] using the MMNDC criteria and the performance was 0.565. Compared to this result, we get more than 12 percentages improvements in MMNDC, though results are not perfectly comparable due to different features and machine learning methods. Fig. 4 shows that double fusion gets consistently better performance than early fusion and late fusion on all of three events in MED 2010. MED 2011 is much harder and more diverse than MED 2010 since we have 15 events now, but Fig. 5 and Fig. 6 indicate that double fusion still gets better performance than early fusion and late fusion on 11 of 15 events. For the other 4 events, double fusion still gets similar results to the best methods for those events, which indicates that double fusion does capture advantages of both early fusion and late fusion.

5 Conclusion

In this paper, we presented an analysis of early fusion and late fusion which aims at combining features from different modalities for multimedia event detection and introduced a double fusion scheme which combines early fusion and late fusion together. Our experiments on about 484 hours of videos come from TRECVID MED 2010 and 2011 showed that this simple strategy is very effective and had a substantial advantage over both early fusion and late fusion strategies. Moreover, we found that weighted combination is better than average combination for late fusion but not for early fusion. How to learn weight for early combination is still an open question, our future work will focus on learning weight for early fusion.

Acknowledgments. This work was supported in part by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20068. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes not withstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government. Support was also provided, in part, by the National Science Foundation, under award CCF-1019104, and the Gordon and Betty Moore Foundation, in the eScience project. We thank the Parallel Data Lab for the use of their resources.

References

1. Smeaton, A.F., Over, P., Kraaij, W.: Evaluation campaigns and TRECVid. In: 8th ACM International Workshop on Multimedia Information Retrieval, MIR 2006 (2006)
2. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videios 'in the wild'. In: IEEE Conference on Computer Vision and Pattern Recognition CVPR 2009 (2009)
3. Hauptmann, A., Yan, R., Lin, W., Christel, M., Wactlar, H.: Can high- level concepts fill the semantic gap in video Retrieval? A case study with broadcast news. IEEE Transaction on Multimedia 9(5), 958–966 (2007)
4. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. Int. J. Computer Vision 42(3), 145–175 (2001)
5. Yang, Y., Zhuang, Y., Wu, F., Pan, Y.: Harmonizing hierarchical manifolds for multimedia document semantics understanding and cross-media retrieval. IEEE Transactions on Multimedia (TMM 2008) 10(3), 437–446 (2008)
6. Liu, J., Yang, Y., Shah, M.: Learning semantic visual vocabularies using diffusion distance. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009 (2009)
7. Jiang, Y.G., Zeng, X.H., Chang, S.F., et al.: Columbia-UCF TRECVID 2010 multimedia event detection: combining multiple modalities, contextual concepts, and temporal matching. In: Proceeding TRECVID Workshop (2010)
8. Iyengar, G., Nock, H., Neti, C.: Discriminative model fusion for semantic concept detection and annotation in video. In: Proceedings of 11th Annual ACM International Conference Multimedia, MM 2003 (2003)
9. Snoek, C.G.M., Worringm, M., Smeulders, A.W.M.: Early versus late fusion in semantic video analysis. In: Proceedings of 13th Annual ACM International Conference Multimedia, MM 2005 (2005)
10. Li, H., Bao, L., Hauptmann, A., et al.: Informedia@ TRECVID 2010. In: Proceedings of TRECVID Workshop (2010)
11. Gehler, P., Nowozin, S.: On feature combination for multiclass object classification. In: Proceedings of International Conference Computer Vision, ICCV 2009 (2009)
12. Cortes, C., Mohri, M., Rostamizadeh, A.: L_2 regularization for learning kernels. In: Proceedings of Uncertainty Artitical Intelligence, UAI 2009 (2009)
13. Erp, M.V., Vuurpijl, L.G., Schomaker, L.: An overview and comparison of voting methods for pattern recognition. In: Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition, IWFHR-8 (2002)
14. Brefeld, U., Gaertner, T., Scheffer, T., Wrobel, S.: Efficient co-regularized least squares regression. In: Proceedings of the 23rd International Conference of Machine Learning, ICML 2006 (2006)
15. Ayache, S., Quénot, G., Gensel, J.: Classifier Fusion for SVM-Based Multimedia Semantic Indexing. In: Amati, G., Carpineto, C., Romano, G. (eds.) ECIR 2007. LNCS, vol. 4425, pp. 494–504. Springer, Heidelberg (2007)
16. van de Sande, K.E.A., Gevers, T., Snoek, C.G.M.: Evaluation of color descriptors for object and scene recognition. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008 (2008)
17. Chen, M.Y., Hauptmann, A.: MoSIFT: Recognition human actions in surveillance videos. Technological report, CMU-CS-09-161, Carnegie Mellon University (2009)
18. Laptev, I., Lindeberg, T.: Space-time interest points. In: Proceedings of International Conference Computer Vision, ICCV 2003 (2003)

19. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Computer Vision and Pattern Recognition 2006, CVPR 2006 (2006)
20. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision (IJCV 2004) 60(2), 91–100 (2004)
21. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines (2001)
22. Vedaldi, A., Fulkerson, B.: VLFeat: An Open and Portable Library of Computer Vision Algorithms (2008)
23. Bernhard, S., Burges, C.J.C., Smola, A.J.: Advances in kernel methods: Support Vector Learning. MIT Press, Cambridge (1999)