Feature Weighting via Optimal Thresholding for Video Analysis

Zhongwen Xu[†] Yi Yang[†§] Ivor Tsang[‡] Nicu Sebe[♮] Alexander G. Hauptmann[§]
[†]ITEE, The University of Queensland, Australia
[§]School of Computer Science, Carnegie Mellon University, USA
[‡]School of Computer Engineering, Nanyang Technological University, Singapore
[♮]Department of Information Engineering and Computer Science, University of Trento, Italy
z.xu3@uq.edu.au {yiyang,alex}@cs.cmu.edu IvorTsang@ntu.edu.sg sebe@disi.unitn.it

Abstract

Fusion of multiple features can boost the performance of large-scale visual classification and detection tasks like TRECVID Multimedia Event Detection (MED) competition [1]. In this paper, we propose a novel feature fusion approach, namely Feature Weighting via Optimal Thresholding (FWOT) to effectively fuse various features. FWOT learns the weights, thresholding and smoothing parameters in a joint framework to combine the decision values obtained from all the individual features and the early fusion. To the best of our knowledge, this is the first work to consider the weight and threshold factors of fusion problem simultaneously. Compared to state-of-the-art fusion algorithms, our approach achieves promising improvements on HMDB [8] action recognition dataset and CCV [5] video classification dataset. In addition, experiments on two TRECVID MED 2011 collections show that our approach outperforms the state-of-the-art fusion methods for complex event detection.

1. Introduction

The huge number of videos uploaded and viewed on the Internet makes video analysis a hot topic in computer vision and multimedia communities. Videos contain rich information which can be represented as motion features (*e.g.*, Space-Time Interest Points (STIP) [10], Dense Trajectories [23]), shape and color features of video frames (*e.g.*, SIFT [12], Color SIFT [21]), and acoustic features (*e.g.*, Mel-Frequency Cepstral Coefficients (MFCC)). However, not any individual feature can capture the whole information of a video. Even for a single feature, the state-of-the-art methods usually combine multiple descriptors. For example, STIP [10] feature combines HOG descriptor for shape information and HOF descriptor for motion information, Dense Trajectories feature [23] is an integration of descriptors of trajectory, HOG, HOF and Motion Boundary Histogram (MBH).

In the video action recognition and event detection tasks, researchers have developed systems which combine multiple features. While performing action recognition on large-scale video datasets, Reddy and Shah [17] found that combining scene features (*e.g.*, Color SIFT) with motion features (*e.g.*, STIP) is beneficial for analyzing real-life videos from the Internet. As for event detection tasks, reports from teams with top performance [26, 14, 15] in TRECVID MED competition show that fusion, either feature-level fusion or decision-level fusion brings performance gain into the detection tasks.

Fusion mechanisms can be grouped into two types which are feature-level fusion and decision-level fusion. In the feature-level fusion, a linear combination of kernel matrices from different features is used to capture the structure of video data [18]. One simple and effective way in the feature-level fusion, namely average early fusion, is to average multiple kernel matrices and the average kernel matrix is used as similarity measure for classifier training. The other fusion mechanism is decision-level fusion, which adopts classifiers to features and then fuses the results based on the confidence scores. Lan *et al*. [9] find that combining the decision values obtained from the kernel matrices of individual features and the average distances of all the features will gain better performance than using the decision values from each individual features only.

The most widely used decision-level fusion method is to assign average weights to confidence scores from each feature, which may restrain the overall performance due to the inconsistency and incomparability of confidence scores from different models. Intuitively, in decision-level fusion, different features should have different weights since they may not contribute equally to the final decision. Taking complex events detection in TRECVID MED task as an example. Table 1 shows the Average Precision (AP) of detection results from Dense Trajectories, STIP and MFCC

Event Name	MFCC	Trajectories	STIP
Birthday Party	21.3%	13.1%	7.8%
Changing a vehicle tire	3.9%	21.9%	4.1%
Working a sewing project	11.8%	17.4%	11.8%

Table 1. Average Precision for three different features

Event Name	MFCC	Trajectories	STIP
Birthday party	0.075	0.106	0.075
Changing a vehicle tire	0.059	0.085	0.075
Working a sewing project	0.046	0.091	0.086

Table 2. Thresholds for different models

respectively. In this experiment, χ^2-kernel SVM is used as the classifier. For the event "Birthday party", the acoustic feature MFCC achieves the best prediction performance, and it is much better than visual motion features. The reason is that singing and laughing sound in a birthday party is well captured by MFCC. Differently, for the event "Changing a vehicle tire", acoustic information becomes less discriminative so that MFCC gets worse performance than Dense Trajectories feature. The situation of STIP is the same as MFCC. It achieves good performance for some events while performs worse for others. In this example we can see that different features do not contribute equally to the task and therefore their weights should not be identical.

Another issue in decision-level fusion is the difference of thresholds among confidence scores from different models. Assume that we retrieve the top 500 videos among 32,000 testing videos according to the confidence scores. Table 2 shows that the threshold of confidence scores from different models can be very different. For example, Dense Trajectories feature has higher threshold than others, which means that in the prediction using Dense Trajectories feature, only videos with very high confidence scores should be considered as positive results. If the effects of the difference of thresholds among predictive results are ignored, it would degrade the discriminative ability of the fusion result.

In this paper, we propose a method for feature fusion. We name the proposed method Feature Weighting via Optimal Thresholding (FWOT). As aforementioned, the weights and thresholds of multiple features are two factors to be considered for feature fusion. In light of this, the fusion algorithm proposed in this paper integrates feature weighting and thresholds selection into a joint framework. Our premise is that the weight and threshold of each feature are correlated and the joint optimization of both makes them mutually beneficial and reciprocal. The optimal weight of a feature is dependent on the threshold, making it not only to accurately reflect the importance of the feature, but also more suitable for making the classification/detection decision. Inspired by [9], we combine the early fusion result at the decision-level fusion. To the best of our knowledge, this is the first work which optimizes weights and thresholds simultaneously for fusion. Instead of directly solving a non-convex and time consuming problem, we preset a series of thresholds as candidates, which in turn transforms the problem from detecting the optimal thresholds to selecting the best thresholds from the candidates. Further, to make the algorithm more flexible and robust, we additionally introduce a group of smooth factors to soften the classification/detection decision from discrete values to continuous domain. In this way, the algorithm is formulated as a Mixed Integer Program (MIP) problem. As the MIP problem is NP-hard, we relax it to a convex optimization problem, which is the lower bound of the original MIP problem. We then apply cutting plane algorithm to efficiently solve the problem with almost linear time complexity. In that way, the optimized weights and thresholds can be obtained.

2. Related Work

Multiple Kernel Learning (MKL) [16] is the most popular way for combining different kernels to utilize the advantages of different features in applications such as visual object classification, object detection and video semantic analysis. Vedaldi *et al.* [22] use the MKL method to learn the optimal combination of exponential χ^2-kernels of edges features, dense and sparse visual words and feature descriptors at different spatial levels. They successfully trained and tested a detector in a reasonable time and achieved the best performance on the PASCAL VOC 2007 and 2008 benchmarks. However, Gehler and Nowozin have recently pointed out in [4] that MKL may be less competitive than average combination when the individual kernels are discriminative already.

Recently, Yang *et al.* have proposed a semi-supervised algorithm to fuse the information from multiple features. The experiment shows that it is beneficial to exploit the unlabeled data for multiple feature fusion when the labeled data are few. Ma *et al.* propose to use multiple features to learn different types of video attributes for event detection. However, the algorithms proposed in [25] and [13] assign an equal weight to different features, even though the features may not be equally important.

Natarajan *et al.* [14, 15] propose a decision-level fusion method particularly for event detection. The algorithm adaptively fuses multiple features, which assigns videos with the weights based on the detection thresholds. The adaptive decision-level fusion assigns lower weights to specific scores if the confidence scores are near the threshold while assigns higher weights to videos if the confidence scores are very far away from the threshold. Thresholds are set before the fusion stage. Though it is a reasonable way to assign weights to features according to the detection threshold, this method highly depends on the preset detection threshold.

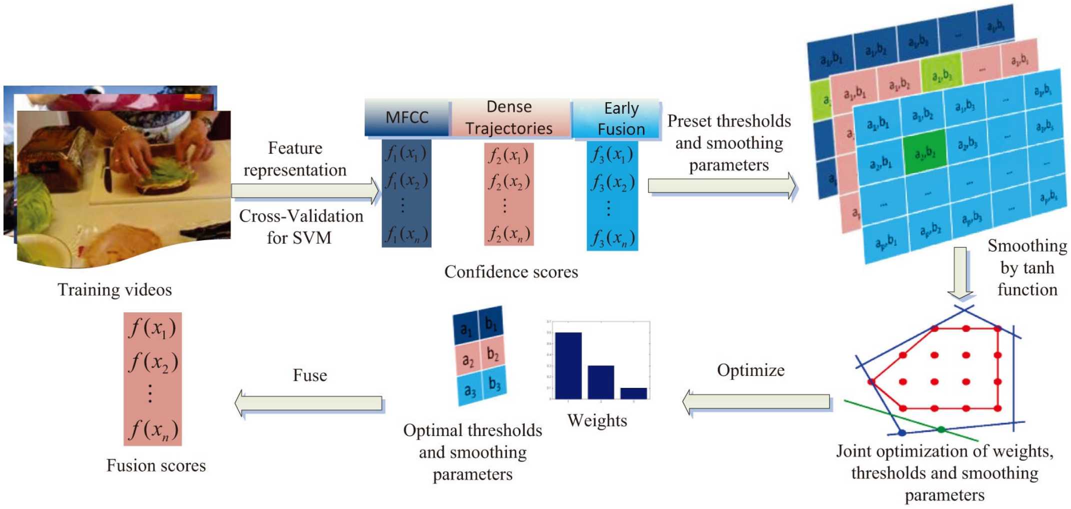
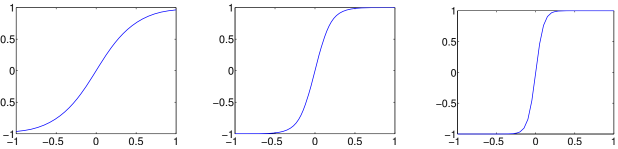

Training videos
Feature representation
Cross-Validation for SVM
MFCC
Dense Trajectories
Early Fusion
$f_1(x_1)$ $f_1(x_2)$ ⋮ $f_1(x_n)$
$f_2(x_1)$ $f_2(x_2)$ ⋮ $f_2(x_n)$
$f_3(x_1)$ $f_3(x_2)$ ⋮ $f_3(x_n)$
Confidence scores
Preset thresholds and smoothing parameters
Smoothing by tanh function
Joint optimization of weights, thresholds and smoothing parameters
Optimize
Weights
Optimal thresholds and smoothing parameters
Fuse
$f(x_1)$ $f(x_2)$ ⋮ $f(x_n)$
Fusion scores

Figure 1. An illustration of our Feature Weighting via Optimal Thresholding (FWOT) fusion method



$a = 0.5$ $a = 0.2$ $a = 0.1$

Figure 2. $f(x) = \tanh \frac{x}{a}$ of different parameters

3. The Proposed Approach

In this section, we first elaborate the formulation of the proposed approach. Then we show the detailed steps to obtain the optimal fusion function. Figure 1 is the illustration of our FWOT method.

3.1. Problem Formulation

Suppose there are n training videos, we denote each video as a variable $x_m \in \mathbb{R}^d (1 \leq m \leq n)$, and its label as $y_m \in \{-1, +1\}$, where $y_m = +1$ indicates x_m is a positive exemplar and $y_m = -1$ indicates x_m is a negative one. Assuming that we have t features, we can train classifiers $f_1(x), f_2(x), \ldots, f_t(x)$ according to features of videos. One simple function to combine the confidence scores is

$$f(x) = \sum_{i=1}^{t} w_i \operatorname{sgn}(f_i(x) - b_i), \tag{1}$$

where w_i and b_i are the weight and the threshold for confidence scores of the i-th feature respectively. The function in (1) indicates that for the i-th feature, if the confidence score is above the threshold b_i, the video would be labeled as $+1$; otherwise -1, and then we combine the label values according to weights w_i. However, the $\operatorname{sgn}(\cdot)$ function here makes the fusion process inflexible, since videos with much higher confidence scores than the threshold and those with confidence scores a little bit higher than the threshold would contribute equally to the fusion result. Instead of using the hard label function $\operatorname{sgn}(\cdot)$, we adopt the $\tanh(\cdot)$ function with a smoothing parameter a to generate soft labels. Figure 2 shows the curves of $\tanh(\cdot)$ when the parameter a is set to different values. It can be seen that when a is getting smaller, $\tanh(\cdot)$ tends to provide hard labels as $\operatorname{sgn}(\cdot)$. To make the model more appropriate to the data distribution and utilize the training videos adaptively, we take a_i as an optimization variable. Thus the final fusion function can be formulated as,

$$f(x) = \sum_{i=1}^{t} w_i \tanh \frac{f_i(x) - b_i}{a_i}. \tag{2}$$

As the smoothing parameters a are tightly correlated to the thresholds, we formulate the problem as selecting the most appropriate combination of thresholds b and smoothing parameters a, based on which the optimal weights w are learned. In particular, after we get the confidence scores for the i-th feature, we can uniformly sample s confidence scores as threshold candidates, which are denoted as $b_{i1}, b_{i2}, \ldots, b_{is}$. We also preset r smoothing parameters $a_{i1}, a_{i2}, \ldots, a_{ir}$ for each feature. Then we learn the weights w simultaneously based on (2).

To step further, we define a function $\Psi : \mathbb{R}^{t \times s \times r} \rightarrow \mathbb{R}^t$ as $[\Psi(X)]_i = \sum_{j,k} X_{ijk}$, and introduce an indicator matrix $D \in \{0,1\}^{t \times s \times r}$ with $\sum_{j,k} D_{ijk} = 1$, where $D_{ijk} = 1$ indicates that the j-th threshold b_{ij} and the k-th smoothing parameter a_{ik} are selected for i-th feature's confidence scores. Furthermore, we define a function $g_D : \mathbb{R}^d \rightarrow \mathbb{R}^t$ as:

$$g_D(x) = \Psi(D \odot F(x)), \tag{3}$$

where $F_{ijk}(x) = \tanh \frac{f_i(x) - b_{ij}}{a_{ik}}$ and $\odot$ is the Hadamard product. Denoting the fusion classifier as $f(x) =$

$w^T g_D(x)$, to learn weights for different features, a straightforward way is to minimize the following risk function:

$$\Omega(\|w\|_p) + C \sum_{m=1}^{n} \ell oss(-y_m w^T g_D(x_m)), \qquad (4)$$

where $\Omega(\|w\|_p)$ is the regularizer, $\ell oss(\cdot)$ is a convex loss function, and $C > 0$ is a regularization parameter. Here we use squared hinge loss and $\Omega(\|w\|_p) = \frac{1}{2}\|w\|^2$, then the objective function can be formulated as follows:

$$\begin{aligned} \min_{D \in \Omega} \min_{w,\rho,\xi} \quad & \tfrac{1}{2}\|w\|^2 + \tfrac{C}{2} \sum_{m=1}^{n} \xi_m^2 - \rho \qquad (5) \\ \text{s.t.} \quad & y_m w^T g_D(x_m) \geq \rho - \xi_m, \forall m = 1, \ldots, n \end{aligned}$$

where $\Omega = \{D | D \in \{0,1\}^{t \times s \times r}, \sum_{j,k} D_{ijk} = 1\}$ is the feasible set of indicator matrix D. Denoting Lagrange multipliers λ_m for the inequality constraint of inner optimization problem as a vector λ, where $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_n]^T$, (5) can be solved by its dual:

$$\min_{D \in \Omega} \max_{\lambda \in \Lambda} -\frac{1}{2} \sum_{m=1}^{n} \sum_{q=1}^{n} \lambda_m \lambda_q y_m y_q k_D(x_m, x_q) - \frac{1}{2C} \lambda^T \lambda, \quad (6)$$

where $k_D(x_m, x_q) = [g_D(x_m)]^T g_D(x_q)$, and $\Lambda = \{\lambda | \lambda_m \geq 0, \sum_{m=1}^{n} \lambda_m = 1\}$ is the domain of the vector λ.

Noting that (6) is a Mixed Integer Program (MIP), in which λ has exponential size. Following [11, 20], we relax (6) to a convex optimization problem. Next, we show that (6) is lower-bounded by

$$\min_{\mu \in \mathcal{M}} \max_{\lambda \in \Lambda} -\frac{1}{2} (\lambda \odot y)^T \left(\sum_{p: D^p \in \Omega} \mu_p K^p + \frac{1}{C} I \right) (\lambda \odot y) \qquad (7)$$

where $K^p_{mq} = k_{D^p}(x_m, x_q)$, $\mathcal{M} = \{\mu | \sum_p \mu_p = 1, \mu_p \geq 0\}$, and $\Lambda = \{\lambda | \sum_m \lambda_m = 1, \lambda_m \geq 0\}$.

According to the minimax inequality stated in [7], problem (6) is lower-bounded by interchanging $\max_{\lambda \in \Lambda}$ and $\min_{D \in \Omega}$, as

$$\max_{\lambda \in \Lambda} \min_{D \in \Omega} -\frac{1}{2} \sum_{m=1}^{n} \sum_{q=1}^{n} \lambda_m \lambda_q y_m y_q k_D(x_m, x_q) - \frac{1}{2C} \lambda^T \lambda. \qquad (8)$$

By introducing a variable θ, this can be simplified as follows,

$$\max_{\lambda \in \Lambda, \theta} -\theta : \theta \geq -S(\lambda, D^p), \forall D^p \in \Omega, \qquad (9)$$

where $S(\lambda, D^p) = -\frac{1}{2} \sum_{m=1}^{n} \sum_{q=1}^{n} \lambda_m \lambda_q y_m y_q k_{D^p}(x_m, x_q)$. By setting the derivative of the Lagrangian of (9) w.r.t. θ to zero, we have the condition for Lagrange multipliers $\mu_p \geq 0$ of inequality constraint in (9) as $\sum_p \mu_p = 1$. Let $\mu = [\mu_1, \mu_2, \ldots, \mu_P]^T$ be the vector for μ_p, (9) can be further rewritten as

$$\max_{\lambda \in \Lambda} \min_{\mu \in \mathcal{M}} \sum_{D^p \in \Omega} \mu_p S(\lambda, D^p) \qquad (10)$$

where $\mathcal{M} = \{\mu | \mu_p \geq 0, \sum_p \mu_p = 1\}$ is the domain of Lagrange multipliers vector μ. Substituting $S(\lambda, D^p)$, and noting that the objective function is concave w.r.t. λ and convex w.r.t. μ, we can get the objective function as follows.

$$\min_{\mu \in \mathcal{M}} \max_{\lambda \in \Lambda} -\frac{1}{2} (\lambda \odot y)^T \left(\sum_{p: D^p \in \Omega} \mu_p K^p + \frac{1}{C} I \right) (\lambda \odot y) \qquad (11)$$

where $K^p_{mq} = k_{D^p}(x_m, x_q)$, $\mathcal{M} = \{\mu | \sum_p \mu_p = 1, \mu_p \geq 0\}$, and $\Lambda = \{\lambda | \sum_m \lambda_m = 1, \lambda_m \geq 0\}$, then we can see that (11) is equivalent to (7).

3.2. Cutting Plane Algorithm for Optimization

In (7), $\sum_{p: D^p \in \Omega} \mu_p K^p$ can be learned from the convex combination of $|\Omega|$ base matrices. Each base matrix K^p is generated from the indicator matrix D^p, which selects the threshold-smoothing parameter pairs from the preset candidates. We use the cutting plane algorithm [6] to solve this problem efficiently. Our approach generates a pool of threshold-smoothing parameter candidates iteratively with the cutting plane algorithm, which makes the number of base matrices in each iteration much smaller than the original problem. Thus, we can solve the sub-problem in each iteration efficiently.

The detailed steps to solve problem (7) are described as follows. Denoting the current active set as $\mathcal{C} \subset \Omega$, we first initialize the Lagrange multiplier vector λ to be $\frac{1}{n}\mathbf{1}$, where $\mathbf{1}$ indicates a vector of n ones, and find the most violated indicator matrix $\hat{D} \in \Omega$. In the first iteration, we let the initial active set be $\mathcal{C} = \{\hat{D}\}$, then transform problem (7) into its primal form and get a new solution of λ. We continue to find the most violated $\hat{D}$ and add it into the active set $\mathcal{C}$. We repeat finding Lagrangian multipliers vector λ and the most violated indicator matrix $\hat{D}$ until it converges.

Assuming that in the P-th iteration of the cutting plane algorithm, the current active set $\mathcal{C} = \{D^1, D^2, \ldots, D^P\}$, and the problem in (7) corresponds to the following primal optimization problem:

$$\begin{aligned} \min_{\mu \in \mathcal{M}, \hat{w}, \rho, \xi} \quad & \tfrac{1}{2} \sum_{p=1}^{P} \frac{1}{\mu_p} \|\hat{w}_p\|^2 + \frac{C}{2} \sum_{m=1}^{n} \xi_m^2 - \rho \qquad (12) \\ \text{s.t.} \quad & \sum_{p=1}^{P} \hat{w}_p^T g^T_{D^p}(x_m) \geq \rho - \xi_m, \forall m = 1, \ldots, n, \end{aligned}$$

which can be solved following [16] as: 1) fix μ and solve the dual of SVM to update λ, 2) fix λ, use the reduced gradient method to update μ. The complete illustration of the method to solve problem (7) is shown in Algorithm 1. *The complexity of our algorithm is the same as Liblinear [3], which is very efficient for large-scale data.*

Algorithm 1: Feature Weighting via Optimal Thresholding

```
Initialize $\lambda = \frac{1}{n}\mathbf{1}$, find most violated $\hat{D}$, let $\mathcal{C} = \{\hat{D}\}$;
repeat
    Initialize $\mu = [1]^T$, $P \leftarrow 1$;
    repeat
        Fix $\mu$, solve the dual of SVM as follows to update $\lambda$
        $\max_{\lambda \in \Lambda} -\frac{1}{2}(\lambda \odot y)^T \left( \sum_{p=1}^{P} \mu_p K^p + \frac{1}{C} I \right) (\lambda \odot y)$;
        Fix $\lambda$, use the reduced gradient method to update $\mu$;
        $P \leftarrow P + 1$ ;
    until convergence;
    Find the most violated indicator matrix $\hat{D}$ and make $\mathcal{C} = \mathcal{C} \cup \{\hat{D}\}$ ;
until convergence;
```

3.3. Finding the Most Violated Indicator Matrix $\hat{D}$

After updating λ and μ in each iteration, we need to solve the following optimization problem to find the most violated $\hat{D}$,

$$\begin{aligned} & \max_{D \in \Omega} \sum_{m=1}^{n} \sum_{q=1}^{n} \lambda_m \lambda_q y_m y_q k_D(x_m, x_q) \\ \Rightarrow & \max_{D \in \Omega} \sum_{m=1}^{n} \sum_{q=1}^{n} \lambda_m \lambda_q y_m y_q [\Psi(D \odot F(x_m))]^T \Psi(D \odot F(x_q)) \\ \Rightarrow & \max_{D \in \Omega} \sum_{m=1}^{n} \sum_{q=1}^{n} \Psi \left[(\lambda_m y_m F(x_m)) \odot (\lambda_q y_q F(x_q)) \odot D \right] \end{aligned} \tag{13}$$

Defining a matrix $\Theta = \sum_{m=1}^{n} \sum_{q=1}^{n} \lambda_m \lambda_q y_m y_q (F_m \odot F_q)$, we can get the global optimal solution of (13) by setting D_{ijk} to 1 if Θ_{ijk} is the element with the largest value in the i-th row of Θ. Otherwise, we set D_{ijk} to 0.

4. Experiments

We test our approach on three publicly available datasets: HMDB action dataset [8], Columbia Consumer Video (CCV) dataset [5] and TRECVID MED 2011 dataset [1] (including DEV-T and DEV-O collections). In the experiments, we use the same pipeline as described in [24] to evaluate the performance of the proposed method on action recognition, video classification and event detection. In CCV dataset, we use all the acoustic and visual features provided by the authors in [5]. In MED datasets, we generate the BoWs representation as follows. For visual features, *e.g.*, MoSIFT [2], STIP [10], Dense Trajectories [23] and SIFT [12], we use the same setting as we did in [13, 26] to generate the 32,768 dimensional BoWs. In addition to visual features, we use 4,096 dimensional MFCC BoWs [26, 15, 19] as the acoustic feature in the event detection experiment.

In the classification process, we adopt LIBSVM to generate the confidence scores from the probability outputs, and χ^2-kernel is applied to each type of features. We calculate the χ^2-kernel for each feature as described in [24]. Except for the confidence scores from basic features, we also use the predictive scores on average of kernel matrices to enhance the performance.

We compare the result with state-of-the-art fusion algorithms, including Early Kernel Fusion (EKF) [18], Multiple Kernel Learning (MKL) [16], and LPBoost [4]. Other late fusion method like linear SVM on top of normalized decision scores from all the different features has similar optimization goal and consistent performance with the LPBoost. Thus in the late fusion comparison algorithms, we only report the result of LPBoost. In the multi-class classification task (HMDB), we use LP-β [4], a variant of LPBoost, which is designed particularly for feature combination problem in multi-class classification. In TRECVID MED DEV-T and DEV-O collections, we additionally compare the result with Adaptive Late Fusion (ALF) [15], which is particularly designed for event detection.

In the stage of presetting threshold-smoothing parameter candidates, we sample every 10 confidence scores as threshold candidates and empirically set smoothing parameter candidates as {0.5, 0.6, . . . , 0.9}. All the parameters in our proposed method and compared algorithms are selected from $\{10^{-4}, 10^{-2}, \ldots, 10^{4}\}$ according to cross-validation except the parameter v in LPBoost and LP-β, which is chosen from {0.5, 0.6,. . . , 0.9} as suggested by [4].

4.1. Experiment on HMDB dataset

HMDB [8] is a large action recognition dataset, which has been recently collected by Kuehne *et al.* There are 6,766 videos in total from 51 distinct action categories in HMDB. Each category contains at least 101 clips. It is claimed in [8] that it is the largest and perhaps the most realistic available dataset for human action recognition. The huge diversity in visible body parts, camera motion, camera viewpoint, number of people in the action and video quality makes it a very difficult benchmark dataset for the state-of-the-art ac-

Method	Mean Accuracy(%)
Dense Trajectories [23]	46.6
EKF [18]	46.8
MKL [16]	46.9
LPBoost [4]	47.2
FWOT	**48.9**

Table 3. Recognition accuracies on the HMDB dataset [8]. The top row shows the performance of the best individual feature, and others indicate performance of fusion methods.

tion recognition algorithms. The recognition accuracy baseline given in [8] is only 20.44% for the HOG/HOF system and 22.83% for the C2 system.

In our experiment, we use the official three standard training/testing splits identified by [8], which contain 70 videos for training and 30 videos for testing in each action. We use four features as basic features, namely MoSIFT, STIP, Dense Trajectories and SIFT. Before the fusion stage, we train a multi-class SVM classifier for each visual feature with one-vs-all approach. Confidence scores for training videos are obtained by 5-fold cross-validation. After weighted fusion, we choose the action category with highest confidence score as the predicted result. Results are shown in Table 3, in which we list the performance of the best individual feature Dense Trajectories to show the improvement of the fusion methods over the individual feature. Comparison in Table 3 shows that for action recognition in unconstrained videos using the HMDB dataset, our proposed method outperforms the state-of-the-art fusion methods by appropriately assigning optimal weights to multiple features.

4.2. Experiment on Columbia Consumer Video dataset

For the video classification task , we use Columbia Consumer Video dataset (CCV) [5] to compare the performance of different fusion methods. In the CCV dataset, there are totally 9,317 videos with 20 semantic categories, in which 4,659 videos are used as training data and 4,658 videos are used as testing data. The semantic categories contain events like "baseball" and "parade", scenes like "beach", and objects like "cat". Consumer videos contain very diverse content and have much fewer textual tags and descriptions, which motivates the content analysis based on both acoustic and visual features. Since the authors have not provided the original videos of the dataset, we use the three features provided by [5]: STIP features with 5,000 dimensional BoWs representation, SIFT features extracted every two seconds with 5,000 dimensional BoWs representation, and MFCC features with 4,000 dimensional BoWs representation.

Similarly to the experiment on HMDB, we use χ^2-kernel to train non-linear SVMs and use 5-fold cross-validation to get the decision values for the training data. Mean Average Precision is used as evaluation metric as in [5]. In Table 4, we report the experiment results of different fusion methods, and the performance of the best individual feature SIFT is reported as well. Since in CCV dataset, the semantic concept is more complex than the simple action in HMDB dataset, fusing the scene information (*e.g.* SIFT) and acoustic information (*e.g.* MFCC) improves the performance of classification. We can see from the table that our proposed method could discriminate features in different situation, and achieve significant improvement over other fusion methods.

Method	Mean AP (%)
SIFT [12]	52.8
EKF [18]	52.9
MKL [16]	57.1
LPBoost [4]	56.8
FWOT	**60.3**

Table 4. Mean AP on the CCV dataset [5]. The top row shows the performance of the best individual feature, and others indicate performance of fusion methods.

Method	Mean AP	Mean Pmiss
Dense Trajectories [23]	0.354	0.399
EKF [18]	0.414	0.358
MKL [16]	0.412	0.357
LPBoost [4]	0.415	0.365
ALF [15]	0.437	0.346
FWOT	**0.442**	**0.338**

Table 5. Comparison of Mean Average Precision (AP) and Mean Pmiss@TER=12.5 (Pmiss) of different methods on MED 2011 DEV-T collection [1]. **LOWER** Mean Pmiss indicates **BETTER** performance. Top row shows the performance of the best individual feature.

4.3. Experiment on TRECVID MED 2011 dataset

Multimedia Event Detection (MED) [1] is a part of the TRECVID tasks. MED raises a question in communities of multimedia and computer vision: given some descriptions of an event and a set of illustrative video exemplars, could a system detect the occurrence of an event using acoustic and visual information (individually or together)? In 2011, NIST collected a dataset which consists of about 32,000 testing videos from various Internet video hosting sites, namely the DEV-O collection. Then a dataset which consists of about 9,700 training videos, namely DEV-T collection, is used as development dataset for the participants in TRECVID 2011. Detailed information about DEV-T and DEV-O collections can be referred to [1]. *MED 11 DEV-O collection*: 10 events are used in the DEV-O collection to test the performance of multimedia event detection system. These events include "Birthday party (BP)", "Changing a vehicle tire (CaVT)", "Flash mob gathering (FMG)", "Getting a vehicle unstuck (GaVU)", "Grooming an animal (GaA)", "Making a sandwich (MaS)", "Parade (PR)", "Parkour (PK)", "Repairing an appliance (RaA)", and "Working on a sewing project (WaSP)". For each event, 111 to 173 video exemplars are provided. The total duration of the DEV-O collection is about 1,200 hours, which makes it possibly the largest available dataset with meaningful labels for video analysis.

MED 11 DEV-T collection: In DEV-T collection, there are totally 18 events. In addition to the 10 events in DEV-O, there are another 8 events in the DEV-T collection, including "Attempting a board trick (AaBT)", "Feeding an animal (FaA)", "Landing a fish (LaF)", "Wedding ceremony (WC)", "Working on a woodworking project (WoaWP)", "Making a cake (MaC)", "Batting a run (BaR)", and "Assembling a shelter (AaS)".

Different from the recognition datasets, many videos in the MED 2011 DEV-T and DEV-O collections do not belong to any events, which are called null data. The videos in DEV-T and DEV-O collections have huge variance in terms of quality, duration, scene and so forth [1], which makes the MED a great challenge for content based video analysis.

In our experiment, all of the positive video exemplars for each event are used in the training data. In DEV-T collection, we use all the null videos as negative exemplars. In DEV-O collection, we sample 1,000 videos, which do not belong to any event, as negative exemplars. As for video representation, we use Dense Trajectories, STIP, TCH [21] and MFCC as the basic features. When detecting one event, we train a binary χ^2-kernel SVM classifier for each feature to obtain the confidence scores. 5-fold cross-validation is used to get the confidence scores for training data. In the evaluation of DEV-T and DEV-O collections, we use two evaluation metrics. One is the Average Precision (AP), which is popularly used as the evaluation metric in imbalanced binary classification problems. The other is *Probability of Miss-Detection based on the Detection Threshold 12.5*, which is the standard evaluation metric used by NIST [1] in MED to evaluate the performance of a detection system. We denote the second evaluation metric as *Pmiss@TER=12.5* for short. Different from AP metric, **lower** Pmiss@TER=12.5 indicates **better** performance.

We show the comparison of Mean AP and Mean Pmiss@TER=12.5 of different methods on DEV-T collection and DEV-O collection in Table 5 and Table 6. We additionally compare our algorithm to Adaptive Late Fusion (ALF), which was proposed in [15] particularly for event detection. Our method achieves the best performance in both collections. Note that in the Adaptive Late Fusion (ALF) algorithm, thresholds are set before the fusion process, and bad thresholds would lead to weak performance of ALF method. Different performance in DEV-T and DEV-O collections shows that ALF may suffer from the difficulty of getting a good detection threshold and show unstable performance in the fusion stage. On the contrary, our method learns proper thresholds in the process of weighting fusion, which makes the fusion method more robust in the event detection system. In Figure 3 we show the comparison of Average Precision and Pmiss@TER=12.5 of different fusion methods on every event in TRECVID MED 11 DEV-O collection. We can see that our fusion method outperforms other state-of-the-art fusion algorithms in 8 out of 10 events in TRECVID MED 11 DEV-O collection.

Method	Mean AP	Mean Pmiss
Dense Trajectories [23]	0.240	0.367
EKF [18]	0.310	0.318
MKL [16]	0.310	0.307
LPBoost [4]	0.322	0.310
ALF [15]	0.210	0.359
FWOT	**0.336**	**0.294**

Table 6. Comparison of Mean Average Precision (AP) and Mean Pmiss@TER=12.5 (Pmiss) of different methods on MED 2011 DEV-O collection [1]. **LOWER** Mean Pmiss indicates **BETTER** performance. Top row shows the performance of the best individual feature.

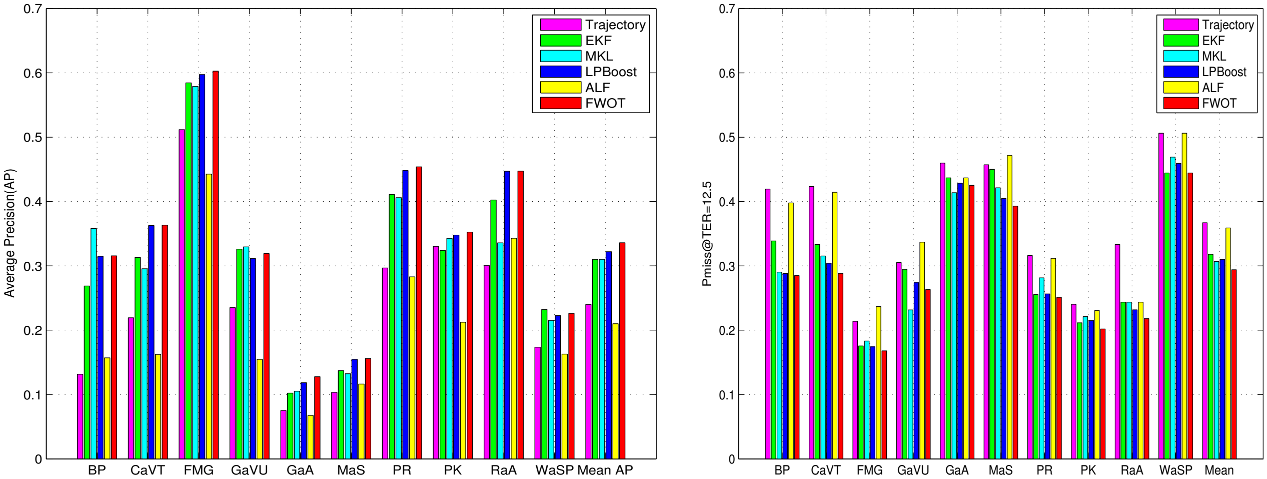
5. Conclusion

In this paper, we have introduced an approach to leverage multiple features by decision-level fusion, which optimizes the weights and thresholds for features in the confidence scores simultaneously. We formulate the problem as selecting the most appropriate combination of thresholds and smoothing parameters, based on which the optimal weights are learned. We first preset lots of thresholds and smoothing parameter candidates, then we use the cutting plane algorithm to obtain the optimal weights and thresholds, which is very efficient even in a large-scale problem. Experiments on HMDB dataset and CCV dataset show that our approach outperforms other state-of-the-art methods on action recognition and consumer video classification. In addition, we achieve the best performance among different fusion methods on a large-scale video dataset TRECVID MED 2011 (including DEV-T and DEV-O collections) using both Average Precision and Pmiss@TER=12.5 metrics. The experimental results confirm that our method is superior to other fusion methods for different video analysis tasks.

6. Acknowledgements

All the features were extracted on Blacklight at Pittsburgh Supercomputing Center (PSC). We would like to thank PSC for providing the computing resource.

This work was partially supported by 973 Program (No. 2012CB316400), partially supported by Singapore A*



(a) Average Precision
HIGHER indicates **BETTER** performance

(b) Pmiss@TER=12.5
LOWER indicates **BETTER** performance

Figure 3. Comparison of AP and Pmiss@TER=12.5 on TRECVID MED 2011 DEV-O dataset [1]

under Grant SERC 112 172 0013 and partially supported by 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 notwithstanding 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.

References

[1] `http://www.nist.gov/itl/iad/mig/med11.cfm`.

[2] M. Chen and A. Hauptmann. Mosift: Recognizing human actions in surveillance videos. In *CMU-CS-09-161*, 2009.

[3] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin. Liblinear: A library for large linear classification. *JMLR*, 9:1871–1874, 2008.

[4] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *CVPR*, 2009.

[5] Y.-G. Jiang, G. Ye, S.-F. Chang, D. Ellis, and A. C. Loui. Consumer video understanding: A benchmark database and an evaluation of human and machine performance. In *ICMR*, 2011.

[6] J. Kelley Jr. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial & Applied Mathematics*, 8(4):703–712, 1960.

[7] S. Kim and S. Boyd. A minimax theorem with applications to machine learning, signal processing, and finance. *SIAM Journal on Optimization*, 19(3):1344–1367, 2008.

[8] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011.

[9] Z. Lan, L. Bao, S.-I. Yu, W. Liu, and A. G. Hauptmann. Double fusion for multimedia event detection. In *Advances in Multimedia Modeling*. 2012.

[10] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[11] Y. Li, I. Tsang, J. Kwok, and Z. Zhou. Tighter and convex maximum margin clustering. In *AISTATS*, 2009.

[12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[13] Z. Ma, Y. Yang, Z. Xu, S. Yan, N. Sebe, and A. G. Hauptmann. Complex event detection via multi-source video attributes. In *CVPR 2013*.

[14] P. Natarajan, S. Wu, S. Vitaladevuni, X. Zhuang, U. Park, R. Prasad, and N. P. Multi-channel shape-flow kernel descriptors for robust video event detection and retrieval. *ECCV*, 2012.

[15] P. Natarajan, S. Wu, S. Vitaladevuni, X. Zhuang, S. Tsakalidis, U. Park, and R. Prasad. Multimodal feature fusion for robust event detection in web videos. In *CVPR*, 2012.

[16] A. Rakotomamonjy, F. Bach, S. Canu, Y. Grandvalet, et al. SimpleMKL. *JMLR*, 9:2491–2521, 2008.

[17] K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. *MVAP*, 2012.

[18] C. G. Snoek, M. Worring, and A. W. Smeulders. Early versus late fusion in semantic video analysis. In *ACM Multimedia*. ACM, 2005.

[19] A. Tamrakar, S. Ali, Q. Yu, J. Liu, O. Javed, A. Divakaran, H. Cheng, and H. Sawhney. Evaluation of low-level features and their combinations for complex event detection in open source videos. In *CVPR*, 2012.

[20] M. Tan, L. Wang, and I. W. Tsang. Learning sparse svm for feature selection on very high dimensional datasets. In *ICML 2010*.

[21] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *TPAMI*, 32(9):1582–1596, 2010.

[22] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *CVPR*, 2009.

[23] H. Wang, A. Klaser, C. Schmid, and C. Liu. Action recognition by dense trajectories. In *CVPR*, 2011.

[24] H. Wang, M. Ullah, A. Klaser, I. Laptev, C. Schmid, et al. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009.

[25] Y. Yang, J. Song, Z. Huang, Z. Ma, N. Sebe, and A. Hauptmann. Multi-feature fusion via hierarchical regression for multimedia analysis. *TMM*, 15(3):572–581, 2013.

[26] S.-I. Yu, Z. Xu, D. Ding, W. Sze, F. Vicente, Z. Lan, Y. Cai, et al. Informedia e-lamp@ trecvid2012: Multimedia event detection and recounting med and mer. In *NIST TRECVID Workshop*, 2012.