Multiple Features But Few Labels? A Symbiotic Solution Exemplified for Video Analysis

Zhigang Ma
Carnegie Mellon University
kevinma@cs.cmu.edu

Yi Yang
The University of Queensland
yi.yang@uq.edu.au

Nicu Sebe
University of Trento
sebe@disi.unitn.it

Alexander G. Hauptmann
Carnegie Mellon University
alex@cs.cmu.edu

ABSTRACT

Video analysis has been attracting increasing research due to the proliferation of internet videos. In this paper, we investigate how to improve the performance on internet quality video analysis. Particularly, we work on the scenario of few labeled training videos being provided, which is less focused in multimedia. To being with, we consider how to more effectively harness the evidences from the low-level features. Researchers have developed several promising features to represent videos to capture the semantic information. However, as videos usually characterize rich semantic contents, the analysis performance by using one single feature is potentially limited. Simply combining multiple features through early fusion or late fusion to incorporate more informative cues is doable but not optimal due to the heterogeneity and different predicting capability of these features. For better exploitation of multiple features, we propose to mine the importance of different features and cast it into the learning of the classification model. Our method is based on multiple graphs from different features and uses the Riemannian metric to evaluate the feature importance. On the other hand, to be able to use limited labeled training videos for a respectable accuracy we formulate our method in a semi-supervised way. The main contribution of this paper is a novel scheme of evaluating the feature importance that is further casted into a unified framework of harnessing multiple weighted features with limited labeled training videos. We perform extensive experiments on video action recognition and multimedia event recognition and the comparison to other state-of-the-art multi-feature learning algorithms has validated the efficacy of our framework.

Categories and Subject Descriptors

H.3.1 [**Information Search and Retrieval**]: Content Analysis and Indexing; I.2.10 [**Vision and Scene Understanding**]: Video analysis

General Terms

Algorithms, Experimentation, Performance

MM'14, November 3–7, 2014, Orlando, Florida, USA.

http://dx.doi.org/10.1145/2647868.2654907.

Keywords

Video Analysis; Multi-feature; Weighted Features; Riemannian Distance; Semi-supervised Learning

1. INTRODUCTION

The sheer amount of videos and their expansion demand effective content analysis for indexing, retrieval and management. Video analysis is essentially a task to understand the semantics, for which it has to bridge the semantic gap between the low-level features and the high-level semantic content description [7].

Encouraging progress on video analysis has been made in recent years. At early times, video analysis was focused on controlled video analysis such as news videos. The attention has been gradually shifted to unconstrained video analysis such as videos recorded by amateurs. Current trend is to analyze internet quality videos due to the exponential growth of videos online. Typical internet quality video analysis includes action recognition and multimedia event analysis, which are studied in this paper. Video action recognition aims to understand the human motions contained in videos, *e.g.*, *running*. More dynamic information is included and it is usually beneficial to analyze several video sequences. Multimedia event analysis is much more complicated than action recognition as a multimedia event contains several concepts, actions within one or various scenes. For example, the event *making a cake* consists of a combination of several concepts such as *cake*, *people*, *kitchen* together with the action *making* within a longer video sequence. To attain reasonable accuracy, we usually need the entire video clip for extracting discriminative cues. In spite of various approaches for video analysis, extracting robust features and building classifiers based on them are the essential thrust. On one hand, a variety of features have been proposed in the literature for video analysis. For example, researchers have proposed capturing spatial-temporal volumes to address the local variations in both space and time. Since both spatial and temporal information are leveraged, this kind of feature has more reliable performance for video semantic analysis. Among others, STIP feature is mostly used [14]. Besides the STIP feature, another recent feature type, *i.e.*, MoSIFT shows promising performance in video semantic analysis [2]. Dense trajectory [33] is gaining increasing attention as it has been shown to be very robust for video analysis. Apart from visual features, some other modalities, which provide different yet complementary information, can also be used to represent videos. For example, auditory features based on Mel-frequency Cepstral Coefficients (MFCC) have also been frequently used to represent videos [13]. Focusing on different characteristics of the videos, these features intuitively should complement each other. That said, it is highly

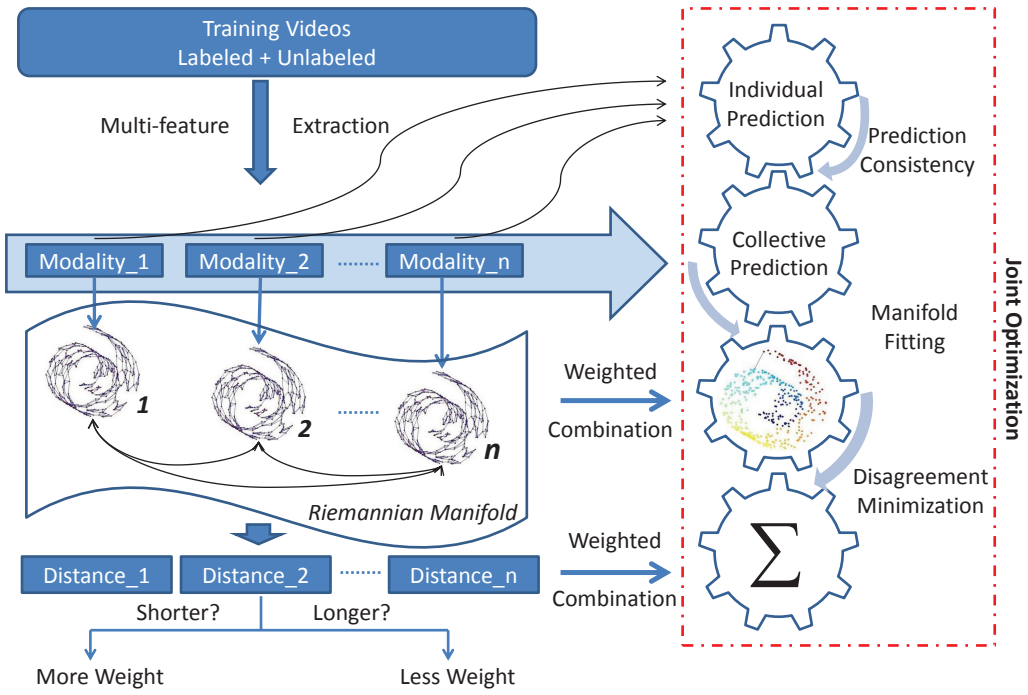

Figure 1: The illustration of our multi-feature learning framework.

possible to further boost the analysis accuracy by utilizing different features together. A simple way is to concatenate feature vectors of different types as the final representation. However, each feature has its own statistical property and simple concatenation is potentially limited in mining the informative cues from different features.

On the other hand, the performance of video analysis is also affected by the classifiers. Traditional classifiers such as Support Vector Machine (SVM) [33] usually require a large amount of accurately labeled training data. But in practice, we usually do not have a large number of precisely labeled videos as the training data and manual labeling is tedious and time-consuming. Using a small number of training data for classifier learning, however, is prone to over-fitting.

This paper aims to boost internet quality video analysis by simultaneously addressing the two aforementioned issues, *i.e.*, how to effectively integrate different features to better utilize the evidences from each feature, and how to guarantee a reasonable performance when only few labeled training data are provided. We propose a unified framework which deals with the two issues simultaneously, thus resulting in improved performance for video analysis.

Previous work has shown that multi-feature learning is a promising way to analyze different features simultaneously. Among others, Canonical Correlation Analysis [8][32], two-view support vector machines, namely SVM-2K and their variants [11][15] are the representative algorithms. Some other algorithms in the literature include [4][1][36][40], *etc.* Nonetheless, the aforementioned work mostly treats each feature equally while in fact for a specific data set usually one type of feature has higher recognition capability than other counterparts. Having been a common issue for other applications such as image annotation, differentiating feature capability in classification was investigated in multi-feature learning. Since we have a similar assumption that different features possibly have different contributions to the video analysis performance *w.r.t.* different datasets, we propose to mine the inconsistency of observations from different features and subsequently, their importance adaptively in our framework.

The second issue to be tackled in this paper is grounded on semi-supervised learning. Semi-supervised learning relieves the burden of manual labeling by properly using unlabeled data together with labeled data. A review of the progress on semi-supervised learning is [43] and constructing graphs from both labeled and unlabeled data to utilize manifold structure is the mainstream approach of semi-supervised learning. We opt to build a normalized Laplacian graph of each feature type to utilize the corresponding unlabeled data. Another usage of these graphs is for mining the inconsistency of different features raised by the first issue we mentioned before. In this paper, we assume that the inconsistency can be evaluated by distances between different graphs. If a graph constructed from one feature type is comparatively farther away from those constructed from other features, this feature is possibly inconsistent with other features from the classification perspective. Our learning scheme considers this inconsistency and balances between different features to attain a more robust classifier. As for the distance calculation, the Riemannian metric is exploited to measure that between different graphs. The underlying reason is that the normalized Laplacian graphs are Symmetric Positive Definite (SPD) matrices. Previous work has suggested that the Riemannian metric is a good measure of dissimilarity between two SPD matrices as SPD matrices exist on the Riemannian manifold [9].

Figure 1 illustrates how our method learns feature importance and the related classifiers jointly. After extracting multiple features (denoted as Modality 1 to Modality n), multiple graphs are constructed. We then calculate the Riemannian distance of each graph *w.r.t.* other graphs. It is reasonable that some features have better prediction capability but less noise than other features. Hence, these stronger features tend to better uncover the manifold structure of the data and it would be beneficial to give them more weights in the learning process. Based on the assumption about the quality

of features, that they have roughly similar performance, with some noise on the results, we propose to learn such importance by distance measuring following some recent work [35]. Shorter distance indicates the related feature has more agreement with other features, thus also meaning the feature is more important. Larger distance indicates the related feature is an outlier *w.r.t.* other features, thus making it less important. The importance is reflected by the weights which are used to combine the graphs and the distances. In the following joint optimization module, there are three constraints to be satisfied. First, the predictions from each single modality should be consistent with the prediction from the combined modalities. Second, the prediction should fit the manifold structure uncovered by the combined graphs. Third, the weights should balance the Riemannian distances to minimize the disagreement from different modalities. For example, if the distance is larger (meaning the feature might have worse performance in prediction), the weight should be smaller. Through the joint optimization process, the weights and the classifiers for predictions are properly learned. In the following step, they can be applied to the testing videos to obtain decision values and weighted fusion.

In summary, the main contributions of this paper are as follows:

- A scheme is developed to evaluate the importance of each feature type by leveraging Riemannian distance atop multimedia manifold structure.
- The effective utilization of weighted features is embedded into a multi-feature semi-supervised learning framework.
- The proposed approach is desirable for internet quality video analysis when only limited labeled training data are available.
- The method is general and can be used in other applications when multiple features are provided with limited labeled training data.

We name our method **R**iemannian weighted **S**emi-**S**upervised **M**ulti-feature learning (RSSM).

2. RELATED WORK

In this section, we briefly review the related work. Some typical work on video analysis is first reviewed. Then we give some discussion on semi-supervised learning and multi-feature learning.

2.1 Video Analysis

Existing work on video analysis is focused on different levels. Video concept analysis aims to understand the abstract or general idea inferred from specific instances and has been widely studied in recent years [18][28][27][31]. Video action recognition, by contrast, makes more effort on analyzing the human motions contained in videos. Hence, how to capture the motion information has been the major focus. A representative work in earlier time is the Space-time Interest Points proposed by Laptev and Lindeberg [14]. By using differential descriptors to detect the local structures in the spatial-temporal domain, they generate features of the videos and show their efficacy for action recognition. Chen *et al.* introduce a novel descriptor, namely MoSIFT for action recognition [2]. MoSIFT is built atop the SIFT descriptor by integrating local motion information. Wang *et al.* show that describing videos by dense trajectories boosts the action recognition accuracy remarkably [33]. Positioned in a higher level with more difficulties, multimedia event analysis is gradually attracting more research interest. An event builds upon many concepts and is unlikely to be inferred with a single image or a few video frames. Some representative work includes [10][20][23][22]. Generally speaking, the research on multimedia event analysis is still based on robust low-level features. Visual features such as STIP and dense trajectories have proved to be effective. Meanwhile, other modalities such as audio features are also helpful in event analysis. This is different from video action recognition which is more dependent on visual features only.

Robust features are essential for video analysis. Nonetheless, each single feature has its advantage but is pragmatically insufficient to capture all the variations of different actions or events. Concerned with this limitation, some work has been proposed to fuse the evidences from multiple features for better multimedia analysis [29][16][37]. Yet they are supervised methods which require a good amount of labeled training data. By contrast, we aim to get reasonable performance with only few labeled training data.

2.2 Semi-supervised Learning

To handle the paucity of labeled training data, semi-supervised learning is a good way as it simultaneously exploits labeled and unlabeled training data [43]. Within the area of video analysis, semi-supervised learning has been widely used for concept analysis [38][42][6]. By contrast, limited work has investigated its usage on more complicated tasks such as action recognition and event analysis [19][34][41]. In [19], Ma *et al.* have proposed a semi-supervised feature analyzing framework integrating $l_{2,1}$-norm regularized feature selection and manifold learning and have applied it to 3D action recognition. Wang *et al.* cast shared structural subspace learning into a semi-supervised learning framework for action recognition and demonstrate its advantage over supervised classifiers [34]. In [41], Yu *et al.* have proposed a semi-supervised algorithm for tracking in the surveillance videos. Though showing the potential of semi-supervised learning, the aforementioned work does not take into account how to reasonably fuse evidences from different features in a statistical way. Hence, in this paper we focus on video action recognition and multimedia event analysis with a newly proposed semi-supervised model that integrates multi-feature learning.

2.3 Multi-feature Learning

Canonical Correlation Analysis (CCA) [8][32] is a classical multi-feature learning approach that maximizes the correlations between two features. Subsequent representative work includes two-view support vector machines (SVM-2K) [11] and multiple kernel learning (MKL) [24] that both build upon SVM. However, CCA and SVM-2K can only analyze two features simultaneously while MKL induces a heavy computational burden due to the computation of multiple kernels. Gehler *et al.* propose learning the correct weighting of different features based on LPBoost for object recognition [5]. Xu *et al.* propose a method to learn the weights, thresholding and smoothing parameters jointly and apply it to action recognition and multimedia event detection [37]. The algorithms mentioned above all work in a supervised fashion. Yang *et al.* have proposed a regression model which integrates semi-supervised learning and multi-feature learning [39]. The model optimizes the classifier inferred from each feature but does not consider the fact that different features have different prediction capability. Additionally, it is a linear algorithm but nonlinear algorithms are more robust for video analysis. Though we can apply some mapping method such as KPCA to preprocess the data to make their method comparable to a nonlinear approach, it is a two-step fashion and the two processes are not tightly coupled.

We develop a nonlinear algorithm that aims to progress beyond existing semi-supervised multi-feature learning algorithms by uncovering the feature importance and using such information for classifier learning. In a nutshell, we formulate our multi-feature learning based on uncovering feature disagreement motivated by [3] which argues that different features may have different capabilities for a classification task. By analyzing the feature disagreement we may exert more influence from the more powerful features on learning the classifier. Particularly, we couple it with the semi-supervised learning process. Normalized Laplacian graphs are constructed for semi-supervised learning and we analyze the distances of one graph from other graphs. These distances serve for the evaluation scheme of the feature disagreement and are fed into the framework. Here a thought should be given to how we choose the distance metric. Normalized Laplacian graphs are Symmetric Positive Definite (SPD) matrices lying on a Riemannian manifold. Though Euclean distance can be also used to approximate the distance between two SPD matrices, previous work has suggested that the Riemannian metric is a more precise measurement [9]. Hence, we choose the Riemannian distance in measuring the distances between graphs to achieve the uncovering of the feature dissimilarity.

3. PROBLEM FORMULATION

Suppose we have n training data represented by m different features and denote them as $X_v|_{v=1}^m \in \mathbb{R}^{d_v \times n}$. d_v is the feature dimension. $X_v = [x_1^v, x_2^v, ..., x_n^v]$ among which l data are labeled. Let $Y \in \{0,1\}^{n \times c}$ be the label matrix. c stands for the class number and $y_i \in \mathbb{R}^c (1 \le i \le n)$ is the label vector with c classes. Let Y_{ij} denote the j-th datum of y_i then $Y_{ij} = 1$ if x_i^v is in the j-th class, while $Y_{ij} = 0$ otherwise. If x_i^v is not labeled, y_i is set to a vector with all zeros, namely, $\forall i > l,\ y_i|_{i=(l+1)}^n = 0^{c \times 1}$.

We begin formulating our algorithm with the classical regularized empirical error that results in a classifier f based on a set of training data $\{x_i, y_i\}_{i=1}^n$ where y_i indicates the label of x_i:

$$\min_f \sum_{i=1}^n loss\left(f(x_i), y_i\right) + \gamma \Omega(f). \tag{1}$$

$loss(\cdot,\cdot)$ is a loss function and $\Omega(f)$ is the regularization function on f with γ as its parameter.

Considering the ease of implementation, we use the following linear transformation:

$$f(x_i) = W^T x_i + b_i, \tag{2}$$

where W is the projection matrix used for classification and b_i is the bias. Using the matrix form and considering each feature type, Eq. (1) can be rewritten as:

$$\min_{W_v, b_v} \sum_{v=1}^m \left\| X_v^T W_v + 1_n b_v^T - F_v \right\|_F^2 + \gamma \sum_{v=1}^m \|W_v\|_F^2 \tag{3}$$

where $1_n \in \mathbb{R}^n$ denotes a column vector with all ones; the second item is added to avoid over-fitting. Note that we replace Y with a predicted label matrix F_v because: 1) many of the training data are unlabeled and F_v can reflect the possible labels of them; 2) it is useful in integrating the predictions from different features as will shown later.

The second step is to construct the normalized Laplacian graph L_i from each feature type. Following the traditional way, we formulate $\tilde{L}_i = D_i - G_i$ where D_i is a diagonal matrix with its diagonal elements as $\sum_{j=1}^n G_{ij}$ and G_i is a weight matrix whose element G_i^{pq} reflects the similarity between x_p and x_q. Next, $L_i = D_i^{-1/2} \tilde{L}_i D_i^{-1/2}$ which is an SPD matrix. Based on L_i, we calculate the Riemannian distance of each L_i to other graphs. For two graphs L_i and L_j, their distance is calculated as $\left\| \log(L_i^{-1/2} L_j L_i^{-1/2}) \right\|_F$. We therefore obtain the specific values for each graph *w.r.t.* to all the other graphs. We sum the values and denote it as R_i for each graph.

Next, we show how to associate the observations from each feature. Since we decide the feature importance by using the Riemannian distance based on the graphs, we assign a weight λ_i to R_i and L_i when performing multi-feature learning. Meanwhile, the feature vectors from all the features are concatenated to form a data matrix $X = [X_1; \cdots; X_v] \in \mathbb{R}^{d \times n}$ and we propose to similarly learn a predicted label matrix F from X. F is supposed to satisfy three conditions: 1) consistent with the known ground truth labels of the training data [44][43]; 2) smooth on the combined manifold of the weighted graphs [44][43]; 3) consistent with the predicted labels from each single feature. By incorporating these constraints, we get the following objective function:

$$\begin{aligned} &\min_{F,F_v,W,W_v,\lambda_v,b,b_v} \left\| X^T W + 1_n b^T - F \right\|_F^2 + \sum_{v=1}^m \left\| X_v^T W_v + 1_n b_v^T - F_v \right\|_F^2 \\ &+ \sum_{v=1}^m \|F_v - F\|_F^2 + Tr\left(F^T (\sum_{v=1}^m \lambda_v L_v) F \right) + Tr\left((F-Y)^T U (F-Y) \right) \\ &+ \alpha \sum_{v=1}^m \lambda_v R_v + \mu \sum_{v=1}^m \lambda_v \log \lambda_v + \gamma (\sum_{v=1}^m \|W_v\|_F^2 + \|W\|_F^2) \\ &s.t. \quad \sum_{v=1}^m \lambda_v = 1, \lambda_v \in [0,1]. \end{aligned} \tag{4}$$

In Eq. (4), the maximum entropy regularization $\sum_{v=1}^m \lambda_v \log \lambda_v$ is added to avoid the trivial solution that $\lambda_k' = 1$ and $\lambda_k = 0$, $\forall k \ne k'$; U is a diagonal matrix whose diagonal element $U_{ii} = \infty$ if x_i is labeled and $U_{ii} = 1$ otherwise. Note that Eq. (4) mixes several types of constraints involving different objects. All the constraints are based on comparable numerical values so they can be mixed in the same objective function.

As nonlinear classifiers have proved to be advantageous for video analysis, we further extend Eq. (4) by introducing a kernel function k defined by the nonlinear feature mapping, *i.e.*, $k(a,b) = \varphi(a)^T \varphi(b)$ for any two data a and b. Thus, we rewrite Eq. (4) as:

$$\begin{aligned} &\min_{F,F_v,\varphi(W),\varphi(W_v),\lambda_v,b,b_v} \left\| \varphi(X)^T \varphi(W) + 1_n b^T - F \right\|_F^2 \\ &+ \sum_{v=1}^m \left\| \varphi(X_v)^T \varphi(W_v) + 1_n b_v^T - F_v \right\|_F^2 + \sum_{v=1}^m \|F_v - F\|_F^2 \\ &+ Tr\left(F^T (\sum_{v=1}^m \lambda_v L_v) F \right) + Tr\left((F-Y)^T U (F-Y) \right) + \alpha \sum_{v=1}^m \lambda_v R_v \\ &+ \mu \sum_{v=1}^m \lambda_v \log \lambda_v + \gamma \left(\sum_{v=1}^m \|\varphi(W_v)\|_F^2 + \|\varphi(W)\|_F^2 \right) \\ &s.t. \quad \sum_{v=1}^m \lambda_v = 1, \lambda_v \in [0,1]. \end{aligned} \tag{5}$$

4. SOLUTION

Our objective function is non-smooth so we propose an alternating algorithm to solve it.

(1) Fixing λ_v to optimize $\varphi(W_v)$, F_v, b_v, $\varphi(W)$, F and b:

By setting the derivative *w.r.t.* b_v to 0, we have:

$$b_v^T = \frac{1}{n} \left(1_n^T F_v - 1_n^T \varphi(X_v)^T \varphi(W_v) \right). \tag{6}$$

Denote a centering matrix as $H = I_n - \frac{1}{n} 1_n 1_n^T$ where $I_n \in \mathbb{R}^{n \times n}$ is an identity matrix. Substituting Eq. (6) into Eq. (5), the problem sub-

sequently converts to the following one when optimizing $\varphi(W_v)$:

$$\begin{aligned}
&\min_{\phi(W_v)} \sum_{v=1}^{m} Tr\left((H\varphi(X_v)^T\varphi(W_v) - HF_v)^T(H\varphi(X_v)^T\varphi(W_v) - HF_v)\right) \\
&+\gamma \sum_{v=1}^{m} Tr\left(\varphi(W_v)^T\varphi(W_v)\right) \\
&\Rightarrow \min_{\varphi(W_v)} \sum_{v=1}^{m} Tr\left(\varphi(W_v)^T\varphi(X_v)H\varphi(X_v)^T\varphi(W_v) - 2\varphi(W_v)^T\varphi(X_v)HF_v\right) \\
&+\gamma \sum_{v=1}^{m} Tr\left(\varphi(W_v)^T\varphi(W_v)\right) \\
&\Rightarrow \min_{\varphi(W_v)} \sum_{v=1}^{m} Tr\left(\varphi(W_v)^T[\varphi(X_v)H\varphi(X_v)^T + \gamma I_v]\varphi(W_v) - 2\varphi(W_v)^T\varphi(X_v)HF_v\right)
\end{aligned} \tag{7}$$

where $I_v \in \mathbb{R}^{d_v \times d_v}$ is an identity matrix. By setting the derivative *w.r.t* $\varphi(W_v)$ to 0, we have:

$$\begin{aligned}
&2\varphi(X_v)H\varphi(X_v)^T\varphi(W_v) - 2\varphi(X_v)HF_v + 2\gamma\varphi(W_v) = 0 \\
&\Rightarrow \varphi(W_v) = \left(\varphi(X_v)H\varphi(X_v)^T + \gamma I_v\right)^{-1}\varphi(X_v)HF_v
\end{aligned} \tag{8}$$

Let $A_v = \left(\varphi(X_v)H\varphi(X_v)^T + \gamma I_v\right)^{-1}$ and substitute $\varphi(W_v)$ and b_v into the objective function, the optimization of F_v equals to:

$$\begin{aligned}
&\min_{F_v} \sum_{v=1}^{m} \left\|H\phi(X_v)^T A_v\phi(X_v)HF_v - HF_v\right\|_F^2 + \sum_{v=1}^{m} \|F_v - F\|_F^2 \\
&+\gamma \sum_{v=1}^{m} \|A_v\phi(X_v)HF_v\|_F^2 \\
&\Rightarrow \min_{F_v} \sum_{v=1}^{m} Tr\left(F_v^T(H\phi(X_v)^T A_v\phi(X_v)H - I_n)^2F_v\right) + \sum_{v=1}^{m} Tr(F_v^T - F^T)(F_v - F) \\
&+\gamma \sum_{v=1}^{m} Tr\left(F_v^T H\phi(X_v)^T A_v^2\phi(X_v)HF_v\right) \\
&\Rightarrow \min_{F_v} \sum_{v=1}^{m} Tr\left(F_v^T[(H\varphi(X_v)^T A_v\varphi(X_v)H - I_n)^2 + I_n + \gamma H\varphi(X_v)^T A_v^2\varphi(X_v)H]F_v\right) \\
&-2\sum_{v=1}^{m} Tr(F_v^T F)
\end{aligned} \tag{9}$$

By setting the derivative *w.r.t.* F_v to 0, we have:

$$\begin{aligned}
&\left((H\phi(X_v)^T A_v\phi(X_v)H - I_n)^2 + I_n + \gamma H\phi(X_v)^T A_v^2\phi(X_v)H\right)F_v = F \\
&\Rightarrow F_v = J_v F
\end{aligned} \tag{10}$$

where $J_v = \left(I_n - H\varphi(X_v)^T A_v\varphi(X_v)H\right)^{-1}$. In the same manner, we can get:

$$b^T = \frac{1}{n}\left(1_n^T F - 1_n^T\varphi(X)^T\varphi(W)\right), \tag{11}$$

$$\varphi(W) = \left(\varphi(X)H\varphi(X)^T + \gamma I\right)^{-1}\varphi(X)HF \tag{12}$$

where $I \in \mathbb{R}^{d\times d}$ is an identity matrix. Let $M = \left(\varphi(X)H\varphi(X)^T + \gamma I\right)^{-1}$ and substitute $\varphi(W_v)$, b_v, F_v, $\varphi(W)$, b into the objective function, we then arrive at:

$$\begin{aligned}
&\min_{F} \left\|H\varphi(X)^T M\varphi(X)HF - HF\right\|_F^2 \\
&+\sum_{v=1}^{m} \left\|\left(H\varphi(X_v)^T A_v\varphi(X_v)HJ_v - HJ_v\right)F\right\|_F^2 + \sum_{v=1}^{m} \|(J_v - I_n)F\|_F^2 \\
&+Tr\left(F^T(\sum_{v=1}^{m}\lambda_v L_v)F\right) + Tr\left((F-Y)^T U(F-Y)\right) \\
&+\gamma\left(\sum_{v=1}^{m} \|A_v\varphi(X_v)HJ_vF\|_F^2 + \|M\varphi(X)HF\|_F^2\right)
\end{aligned} \tag{13}$$

By setting its derivative *w.r.t.* F to 0, we obtain:

$$F = \left(\begin{array}{l} \sum_{v=1}^{m} J_v(H - \varphi(X_v)^T A_v\varphi(X_v)H)J_v \\ +H - H\varphi(X)^T M\varphi(X)H \\ +\sum_{v=1}^{m}\lambda_v L_v + \sum_{v=1}^{m}(J_v - I_n)^2 + U \end{array}\right)^{-1} UY \tag{14}$$

(2) Fixing $\varphi(W_v)$, F_v, b_v, $\varphi(W)$, F and b to optimize λ_v:

The problem is equivalent to:

$$\begin{aligned}
&\min_{\lambda_v} Tr\left(F^T(\sum_{v=1}^{m}\lambda_v L_v)F\right) + \mu\sum_{v=1}^{m}\lambda_v\log\lambda_v + \alpha\sum_{v=1}^{m}\lambda_v R_v \\
&s.t. \quad \sum_{v=1}^{m}\lambda_v = 1, \lambda_v \in [0,1]
\end{aligned} \tag{15}$$

By using a Lagrange multiplier ξ we convert the above problem to a Lagrange function as:

$$\begin{aligned}
L(\lambda_v, \xi) = &Tr(F^T\sum_{v=1}^{m}\lambda_v L_v F) + \mu\sum_{v=1}^{m}\lambda_v\log\lambda_v \\
&+\alpha\sum_{v=1}^{m}\lambda_v R_v - \xi(\sum_{v=1}^{m}\lambda_v - 1)
\end{aligned} \tag{16}$$

By setting its derivative *w.r.t.* λ_v and ξ to 0 respectively, we have:

$$\left\{\begin{array}{l} Tr(F^T L_v F) + \mu\log\lambda_v + \mu + \alpha R_v - \xi = 0 \\ \sum_{v=1}^{m}\lambda_v - 1 = 0 \end{array}\right. \tag{17}$$

We thus obtain:

$$\lambda_v = \frac{\exp\left(\frac{-Tr(F^T L_v F) - \alpha R_v - \mu}{\mu}\right)}{\sum_{v=1}^{m}\exp\left(\frac{-Tr(F^T L_v F) - \alpha R_v - \mu}{\mu}\right)} \tag{18}$$

Note that for any matrix Q, $Q(Q^T Q + \mu I)^{-1} = (QQ^T + \mu I)Q$. Hence, we can rewrite $\varphi(W_v)$, J_v, $\varphi(W)$ and F as:

$$\varphi(W_v) = \varphi(X_v)H\left(H\varphi(X_v)^T\varphi(X_v)H + \gamma I_n\right)^{-1}F_v, \tag{19}$$

$$J_v = \left(I_n - H\varphi(X_v)^T\varphi(X_v)H[H\varphi(X_v)^T\varphi(X_v)H + \gamma I_n]^{-1}\right)^{-1}, \tag{20}$$

$$\varphi(W) = \varphi(X)H\left(H\varphi(X)^T\varphi(X)H + \gamma I_n\right)^{-1}F, \tag{21}$$

$$\begin{aligned}
F = \Bigg(&\sum_{v=1}^{m} J_v H J_v - \sum_{v=1}^{m} J_v\varphi(X_v)^T\varphi(X_v)H[H\varphi(X_v)^T\varphi(X_v)H + \gamma I_n]^{-1}J_v \\
&+H - H\varphi(X)^T\varphi(X)H\left(H\varphi(X)^T\varphi(X)H + \gamma I_n\right)^{-1} + \sum_{v=1}^{m}\lambda_v L_v \\
&+\sum_{v=1}^{m}(J_v - I_n)^2 + U\Bigg)^{-1} UY
\end{aligned} \tag{22}$$

Let $K_{tr}^v \in \mathbb{R}^{n\times n}$ and $K_{tr} \in \mathbb{R}^{n\times n}$ denote the kernel matrices calculated from X_v and X. Let $K_{te}^v \in \mathbb{R}^{n_{te}\times n}$ and $K_{te} \in \mathbb{R}^{n_{te}\times n}$ denote the kernel matrices between the testing data and the training data based on the v-th feature and the concatenated feature respectively. n_{te} indicates the number of testing data. We can therefore arrive at Algorithm 1.

Discussion on the computational complexity:

Computing the graph Laplacian is $O(n^2)$ and computing the kernel matrix from the training data is $O(d_v n^2)$. During the training process, computing the inverse of a few matrices in training has the complexity of $O(n^3)$. During the testing process, computing

the kernel matrix between the testing data and the training data is $O(d_v n n_{te})$.

Algorithm 1: Optimization procedure for RSSM.

Input:
K_{tr}^v, K_{tr}, K_{te}^v, K_{te}, L_v, U, R_v and Y;
Parameters α, μ and γ.

Output:
Optimized classification model.

1: Set $t=0$ and initialize $\lambda_v = \frac{1}{m}$;
2: **repeat**
Update J_v as $J_v = \left(I_n - HK_{tr}^v H(HK_{tr}^v H + \gamma I_n)^{-1}\right)^{-1}$;
Update F as
$F = [\sum_{v=1}^{m} J_v H J_v - \sum_{v=1}^{m} J_v K_{tr}^v H(HK_{tr}^v H + \gamma I_n)^{-1} J_v + H - HK_{tr}H(HK_{tr}H + \gamma I_n)^{-1} + \sum_{v=1}^{m} \lambda_v L_v + \sum_{v=1}^{m} (J_v - I_n)^2 + U]^{-1} UY$;
Update F_v as $F_v = J_v F$;
Update λ_v according to Eq. (18);
Update the objective function value obj;
$t = t+1$.
until *Convergence:* $|obj_{t+1} - obj_t| / obj_t \leq 10^{-3}$;
3: $Model = H(HK_{tr}^v H + \gamma I_n)^{-1} F_v$.

5. EXPERIMENTS

We evaluate our algorithm on two tasks of video analysis: video action recognition and multimedia event recognition. The comparison to several state-of-the-art algorithms is first presented with discussion. Additionally, we have some experimental study on the weights learned by our method, the contributions of different features and different BoW representations.

5.1 Datasets

We use four datasets in the experiments. For action recognition, Youtube action dataset [17], UCF50 action dataset [25] and HMDB dataset [12] are used. Youtube dataset has 1600 video sequences from 11 actions. UCF50 consists of 6681 video sequences from 50 actions. HMDB is made up of 6849 video clips from 51 actions. For multimedia event recognition, we use the TRECVID MED 2013 MEDTEST data. In this dataset, there are positive videos of 20 events and the remaining videos are null videos. As our experiment is to evaluate the performance for recognition, we only use all the positive videos, thus resulting in a subset of 3489 videos. We name this subset Multimedia Event Dataset (MED) for convenience.

Following [30], we extract three dynamic visual features, *i.e.*, STIP [14], MoSIFT [2] and dense trajectories (Trajectory) [33] for all the four datasets. We additionally extract MFCC feature from MED dataset as the audio information has been shown to be helpful for multimedia event analysis. Based on each feature type, we generate the 4096 dimensional bag-of-words representation for the videos.

5.2 Comparison Algorithms

The following is a brief introduction of the comparison algorithms in our experiments.

- RSSM: The proposed method in this paper. We apply the χ^2 kernel due to its robustness for bag-of-words features [33].
- SVM: We use SVM with χ^2 kernel as it has demonstrated to be advantageous for video analysis [33]. As both early fusion and late fusion are typical for using multiple features, we implement SVM with both approaches and denote them as SVM_{ef} and SVM_{lf}.
- SVM-2K [11]: A representative multi-feature learning algorithm.
- SimpleMKL [24]: An algorithm that utilizes multiple features through multiple kernel learning.
- LP-β [5]: A variant of LPBoost which is designed particularly for feature combination problem in multi-class classification.
- Feature Weighting via Optimal Thresholding (FWOT) [37]: A recent feature fusion method that learns the weights, thresholding and smoothing parameters jointly.
- Multi-feature Learning via Hierarchical Regression (MLHR) [39]: A regression model which integrates semi-supervised learning and multi-feature learning.

To transform SVM-2K and MLHR to nonlinear approaches so that they are comparable to other methods, we perform full rank principal component analysis [26] with the χ^2 kernel to map the low-level features into a Hilbert space $\mathcal{H}$. The representations in $\mathcal{H}$ are the input for these two methods.

5.3 Setup

Similarly to [34], we randomly split Youtube and UCF50 to multiple (5 in our experiments) training and testing sets in a half-and-half fashion. For HMDB, we use the three provided partitions to form the training and testing sets. All the results reported on these three datasets are the average results from these splits. For MED data, NIST already provided the standard development (training) set and evaluation (testing) set so we use the provided partition. Mean Average Precision (MAP) is used as the evaluation metric. For the training sets, 2%, 10%, 25%, 50% and 75% training data are labeled respectively and the supervised algorithms only use the labeled data for training.

The parameters involved are set as follows. The parameter for kernel calculation is fixed to the mean of the pairwise distances among the training samples, which is a widely adopted strategy in the literature; the regularization parameters are all tuned from $\{0.001, 0.1, 10, 1000\}$. For SVM-2K and SimpleMKL, we use the code shared by the authors and follow their parameter settings. We report the best results for each algorithm.

5.4 Recognition Results

The recognition results on different datasets are shown from Table 1 to Table 4. It can be seen that our method is consistently better than other comparison algorithms with different percentage of training data labeled on all the four datasets. Our method is significantly better judged by the t-test (with a significance level of 0.001). Specifically, we have the following observations and discussion: 1) Semi-supervised algorithms, *i.e.*, RSSM and MLHR, perform better than supervised algorithms probably due to the usage of unlabeled training data; 2) The advantage of RSSM and MLHR over other supervised algorithms is especially visible when only few training video are labeled, *e.g.*, 2%, 10% or 25% of the training videos. 3) RSSM outperforms MLHR, which indicates that properly mining feature importance can give us extra benefit; 4) When increasing the number of labeled training data, the recognition accuracy of all comparison algorithms improves, which is

Table 1: Action recognition results on Youtube dataset (MAP±Standard Deviation). The best results are highlighted in bold.

Comparison Algorithms	2%	10%	25%	50%	75%
RSSM	**0.275±0.021**	**0.498±0.014**	**0.689±0.021**	**0.798±0.016**	**0.864±0.016**
SVM_{ef}	0.115±0.010	0.265±0.032	0.532±0.014	0.657±0.015	0.731±0.014
SVM_{lf}	0.117±0.007	0.271±0.006	0.537±0.008	0.667±0.010	0.740±0.005
SimpleMKL	0.216±0.030	0.364±0.009	0.528±0.017	0.627±0.021	0.673±0.020
LP-β	0.120±0.014	0.285±0.030	0.564±0.038	0.660±0.033	0.732±0.018
FWOT	0.137±0.013	0.291±0.009	0.589±0.011	0.681±0.008	0.769±0.011
MLHR	0.260±0.011	0.483±0.012	0.663±0.010	0.772±0.015	0.824±0.016

Table 2: Action recognition results on UCF50 dataset (MAP±Standard Deviation). The best results are highlighted in bold.

Comparison Algorithms	2%	10%	25%	50%	75%
RSSM	**0.292±0.019**	**0.577±0.016**	**0.796±0.011**	**0.871±0.005**	**0.920±0.016**
SVM_{ef}	0.140±0.001	0.334±0.014	0.673±0.013	0.802±0.011	0.867±0.011
SVM_{lf}	0.141±0.001	0.377±0.014	0.703±0.007	0.826±0.008	0.885±0.006
SimpleMKL	0.177±0.021	0.387±0.017	0.544±0.010	0.650±0.004	0.703±0.005
LP-β	0.126±0.003	0.388±0.015	0.650±0.007	0.826±0.008	0.888±0.006
FWOT	0.162±0.001	0.394±0.014	0.713±0.010	0.841±0.010	0.894±0.009
MLHR	0.283±0.019	0.557±0.014	0.771±0.008	0.869±0.006	0.913±0.005

Table 3: Action recognition results on HMDB dataset (MAP±Standard Deviation). The best results are highlighted in bold.

Comparison Algorithms	2%	10%	25%	50%	75%
RSSM	**0.095±0.003**	**0.197±0.013**	**0.265±0.002**	**0.334±0.001**	**0.382±0.004**
SVM_{ef}	0.058±0.001	0.101±0.013	0.225±0.015	0.280±0.006	0.319±0.008
SVM_{lf}	0.058±0.001	0.115±0.012	0.232±0.013	0.285±0.008	0.324±0.011
SimpleMKL	0.053±0.005	0.104±0.013	0.145±0.002	0.190±0.008	0.212±0.008
LP-β	0.058±0.001	0.135±0.010	0.173±0.010	0.257±0.008	0.309±0.006
FWOT	0.063±0.001	0.133±0.010	0.239±0.016	0.299±0.007	0.338±0.010
MLHR	0.071±0.001	0.182±0.016	0.256±0.003	0.317±0.002	0.357±0.006

Table 4: Multimedia event recognition results on MED dataset (MAP). The best results are highlighted in bold.

Comparison Algorithms	2%	10%	25%	50%	75%
RSSM	**0.229**	**0.414**	**0.538**	**0.601**	**0.662**
SVM_{ef}	0.115	0.265	0.396	0.503	0.551
SVM_{lf}	0.117	0.271	0.401	0.510	0.554
SimpleMKL	0.186	0.271	0.345	0.397	0.438
LP-β	0.156	0.234	0.404	0.521	0.569
FWOT	0.137	0.291	0.410	0.533	0.577
MLHR	0.211	0.401	0.514	0.585	0.630

Table 5: Riemmanian distances and the learned weights on Youtube dataset *w.r.t.* different percentage of training data labeled.

Modality	Riemmanian Distance	2%	10%	25%	50%	75%
STIP	56.7962	0.3363	0.3423	0.3418	0.3430	0.3440
MoSIFT	57.1440	0.3356	0.3414	0.3412	0.3365	0.3334
Trajectory	57.4910	0.3282	0.3162	0.3170	0.3205	0.3227

consistent with our experience that more labeled training data always help; 5) The advantage of semi-supervised methods over supervised methods tends to plateau as more training data are labeled, which is especially visible on UCF50 dataset; 6) FWOT generally obtains more robust performance in comparison with other supervised multi-feature learning methods such as SimpleMKL and LP-β. The better performance of RSSM validates the effectiveness of our approach that simultaneously leverages unlabeled data and feature importance.

5.5 Riemmanian Distances v.s. Weights

Our method leverages the Riemmannian distances between multiple graphs constructed from multiple modalities for learning the weights of the corresponding modalities. In this subsection, we

Table 6: Riemmanian distances and the learned weights on UCF50 dataset *w.r.t.* different percentage of training data labeled.

Modality	Riemmanian Distance	2%	10%	25%	50%	75%
STIP	37.6956	0.3331	0.3333	0.3418	0.3334	0.3347
MoSIFT	38.2276	0.3309	0.3323	0.3412	0.3331	0.3302
Trajectory	35.1621	0.3360	0.3344	0.3170	0.3335	0.3351

Table 7: Riemmanian distances and the learned weights on HMDB dataset *w.r.t.* different percentage of training data labeled.

Modality	Riemmanian Distance	2%	10%	25%	50%	75%
STIP	36.7737	0.3334	0.3334	0.3335	0.3336	0.3336
MoSIFT	39.3522	0.3333	0.3333	0.3328	0.3327	0.3325
Trajectory	36.6340	0.3334	0.3334	0.3337	0.3337	0.3339

Table 8: Riemmanian distances and the learned weights on MED dataset *w.r.t.* different percentage of training data labeled.

Modality	Riemmanian Distance	2%	10%	25%	50%	75%
STIP	83.5899	0.2513	0.2500	0.2501	0.2526	0.2560
MoSIFT	83.7743	0.2506	0.2500	0.2501	0.2505	0.2487
Trajectory	83.5189	0.2518	0.2506	0.2502	0.2546	0.2580
MFCC	85.7730	0.2459	0.2463	0.2494	0.2422	0.2373

give the weights learned by our algorithm in pair of the distances from Table 5 to Table 8. It can be seen that different features have varying importance on a particular dataset. On Youtube dataset, the graph constructed from STIP feature is closer to those constructed from MoSIFT and Trajectory features. The resulting weights learned from our model reflect the distances with more weight given to STIP feature. On UCF50 dataset, the graph constructed from Trajectory feature is closer to those constructed from STIP and MoSIFT features. Though the difference between the weights learned from our model is not so obvious as on Youtube dataset, we can still see that more weights are given to Trajectory feature. On HMDB dataset, we similarly observe that the graph constructed from Trajectory feature is closer to other graphs and it obtains subtly more weight from our model. On MED dataset, Trajectory feature similarly gets more weights in comparison with other three features. MFCC feature is given least weight by our model, which is consistent with previous experience on TRECVID MED that visual features are still more discriminative than audio features. The slight difference of the learned weights on UCF50, HMDB and MED datasets may indicate that these three datasets are more difficult to analyze than Youtube dataset. Our model is still able to learn proper weights of different features, thus resulting in a reasonable way to fuse the heterogeneous evidences. On the other hand, in our algorithm the weights and the corresponding classifiers for different features are learned jointly. The integration of both contributes to the performance gain. In spite of the slight difference of the weights, the classifiers are still optimized under the influence of the learned weights.

We also observe that the relative importance of different feature does not change a lot over different datasets. This result may be because that for all the datasets we have the same feature extraction pipeline. Besides, the literature has also shown that STIP and Trajectory are better for video analysis, especially action recognition and event analysis. It is worth pointing out that all the other comparison algorithms have used the same features as our method but our method still outperforms other methods, which again indicates the efficacy of our method.

5.6 Performance *w.r.t.* Different Features

As a classical multi-feature learning algorithm, SVM-2K can only handle two features. To compare our method to this classical algorithm, we conduct experiments on Youtube dataset by leaving one feature out. We therefore have three combinations, *i.e.*, STIP+MoSIFT, STIP+Trajectory and MoSIFT+Trajectory. In this way, we can also study the predicting capability of different features. We would point out that using only two features actually restrains the performance of RSSM since the feature disagreement loses its power ($R_1 = R_2$). However, we can see from the comparison in Figure 2 that our method still consistently outperforms SVM-2K no matter which two features are used. The advantage is possibly attributed to the usage of unlabeled training data.

5.7 BoW v.s. Spatial BoW

The spatial BoW representation introduced in [21] has shown good performance for video analysis. In this subsection, we show the performance gain by using spatial BoW compared to standard BoW representation on Youtube dataset. We use 1x1, 2x2 and 3x1 spatial grids to generate the spatial BoW representation. For each grid, we use the standard BoW representation with 4,096 dimensions, thus resulting in 32,768 dimension spatial BoW feature for each feature type. Note that we did not use the spatial BoW representation in the above experiments due to the consideration of computational efficiency. The comparison is presented in Figure 3. We observe from the figure that using spatial BoW representation can further boost the accuracy in comparison with using standard BoW representation.

6. CONCLUSION

We have studied how to improve the performance of internet quality video analysis, particularly on action recognition and multimedia event recognition. Considering that: 1) Discriminative features are the basis of video analysis; 2) A variety of good features are available with different capability of capturing the information from videos; and 3) Traditional classifiers work with many labeled training data which are costly to obtain in practice, we tackled the

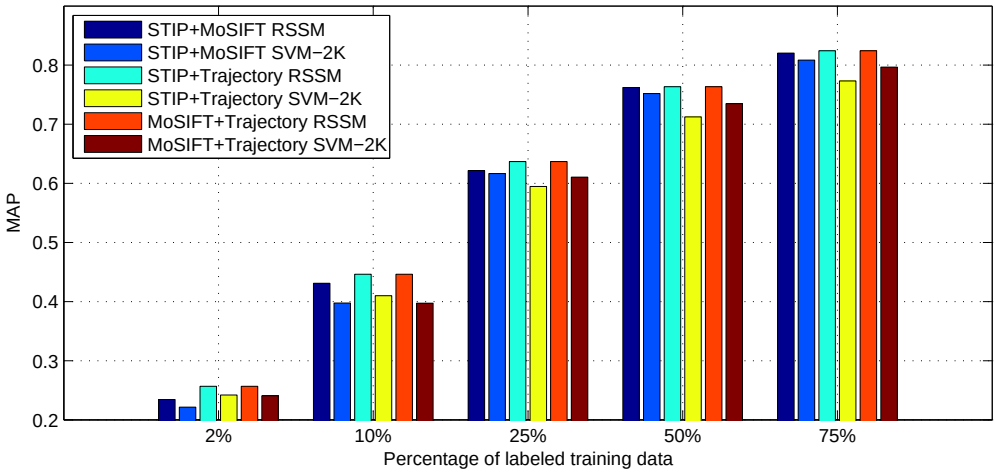

Figure 2: Action recognition results using different combinations of two features on Youtube dataset. Both the results of RSSM and SVM-2K are given in the figure. RSSM is consistently better than SVM-2K.

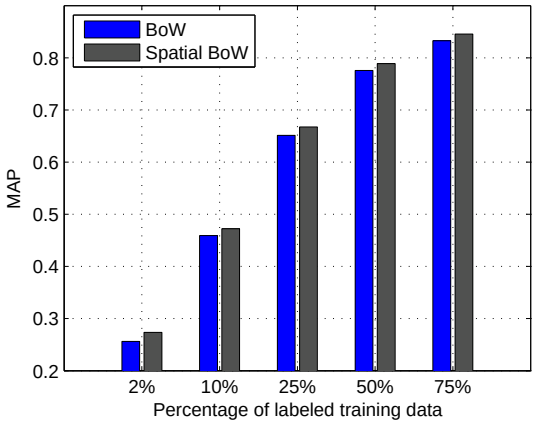

Figure 3: Performance comparison between using BoW representation and using spatial BoW representation on Youtube dataset. It can be seen that using spatial BoW representation results in better performance than using BoW representation.

problem by proposing a joint learning framework that integrates semi-supervised learning, multi-feature learning and the Riemannian metric. Specifically, the main merit of our method is its ability of effectively exploiting multiple features with the consideration of the different predicting capability of these features. On top of that, our method can attain reasonable performance when only few training videos are labeled, which makes our method desirable in real-world scenarios. We have tested our method on video action recognition and multimedia event recognition using four widely used datasets. Our method is advantageous compared to several other state-of-the-art algorithms. The experimental results suggest that unlabeled data, multiple features and feature importance as priors all contribute to the boosted recognition accuracy. Lastly, it is worth mentioning that our framework can be also used in other applications when we have multiple features but few labeled training data, which is another nice property.

7. ACKNOWLEDGMENTS

This paper was partially supported by the European Commission project xLiMe, the Open Project Program of the State Key Lab of CAD&CG (Grant No. A1402), Zhejiang University, the US Department of Defense the U. S. Army Research Office (W911NF-13-1-0277) and 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 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, ARO, or the U.S. Government.

8. REFERENCES

[1] R. K. Ando and T. Zhang. Two-view feature generation model for semi-supervised learning. In *ICML*, pages 25–32, 2007.

[2] M.-Y. Chen and A. G. Hauptmann. Mosift: Recognizing human actions in surveillance videos. In *Technical Report CMU-CS-09-161, Carnegie Mellon University*, 2009.

[3] C. M. Christoudias, R. Urtasun, and T. Darrell. Multi-view learning in the presence of view disagreement. In *UAI*, 2008.

[4] M. A. Davenport, C. Hegde, M. F. Duarte, and R. G. Baraniuk. Joint manifolds for data fusion. *IEEE Transactions on Image Processing*, 19(10):2580–2594, 2010.

[5] P. V. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *ICCV*, pages 221–228, 2009.

[6] Y. Han, Y. Yang, Y. Yan, Z. Ma, N. Sebe, and X. Zhou. Semi-supervised feature selection via spline regression for video semantic recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 2014.

[7] A. G. Hauptmann, R. Yan, W.-H. Lin, M. G. Christel, and H. D. Wactlar. Can high-level concepts fill the semantic gap in video retrieval? a case study with broadcast news. *IEEE Transactions on Multimedia*, 9(5):958–966, 2007.

[8] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3).

[9] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. T. Harandi. Kernel methods on the riemannian manifold of symmetric positive definite matrices. In *CVPR*, pages 73–80, 2013.

[10] L. Jiang, A. G. Hauptmann, and G. Xiang. Leveraging high-level and low-level features for multimedia event detection. In *ACM Multimedia*, pages 449–458, 2012.

[11] T.-K. Kim, J. Kittler, and R. Cipolla. Discriminative learning and recognition of image set classes using canonical correlations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):1005–1018, 2007.

[12] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: A large video database for human motion recognition. In *ICCV*, pages 2556–2563, 2011.

[13] Z.-Z. Lan, L. Bao, S.-I. Yu, W. Liu, and A. G. Hauptmann. Multimedia classification and event detection using double fusion. *Journal of Multimedia Tools and Applications*, 2013.

[14] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, pages 432–439, 2003.

[15] G. Li, S. C. H. Hoi, and K. Chang. Two-view transductive support vector machines. In *SDM*, pages 235–244, 2010.

[16] Y. Li, B. Geng, D. Tao, Z.-J. Zha, L. Yang, and C. Xu. Difficulty guided image retrieval using linear multiple feature embedding. *IEEE Transactions on Multimedia*, 14(6):1618–1630, 2012.

[17] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos. In *CVPR*, pages 1996–2003, 2009.

[18] J. Luo, J. Yu, D. Joshi, and W. Hao. Event recognition: viewing the world with a third eye. In *ACM Multimedia*, pages 1071–1080, 2008.

[19] Z. Ma, F. Nie, Y. Yang, J. R. R. Uijlings, N. Sebe, and A. G. Hauptmann. Discriminating joint feature analysis for multimedia data understanding. *IEEE Transactions on Multimedia*, 14(6):1662–1672, 2012.

[20] Z. Ma, Y. Yang, Y. Cai, N. Sebe, and A. G. Hauptmann. Knowledge adaptation for ad hoc multimedia event detection with few exemplars. In *ACM Multimedia*, pages 469–478, 2012.

[21] M. Marszalek, C. Schmid, H. Harzallah, and J. van de Weijer. Learning object representations for visual object class recognition. In *Visual recognition challenge workshop*, 2007.

[22] M. Mazloom, E. Gavves, K. E. A. van de Sande, and C. Snoek. Searching informative concept banks for video event detection. In *ICMR*, pages 255–262, 2013.

[23] M. Merler, B. Huang, L. Xie, G. Hua, and A. Natsev. Semantic model vectors for complex video event recognition. *IEEE Transactions on Multimedia*, 14(1):88–101, 2012.

[24] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

[25] K. K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. *Mach. Vis. Appl.*, 24(5):971–981, 2013.

[26] B. Schölkopf, A. J. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[27] M.-L. Shyu, Z. Xie, M. Chen, and S.-C. Chen. Video semantic event/concept detection using a subspace-based multimedia data mining framework. *IEEE Transactions on Multimedia*, 10(2):252–259, 2008.

[28] C. Snoek, M. Worring, J. van Gemert, J.-M. Geusebroek, and A. W. M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *ACM Multimedia*, pages 421–430, 2006.

[29] J. Song, Y. Yang, Z. Huang, H. T. Shen, and R. Hong. Multiple feature hashing for real-time large scale near-duplicate video retrieval. In *ACM Multimedia*, pages 423–432, 2011.

[30] A. Tamrakar, S. Ali, Q. Yu, J. Liu, O. Javed, A. Divakaran, H. Cheng, and H. S. Sawhney. Evaluation of low-level features and their combinations for complex event detection in open source videos. In *CVPR*, pages 3681–3688, 2012.

[31] V. S. Tseng, J.-H. Su, J.-H. Huang, and C.-J. Chen. Integrated mining of visual features, speech features, and frequent patterns for semantic video annotation. *IEEE Transactions on Multimedia*, 10(2):260–267, 2008.

[32] A. Vinokourov, J. Shawe-Taylor, and N. Cristianini. Inferring a semantic representation of text via cross-language correlation analysis. In *NIPS*, pages 1473–1480, 2002.

[33] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, pages 3169–3176, 2011.

[34] S. Wang, Y. Yang, Z. Ma, X. Li, C. Pang, and A. G. Hauptmann. Action recognition by exploring data distribution and feature correlation. In *CVPR*, pages 1370–1377, 2012.

[35] X. Wang, W. Bian, and D. Tao. Grassmannian regularized structured multi-view embedding for image classification. *IEEE Transactions on Image Processing*, 22(7):2646–2660, 2013.

[36] D. P. Williams. Bayesian data fusion of multiview synthetic aperture sonar imagery for seabed classification. *IEEE Transactions on Image Processing*, 18(6):1239–1254, 2009.

[37] Z. Xu, Y. Yang, I. Tsang, N. Sebe, and A. G. Hauptmann. Feature weighting via optimal thresholding for video analysis. In *ICCV*, pages 3440–3447, 2013.

[38] Y. Yang, F. Nie, D. Xu, J. Luo, Y. Zhuang, and Y. Pan. A multimedia retrieval framework based on semi-supervised ranking and relevance feedback. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(4):723–742, 2012.

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

[40] S. Yu, B. Krishnapuram, R. Rosales, H. Steck, and R. B. Rao. Bayesian co-training. In *NIPS*, 2007.

[41] S.-I. Yu, Y. Yang, and A. G. Hauptmann. Harry potter's marauder's map: Localizing and tracking multiple persons-of-interest by nonnegative discretization. In *CVPR*, pages 3714–3720, 2013.

[42] T. Zhang, C. Xu, G. Zhu, S. Liu, and H. Lu. A generic framework for video annotation via semi-supervised learning. *IEEE Transactions on Multimedia*, 14(4):1206–1219, 2012.

[43] X. Zhu. Semi-supervised learning literature survey. In *Technical Report 1530, University of Wisconsin, Madison*, 2007.

[44] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003.