Exploring Inter-feature and Inter-class Relationships with Deep Neural Networks for Video Classification

Zuxuan Wu[1], Yu-Gang Jiang[1], Jun Wang[2], Jian Pu[1], Xiangyang Xue[1]
[1]School of Computer Science, Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai, China
[2]Institute of Data Science and Technology, Alibaba Group, Seattle, WA, USA
{zxwu, ygj, jianpu, xyxue}@fudan.edu.cn, wongjun@gmail.com

ABSTRACT

Videos contain very rich semantics and are intrinsically multimodal. In this paper, we study the challenging task of classifying videos according to their high-level semantics such as human actions or complex events. Although extensive efforts have been paid to study this problem, most existing works combined multiple features using simple fusion strategies and neglected the exploration of inter-class semantic relationships. In this paper, we propose a novel unified framework that jointly learns feature relationships and exploits the class relationships for improved video classification performance. Specifically, these two types of relationships are learned and utilized by rigorously imposing regularizations in a deep neural network (DNN). Such a regularized DNN can be efficiently launched using a GPU implementation with an affordable training cost. Through arming the DNN with better capability of exploring both the inter-feature and the inter-class relationships, the proposed regularized DNN is more suitable for identifying video semantics. With extensive experimental evaluations, we demonstrate that the proposed framework exhibits superior performance over several state-of-the-art approaches. On the well-known Hollywood2 and Columbia Consumer Video benchmarks, we obtain to-date the best reported results: 65.7% and 70.6% respectively in terms of mean average precision.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Indexing methods*

Keywords

Multimodal Features; Class Relationships; Deep Neural Networks; Action and Event Recognition

1. INTRODUCTION

Techniques for recognizing high-level semantics in diverse unconstrained videos can be deployed in many applications, such as Web video search and video surveillance systems. However, it is well-known that recognizing or classifying the video semantics is an extremely challenging task due to various factors, such as the semantic gap between low-level video features and the complex semantics. While significant progress has been made in recent years, most state-of-the-art solutions often utilized a large set of features with simple fusion strategies to model high-level video semantics. For instance, two popular ways of combining multiple video features are early fusion and late fusion [41]. Early fusion concatenates all the feature vectors into a long representation for model training and testing, while late fusion trains a model using each feature separately and combines the outputs of all the models. Both methods do not have the capability of explicitly modeling the correlations among the video features, which can be exploited for deriving better representations. In addition, the existing video classification methods often neglected the inter-class relationships among video semantics. Note that such semantic correlations can be exploited to boost the classification performance since knowing the presence of one class may help predict other correlated semantics contained in the same video. Although there exist a few works investigating multi-feature fusion or exploring the inter-class relationships, as will be discussed in the next section, they mostly addressed the two problems separately. Also, many existing methods are computationally expensive; thus, they are less feasible for large scale applications.

Realizing the limitations of the existing works, in this paper, we propose a unified framework based on deep neural network (DNN), which jointly learns feature relationships and class relationships, and simultaneously carries out video classification within the same framework utilizing the learned relationships. Figure 1 gives a conceptual diagram of the proposed approach. First, we extract various video features including local visual descriptors and audio descriptors. The features are then used as the inputs of a DNN, where the first two layers are input layer and feature transformation layer. The third layer of the network is called *fusion layer*, where structural regularization is imposed on the network weights to identify and utilize the feature relationships. Specifically, the regularization terms are designed based on the observation of two natural properties of the inter-feature relationships, *correlation* and *diversity*. The former means that different features may share some common patterns in a middle level representation lying between the original features and the high-level semantics. The latter emphasizes the unique characteristics of different features,

MM'14, November 03 - 07 2014, Orlando, FL, USA

http://dx.doi.org/10.1145/2647868.2654931.

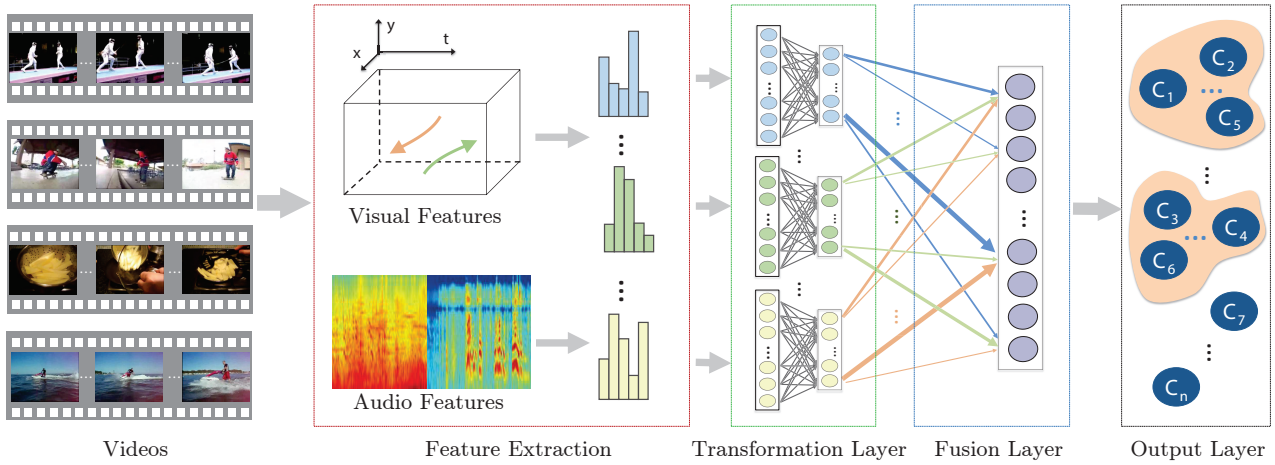


Figure 1: Overview of the proposed DNN-based video classification framework. Various visual/audio features are first extracted and then used as inputs of a DNN. The features are transformed (abstracted) using one layer of neurons before fusion. On the fusion layer, regularization on the network parameters is imposed to ensure that different features can share correlated dimensions while preserving their unique characteristics. As indicated by line width in the figure, some dimensions of different features may be highly correlated (the thick lines pointing to the same neuron). After that, the weights between the fusion and the output layer are also regularized to identify groups of classes. Both the learned inter-feature and inter-class relationships are utilized for improved classification performance.

which serve as complementary information for predicting the video semantics. Through modeling these two properties using a feature correlation matrix, we impose a trace-norm regularization over the fusion weights to reveal the hidden correlation and diversity of the features.

For the inter-class relationships, we impose regularizations on the weights of the final *output layer* to automatically identify the grouping structures of video classes, as well as the outlier classes. Semantic classes within the same group share commonalities or correlations that can be utilized as knowledge sharing for improved classification performance, while the outlier classes should be excluded from negative knowledge sharing. We will show that by imposing a similar trace-norm based regularization on the weights of the output layer, we are able to explore such complex inter-class relationships effectively to produce better video classification results. Therefore, this allows us to develop a unified framework using a regularized DNN, which can be easily implemented using a GPU with affordable training cost.

Notice that it is feasible to use raw video data as inputs instead of the hand-crafted features like the recent works on image classification using deep learning [22]. In this case, the convolutional neural network (CNN) can be adopted for performing feature extraction from the raw data. The reasons of using the hand-crafted features in our proposed framework are two-folds. First, the hand-crafted features are widely used in video classification and remain the central components of some video analytical systems that generated recent state-of-the-art results on tasks like human action recognition [46] and event recognition [1, 23]. By using these features it is easy to make fair comparisons with the traditional semantic classification approaches such as the popular SVM classifiers. Second, extracting features using neural networks needs more layers of neurons that incur a significant number of additional parameters to be tuned, requiring much more training data. Note that in many video classification tasks, the amount of available training data is far less from sufficient for training a neural network with too many layers. Therefore, in this paper, we focus the proposed regularized DNN on the tasks of feature fusion and video semantic classification.

To the best of our knowledge, this work represents the first attempt to capture both the feature and the class relationships in a DNN for video classification. Our major contributions are summarized as follows:

1. We propose to impose structural regularization on the *fusion layer* in a DNN to identify the correlations of multiple features, while still maintaining their diversities. This unique capacity distinguishes the proposed method from most of the existing works that often adopted shallow fusion process without considering the deep exploration of the feature correlations.

2. We also propose to explore inter-class relationships through imposing a similar structure regularization on *output layer* of the DNN. Therefore, both the inter-feature and the inter-class relationships are formulated and explored in a unified framework, which can be easily implemented with a GPU and trained with an affordable time cost.

3. Extensive empirical evaluations are provided to corroborate the effectiveness of the proposed framework in detail, and we attained to-date the highest performance on the widely used real-world benchmarks.

The remaining sections of this paper are organized as follows. Section 2 discusses related works. Section 3 elaborates the proposed framework, including both formulation and optimization. Extensive experimental results and comparisons with alternative methods and the state of the arts are reported and discussed in Section 4. Finally, Section 5 concludes this paper.

2. RELATED WORK

Extensive studies have been conducted in the field of video classification, and typical approaches often combined several multiple features in a standard machine learning pipeline using classifiers like the SVM. Most of the existing work focused on developing effective features [10, 24, 46], novel recognition methods [29, 39], or comprehensive systems that integrate multiple features and classifiers for competitive classification performance [23, 30]. Besides accuracy, efficiency is another important factor that should be considered in the design of a modern video classification system. Several recent studies focused on this issue by using efficient classification methods [27] or parallel computing [49].

In the following we focus our discussion on works investigating multi-feature fusion, and on those exploring inter-class relationships, which are more relevant to this paper.

2.1 Fusing Multiple Features

As aforementioned, there exist two popular fusion strategies, known as early fusion and late fusion. Although both strategies are not able to explore hidden feature relationships such as the correlations between features, they are widely used in many state-of-the-art systems due to the simplicity [1]. In addition, both of them require to design fusion weights that indicate the importance of each individual feature. The weights can be set as equal using heuristics (a.k.a. average fusion), or learned using the cross validation method. Some other works also employed multiple kernel learning (MKL) [5] to estimate the fusion weights [7, 31], which may lead to performance gain that is nevertheless often observed to be insignificant [45].

More recently, several advanced feature fusion approaches have been proposed. In [50], an optimization framework was used for robust late fusion to derive better combination of multiple feature modalities. It seeks a shared low-rank matrix to remove noises of certain modalities, which requires to iteratively compute singular value decomposition with a cubic time complexity, and thus is less scalable for large scale real-world applications. In a following up work by Liu et al. [25], dynamic fusion was adopted to find the best feature combination for each sample. This approach was proved effective but is extremely time-consuming. In [17], Jiang et al. proposed to construct an audio-visual joint codebook based on the discovered correlations between audio and visual features for video concept classification. The approach pointed out a promising direction as this is among the first works performing deep mining of feature correlations. However, the used visual features were computed on each segmented patches from video frames, which is computationally prohibitive for most real-word scenarios. The approach was further enhanced in [18], where the temporal interaction of audio-visual features was investigated. Jhuo et al. [16] improved the speed of training the audio-visual joint codebook by using standard local visual features like the SIFT, instead of the segmentation-based region features.

There are also a few studies on combining multiple features in neural networks, which are closely related to our work. A deep denoised auto-encoder was adopted to learn a shared representation from mutimodal inputs [32], and similarly, a deep Boltzmann machine was utilized to fuse visual and textual features [42]. However, both methods integrated multiple features without elaborately considering the feature correlation and diversity. One important message delivered in this paper is that *regularized fusion* of multiple features is intuitively reasonable and empirically effective, compared with those "free" forms of fusion approaches.

2.2 Exploring Inter-class Relationships

There are many existing works on exploring class relationships, often called context, for improved classification performance. In [44], Torralba et al. highlighted the importance of context in object detection. In [6, 37], co-occurrence context was utilized to enforce object recognition in images. For video classification, Qi et al. [36] proposed to use a multi-label learning method based on ideas from the Gibbs random field. Jiang et al. [19] proposed a semantic diffusion method to utilize class-relationships for video annotation. The algorithm is also capable of adapting the pre-defined class relationships to a new test data domain. Weng et al. [47] proposed a domain-adaptive method that not only explored the class relationships, but also utilized temporal structural information in long broadcast news videos for better annotation performance. Most of these approaches, however, are largely based on the co-occurrence statistics of the video classes, and cannot be used in the cases where the classes share commonalities but do not explicitly co-occur. Our approach can automatically learn such commonalities via a regularized DNN using a rigorous formulation.

The regularization terms used in our approach are partly motivated by recent advances in Multiple Task Learning (MTL) [14, 52]. MTL trains multiple models simultaneously and improves the performance of a task (classifier) with the help of other related tasks. Recent years have witnessed practical successes of MTL in many applications, such as disease prediction [51, 53] and financial stock selection [13]. Sharing commonalities between different tasks is the central idea of MTL and several approaches have been developed with regularizations on shared common patterns across classes [9, 21, 35]. These works considered the class relationships in classification or regression tasks with conventional shallow learning models, but never injected similar regularizations into neural networks.

In fact, multi-layered neural network can be considered as one of the earliest MTL models [8] (see Figure 2 (b)). In such a neural network, each unit in the output layer corresponds to a task and the hidden layer neurons can be regarded as shared common patterns. In this paper, we argue that imposing explicit forms of regularization ensures the deep modeling of complex class relationships for video classification, and thus generates better performance than the traditional neural network with implicit task sharing.

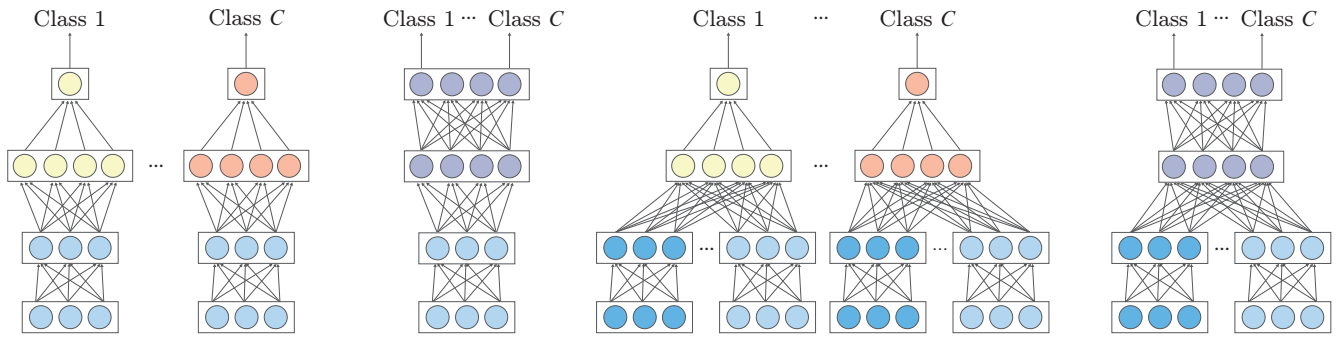
3. METHODOLOGY

This section elaborates our proposed method. We start from introducing the notations and the problem setup. Then a brief introduction of standard DNN is given to support later discussions.

3.1 Notations and Problem Setup

Assume that we are given a training set with N video clips, which are associated with C semantic classes. Here each video clip is represented by M different features, such as various visual and audio descriptors. Therefore, we can denote each training sample as an $(M+1)$-tuple:

$$(\mathbf{x}_n^1, \cdots, \mathbf{x}_n^m, \cdots, \mathbf{x}_n^M, \mathbf{y}_n), n = 1, \cdots, N,$$



(a) Single-feature Single-class (b) Single-feature Multi-class (c) Multi-feature Single-class (d) Multi-feature Multi-class

Figure 2: Illustration of different neural network structures. (b) is the most popular structure for multi-class prediction, while (d) was used in works like [42] to combine multiple features, where features are processed separately in the network and then merged through a middle layer. In this paper, we impose regularizations on the same structure as shown in (d) to explore both the inter-feature and the inter-class relationships.

where $\mathbf{x}_n^m$ represents the m-th feature representation of the n-th video sample, and $\mathbf{y}_n = [y_{n1} \cdots y_{nc} \cdots y_{nC}]^\top \in \mathbb{R}^C$ is the corresponding semantic label with the c-th element $y_{nc} = 1$ if the n-th video sample is associated with the c-th semantic class. The goal is to train prediction models that can classify new test videos. A straightforward way is to independently train one classifier for each semantic class, and different features can be combined using either the early fusion or the late fusion scheme. However, such an independent training strategy does not explore the inter-feature as well as the inter-class relationships. Here, we propose a DNN based video classification model that carries out *feature sharing* in the fusion layer through exploring the correlation and diversity of multiple features, as shown in Figure 1. In addition, the prediction layer of our deep neural network is also regularized to enforce *knowledge sharing* across different classes. Hence, both kinds of relationships are explicitly explored in a uniform learning process. Below we first introduce the standard DNN with a single feature and then present the details of our proposed regularized DNN.

3.2 DNN Learning with Single Feature

Inspired by the biological nervous systems, DNN uses a large number of interconnected neurons to construct complex computational models. Through organizing the neurons in multiple layers, this method possesses strong non-linear abstraction capacity and is able to learn arbitrary mapping functions from inputs to outputs as long as being given sufficient training data. Below we briefly review a standard DNN with only one feature as the input, i.e., $M = 1$.

In a DNN with a total of L layers, we denote $\mathbf{a}_{l-1}$ and $\mathbf{a}_l$ as the input and the output of the l-th layer for a single feature, $l = 1, \cdots, L$, while $\mathbf{W}_l$ and $\mathbf{b}_l$ refer to the weight matrix and the bias vector of the l-th layer, respectively. The transition function from the $(l-1)$-th layer to the l-th layer can be formulated as:

$$\mathbf{a}_l = \begin{cases} \sigma\left(\mathbf{W}_{l-1}\mathbf{a}_{l-1} + \mathbf{b}_{l-1}\right), & l > 1; \\ \mathbf{x}, & l = 1. \end{cases} \tag{1}$$

Here $\sigma(\cdot)$ is a nonlinear sigmoid function, which is often defined as

$$\sigma(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{x}}}.$$

Figure 2 (a) and (b) show two types of four-layered neural networks using a single feature as the input.

To derive the optimal weights for each layer, one can formulate the following optimization problem:

$$\min_{\mathbf{W}} \quad \sum_{i=1}^{N} \ell(f(\mathbf{x}_i), \mathbf{y}_i) + \frac{\lambda_1}{2} \sum_{l=1}^{L-1} \|\mathbf{W}_l\|_F^2, \tag{2}$$

where the first part measures the empirical loss on the training data by summing the discrepancy between the outputs of the network $\hat{\mathbf{y}}_i = \mathbf{a}_L = f(\mathbf{x}_i)$ and the ground-truth labels $\mathbf{y}_i$, and the second part is a regularization term preventing overfitting. For simplicity, we can absorb $\mathbf{b}$ into the weights coefficient $\mathbf{W}$ by adding an additional dimension to the feature vectors with a constant value one.

3.3 Regularization on Feature Relationships

A single feature based DNN can be powerful in some cases. However, it can only be used with a single aspect of the data to perform semantic prediction. For complex data like the videos, the semantic information can be carried by different feature representations including both visual and audio clues. Note that simple fusion strategies, such as the early or late fusion, usually result in limited performance gain since the intrinsic relations among multiple feature representations are overlooked [4]. In addition, such simple fusion methods often incur extra efforts for training the classifiers. Therefore, it is desired to obtain a compact yet meaningful fused representation that fully leverages the complementary clues from various features. Below we extend the basic DNN to a regularized variant that is able to accommodate the deep fusion process of multiple features.

We are given a total of M features

$$\{\mathbf{x}_i^1, \cdots, \mathbf{x}_i^m, \cdots, \mathbf{x}_n^M\}, i = 1, \cdots, n$$

for each video sample. Motivated by the multisensory integration process of primary neurons in biological systems [33,

43], we propose to use one additional layer for the fusion of all the features, as shown in Figure 1. Accordingly, the transition equation for this fusion layer can be written as the following:

$$\mathbf{a}_F = \sigma\left(\sum_{m=1}^{M} \mathbf{W}_E^m \mathbf{a}_E^m + \mathbf{b}_E\right), \tag{3}$$

where E and F are the indices of the last layer of feature extraction and the fusion layer respectively (i.e., $F = E+1$). Here $\mathbf{a}_E^m \in \mathbb{R}^P$ denotes the extracted mid-level representation for the m-th feature which is first linearly transformed by the weight $\mathbf{W}_E^m$ and then non-linearly mapped to the new representation $\mathbf{a}_F$ using a sigmoid function.

Since all the feature representations correspond to the same video data, it is easy to understand that various features can be used to reveal the common latent patterns related to the video semantics. In addition, as mentioned earlier, different features could also be complementary because they have distinct characteristics. Therefore, the fusion process should aim to capture the relations among the features, while being able to reserve their unique characteristics at the same time. Instead of simply adding up multiple feature information, we specifically formulate an objective function that can regularize the fusion process to explore such correlations and diversity among the multiple features simultaneously. In particular, the weights, $\mathbf{W}_E^1, \cdots, \mathbf{W}_E^M$, which transform all features into a shared representation, are first vectorized into P dimensional vectors separately, where P is the product of the $\mathbf{a}_E^m$'s ($m = 1, \cdots, M$) dimension and the $\mathbf{a}_F$'s dimension. Here, we assume the extracted features are of the same dimension. Then we stack these coefficient vectors into a matrix $\mathbf{W}_E \in \mathbb{R}^{P \times M}$, where each column of $\mathbf{W}_E$ corresponds to the weights of a single feature. Hence, the element $\mathbf{W}_E(i,j)$ is given as

$$\mathbf{W}_E(i,j) = \mathbf{W}_E^i(j), \quad i = 1, \cdots, M, \quad j = 1, \cdots, P.$$

Then we can formulate the following objective to design a regularized DNN:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{\Psi}} \quad & \mathcal{L} + \frac{\lambda_1}{2}\left(\sum_{l=1}^{E}\sum_{m=1}^{M} \|\mathbf{W}_l^m\|_F^2 + \sum_{l=F}^{L-1} \|\mathbf{W}_l\|_F^2\right) \\ & + \frac{\lambda_2}{2}\mathrm{tr}(\mathbf{W}_E \mathbf{\Psi}^{-1} \mathbf{W}_E^T) \\ \text{s.t.} \quad & \mathbf{\Psi} \succeq 0, \end{aligned} \tag{4}$$

where $\mathcal{L} = \sum_{i=1}^{N} \ell(\hat{\mathbf{y}}_i, \mathbf{y}_i)$. Compared with the objective function in Equation 2 for the standard single feature neural network, the above cost function includes one additional regularization term. Note that the matrix $\mathbf{W}_E$ represents the coefficients over all the features. Here we use a symmetric and positive semidefinite matrix $\mathbf{\Psi} \in \mathbb{R}^{M \times M}$ to model the inter-feature correlation and introduce the last regularization term with the trace norm that can help learn the inter-feature relationship [12, 52]. Note that the entries with large values in $\mathbf{\Psi}$ indicate strong feature correlations, while small-valued entries denote the diversity among different features since they are less correlated. The coefficients λ_1 and λ_2 control the contributions from different regularization terms. Finally, the objective of learning the regularized DNN is performed as a joint optimization procedure over the weight matrix $\mathbf{W}$ and the feature correlation matrix $\mathbf{\Psi}$.

3.4 Regularization on Class Relationships

To recognize or classify C semantic categories, one can simply adopt the one-vs-all strategy to independently train C classifiers. Figure 2 (a) and (c) illustrate this one-vs-all training scheme with a total of C four-layered neural networks for the single-feature and multi-feature settings, respectively. Clearly, each of these C neural networks is separately learned, where knowledge sharing among different semantic categories is completely neglected. However, it is well recognized that video semantics also share some *commonality*, which indicates that certain semantic categories could be strongly correlated [19, 36]. Therefore, it is critical to explore such a commonality by simultaneously learning multiple video semantics, which can often lead to better learning performance. Note that, the commonality among multiple classes is often represented by the parameter sharing among different prediction models [3, 26]. Compared with the popular SVM method, it is more natural for DNN to perform multi-class training simultaneously. As shown in Figure 2 (b), by adopting a set of C units in the output layer, a single-feature based DNN can be easily extended to multi-class problems, and this structure has been widely adopted. Motivated by the regularization framework used in the standard MTL methods [3, 26], here we present a regularized DNN that aims at training multiple classifiers simultaneously with deeper exploration of the class relationships. To enforce the semantic sharing, we extend the original objective for a standard DNN to the following form:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{\Omega}} \quad & \sum_{i=1}^{N} \ell(f(\mathbf{x}_i), \mathbf{y}_i) + \frac{\lambda_1}{2}\sum_{l=1}^{L-1} \|\mathbf{W}_l\|_F^2 \\ & + \lambda_2 \mathrm{tr}(\mathbf{W}_{L-1} \mathbf{\Omega}^{-1} \mathbf{W}_{L-1}^T). \\ \text{s.t.} \quad & \mathbf{\Omega} \succeq 0. \end{aligned} \tag{5}$$

Note that some previous MTL works assumed that the class relationships are explicitly given and are ready for use as prior knowledge [26], while our method does not require this. Following the convex formulation of MTL [52], here we impose a trace norm regularization term over the coefficients $\mathbf{W}_{L-1}$ of the output layer with the class relationships augmented as a matrix variable $\mathbf{\Omega} \in \mathbb{R}^{C \times C}$. Note that the constraint $\mathbf{\Omega} \succeq 0$ indicates that the class relationship matrix is positive semidefinite since it can be viewed as the similarity measure of the semantic classes. The coefficients λ_1 and λ_2 are regularization parameters. During the learning procedure, the optimal weight matrices $\{\mathbf{W}_l\}_{l=1}^{L}$ and the class relationship matrix $\mathbf{\Omega}$ are simultaneously derived.

3.5 The Unified Objective

To unify the above objectives into a joint framework, we now present a novel DNN formulation that explores both the inter-feature and the inter-class relationships. In our framework, we use one layer of neurons to fuse multiple features, where the objective is to bridge the gap between low-level features and the high-level video semantics. In the final layer of generating the predictions, we impose a trace norm regularization among different semantics to better learn the predictions of multiple classes. Mathematically, we incorporate the feature regularization in Equation 4 and the class regu-

larization in Equation 5 into the following objective function:

$$\begin{aligned}\min_{\mathbf{W},\mathbf{\Psi},\mathbf{\Omega}} \quad & \mathcal{L} + \frac{\lambda_1}{2}\left(\sum_{l=1}^{E}\sum_{m=1}^{M}\|\mathbf{W}_l^m\|_F^2 + \sum_{l=F}^{L-1}\|\mathbf{W}_l\|_F^2\right) \\ & + \frac{\lambda_2}{2}\mathrm{tr}(\mathbf{W}_E\mathbf{\Psi}^{-1}\mathbf{W}_E^T) \\ & + \frac{\lambda_3}{2}\mathrm{tr}(\mathbf{W}_{L-1}\mathbf{\Omega}^{-1}\mathbf{W}_{L-1}^T), \\ \text{s.t.} \quad & \mathbf{\Psi} \succeq 0 \quad \mathrm{tr}(\mathbf{\Psi}) = 1, \\ & \mathbf{\Omega} \succeq 0 \quad \mathrm{tr}(\mathbf{\Omega}) = 1,\end{aligned} \tag{6}$$

where λ_1, λ_2, and λ_3 are regularization parameters. Compared with the original objective in Equation 2, we have two trace-norm regularization terms that are tailored for the fusion of multiple features and the exploration of the inter-class relationships, respectively. Two additional constraints $\mathrm{tr}(\mathbf{\Psi}) = 1$ and $\mathrm{tr}(\mathbf{\Omega}) = 1$ are used to restrict the complexity, as suggested in [52]. Finally, the above cost function is minimized with respect to the network weights $\{\mathbf{W}_l\}_{l=1}^L$, the inter-feature relationship matrix $\mathbf{\Psi}$, and the inter-class correlation matrix $\mathbf{\Omega}$.

3.6 Optimization

Among the variables in the minimization problem of Equation 6, two pairs of variables, i.e., $(\mathbf{W}_E, \mathbf{\Psi})$ and $(\mathbf{W}_{L-1}, \mathbf{\Omega})$, are coupled with each other. Therefore, we adopt the alternative optimization method to iteratively minimize the objective with respect to $\mathbf{W}_l^m$ $(l = 1, \cdots L, m = 1, \cdots, M)$, $\mathbf{\Psi}$, and $\mathbf{\Omega}$.

By fixing both $\mathbf{\Psi}$ and $\mathbf{\Omega}$, we first consider the minimization problem over $\mathbf{W}_l^m$, which is degenerated to a set of unconstrained univariate optimization problems:

$$\begin{aligned}\min_{\mathbf{W}_l^m} \quad & \mathcal{L} + \frac{\lambda_1}{2}\left(\sum_{l=1}^{E}\sum_{m=1}^{M}\|\mathbf{W}_l^m\|_F^2 + \sum_{l=F}^{L-1}\|\mathbf{W}_l\|_F^2\right) \\ & + \frac{\lambda_2}{2}\mathrm{tr}(\mathbf{W}_E\mathbf{\Psi}^{-1}\mathbf{W}_E^T) + \frac{\lambda_3}{2}\mathrm{tr}(\mathbf{W}_{L-1}\mathbf{\Omega}^{-1}\mathbf{W}_{L-1}^T).\end{aligned} \tag{7}$$

As all the terms of the above objective function are smooth, the gradient can be easily evaluated. Denote $\mathbf{G}_l^m$ as the gradient with respect to $\mathbf{W}_l^m$, the weight matrix for the l-th layer and the m-th feature is updated as:

$$\mathbf{W}_l^m = \mathbf{W}_l^m - \eta\mathbf{G}_l^m, \tag{8}$$

where η is the step length of the gradient descent.

We then introduce the solution of minimizing the objective function over $\mathbf{\Psi}$ with other variables being fixed. The problem in Equation 6 degenerates to:

$$\begin{aligned}\min_{\mathbf{\Psi}} \quad & \mathrm{tr}(\mathbf{W}_E\mathbf{\Psi}^{-1}\mathbf{W}_E^T), \\ \text{s.t.} \quad & \mathbf{\Psi} \succeq 0 \quad \mathrm{tr}(\mathbf{\Psi}) = 1.\end{aligned} \tag{9}$$

Adopting the Cauchy-Schwarz inequality, we obtain the analytical solution for the above minimization problem as:

$$\mathbf{\Psi} = \frac{(\mathbf{W}_E^T\mathbf{W}_E)^{\frac{1}{2}}}{\mathrm{tr}((\mathbf{W}_E^T\mathbf{W}_E)^{\frac{1}{2}})}. \tag{10}$$

Similarly, the optimal solution for $\mathbf{\Omega}$ is derived as:

$$\mathbf{\Omega} = \frac{(\mathbf{W}_{L-1}^T\mathbf{W}_{L-1})^{\frac{1}{2}}}{\mathrm{tr}((\mathbf{W}_{L-1}^T\mathbf{W}_{L-1})^{\frac{1}{2}})}. \tag{11}$$

Algorithm 1 Training Procedure of Regularized DNN

Require: $\mathbf{x}_n^m$: the representation of the m-th feature for the n-th video sample; $\mathbf{y}_n$: the semantic label of the n-th video sample;

1: Initialize $\mathbf{W}_l^m$ randomly, $\mathbf{\Psi} = \frac{1}{M}\mathbf{I}_M$ and $\mathbf{\Omega} = \frac{1}{C}\mathbf{I}_C$, where $\mathbf{I}_M$ and $\mathbf{I}_C$ are identity matrices;
2: **for** *epoch* $= 1$ to K **do**
3: Back propagate the prediction error from layer L to layer 1 by evaluating the gradient $\mathbf{G}_l^m$, and update the weight matrix $\mathbf{W}_l^m$ for each layer and each feature as:

$$\mathbf{W}_l^m = \mathbf{W}_l^m - \eta\mathbf{G}_l^m;$$

4: Update the feature relationship matrix $\mathbf{\Psi}$ according to Equation 10:

$$\mathbf{\Psi} = \frac{(\mathbf{W}_E^T\mathbf{W}_E)^{\frac{1}{2}}}{\mathrm{tr}((\mathbf{W}_E^T\mathbf{W}_E)^{\frac{1}{2}})};$$

5: Update the class relationship matrix $\mathbf{\Psi}$ according to Equation 11:

$$\mathbf{\Omega} = \frac{(\mathbf{W}_{L-1}^T\mathbf{W}_{L-1})^{\frac{1}{2}}}{\mathrm{tr}((\mathbf{W}_{L-1}^T\mathbf{W}_{L-1})^{\frac{1}{2}})}.$$

6: **end for**

Note that Zhang et al. adopted a similar solution as in Equation 11 to identify task correlations for a linear kernel based regression and classification task [52]. However, our method integrates more complex structural regularizations in a neural network architecture, where both the inter-feature and the inter-class relationships are explored for a completely different application. Hence, the difference of our method is fairly significant.

In our approach, the inter-feature and inter-class relationships are first estimated based on the corresponding weights in the neural networks. The relationships are then used in turn to adjust the network weights for improved classification performance. Using the trace norm allows us to derive the analytical solution in Equation 10 and Equation 11, which satisfies our goal of learning the relationships $\mathbf{\Psi}$ and $\mathbf{\Omega}$ based on $\mathbf{W}$. More specifically, the training procedure of the proposed method is summarized in Algorithm 1. For each epoch, additional efforts are required to compute the gradient matrix $\mathbf{G}_l^m$ for updating $\mathbf{W}_l^m$, as well as to update the matrices $\mathbf{\Omega}$ and $\mathbf{\Psi}$. The complexity of calculating the trace norms is the same as that of the $\ell 2$ norm. The update of $\mathbf{\Omega}$ and $\mathbf{\Psi}$ requires operations with a cubic complexity with respect to the number of features M and the number of video classes C, respectively. Note that these two numbers are often relatively much smaller than the number of data points used for DNN training. Therefore, the training cost of the proposed regularized DNN is very similar to that of a standard DNN. Our empirical study further confirms the efficiency of our method, as will be discussed in the next section.

4. EXPERIMENTS

4.1 Experimental Setup

4.1.1 Datasets and Evaluation Measure

We adopt three challenging benchmarks on action and event recognition to evaluate our method, as described in the following.

Hollywood2 [24]. The Hollywood2 dataset is one of the most popular benchmarks on action recognition in videos. Collected from 69 Hollywood movies, it contains 1,707 action video clips covering 12 classes: answering phone, driving car, eating, fighting, getting out of car, hand shaking, hugging, kissing, running, sitting down, sitting up and standing up. Following [24], the dataset is split into a training set with 823 videos and a test set with 884 videos.

Columbia Consumer Videos (CCV) [20]. The CCV dataset is a well-known benchmark on Internet consumer video analysis. It contains 9,317 videos collected from YouTube with annotations of 20 semantic classes, including objects (e.g., "cats"), scenes (e.g., "playground"), and events (e.g., "parade"). Since most of the classes are complex events, it requires a joint use of multiple feature clues like visual and audio representations to perform better classification. The dataset is evenly split into a training set and a test set.

CCV+. Since both the Hollywood2 and the CCV datasets are small in terms of the number of annotated classes, we additionally collected and annotated another 20 classes with in total of 5,159 video clips. These clips are merged with the CCV to form a larger dataset of 40 classes, named CCV+, containing 7,244 videos for training and 7,232 videos for testing. See Figure 3 for the list of class names and a few example frames.

For all the three datasets, the performance is measured by average precision (AP) for each class and mean AP (mAP) for overall results of all the classes.

4.1.2 Feature Representations

Visual Features. For the visual features, we consider the dense trajectory descriptors [46], which have exhibited strong performance on various benchmark datasets. Briefly, densely sampled local frame patches are first tracked over time and four descriptors are then computed for each trajectory: a 30-d trajectory shape descriptor, a 96-d histogram of oriented gradients (HOG) descriptor, a 108-d histogram of optical flow (HOF) descriptor, and a 108-d motion boundary histogram (MBH) descriptor. Finally, each descriptor is quantized into a 4,000-d bag-of-words representation, same as the settings used in [46].

Audio Features. It is well-known that the audio soundtracks contain useful clues for identifying some video semantics. Two types of video features are considered in our study. The first one is the popular MFCCs (Mel-Frequency Cepstral Coefficients), which are computed for every 32ms time-window with 50% overlap and then quantized into a bag-of-words representation. The second one is called Spectrogram SIFT [54]. The 1-d soundtrack of a video is transformed into a 2-D image based on the constant-Q spectrogram, on which standard SIFT descriptors are extracted and quantized into the bag-of-words representation.

Note that all these visual and audio features are adopted because they have demonstrated strong performance on various benchmarks; however, evaluating the performance of each single feature in detail is beyond the scope of this work. All the representations are normalized with RootSift [2], which has been shown to be more suitable for histogram-based features than the conventional L2 normalization.

4.1.3 Compared Approaches

To validate the effectiveness of our method, we compare with the following approaches:

Early Fusion with Neural Networks (NN-EF). All the features are concatenated into a long vector and then used as the input to train a neural network.

Late Fusion with Neural Networks (NN-LF). A separate neural network is trained using each feature independently and then the outputs of all the networks are fused to obtain the final classification results.

Early Fusion with SVM (SVM-EF). The popular χ^2 kernel SVM is adopted and the features are combined on the kernel level before classification.

Late Fusion with SVM (SVM-LF). A separate SVM classifier is trained for each feature and prediction results are then combined.

Multimodal Deep Boltzmann Machines (M-DBM). It is a fusion model proposed in [42], where multiple feature representations are used as the inputs of the Deep Boltzmann Machines.

Discriminative Model Fusion (DMF) [40]. As one of the earliest approaches in multimedia for context-based classification, DMF uses the outputs of an initial classifier, e.g., a standard DNN in our case, as features to train an SVM model as the second level classifier for final prediction.

Domain Adaptive Semantic Diffusion (DASD) [19]. This method uses a graph diffusion formulation for context-based classification. The prediction outputs of a normal DNN (without the regularizations) are used as inputs of the DASD in a post-processing refinement step. The approach also requires input of precomputed class relationships, which are usually estimated based on statistics of label co-occurrences in training data.

The first five approaches can be regarded as alternatives for feature fusion, while the last two approaches focus on the use of the class relationships. All the neural network based experiments are conducted on a single Nvidia Telsa K20 5GB GPU with MATLAB Parallel Computing Toolbox, which speeds up the training procedure by about 5 times than a decent Intel XEON CPU with 16 cores.

4.2 Results and Analysis

In this section, we first report results of our approach by disabling the regularizations on the output layer and the fusion layer respectively, in order to understand the contributions of only exploring the inter-feature or the inter-class relationships. This also ensures fair comparisons with the competing approaches. After that, we report results of using the entire framework, compare with recent state-of-the-art results, and analyze the effect of the number of training samples. Finally, we provide discussions on computational efficiency.

Throughout the experiments, we set the learning rate of the neural networks to 0.7, fix λ_1 to 3×10^{-5} in order to prevent overfitting, and tune λ_2 and λ_3 in the same range as λ_1. Following the conventional settings of DNN, we also adopt the mini batch gradient descent with the batch size being 70.

4.2.1 Effect of Exploring Feature Relationships

We first report results by only using fusion layer regularization on the DNN with the output layer regularization being disabled. Table 1 compares our approach, namely DNN-Fusion Regularization (DNN-FR), with the alternative feature fusion methods. As seen in the table, our approach achieves the best performance with clear gains over

Approaches	Hollywood2	CCV	CCV+
NN-EF	62.0%	66.7%	70.5%
NN-LF	58.5%	61.9%	64.7%
SVM-EF	62.6%	67.5%	70.0%
SVM-LF	62.1%	64.9%	68.5%
M-DBM[42]	61.5%	67.2%	70.1%
DNN-FR	**64.5%**	**69.1%**	**71.8%**

Table 1: Performance comparison (mAP) on the three datasets, using approaches that only explore the inter-feature relationships.

Approaches	Hollywood2	CCV	CCV+
DMF [40]	61.8%	67.6%	68.5%
DASD [19]	60.9%	66.8%	70.2%
DNN-CR	**63.0%**	**69.3%**	**72.1%**

Table 2: Performance comparison (mAP) on the three datasets, using approaches that only explore the inter-class relationships.

all the compared methods. Note that the M-DBM approach also utilizes a neural network for feature fusion, but in a *free* manner without explicitly taking feature relations into the learning process. These results validate the effectiveness of imposing the proposed fusion regularization in neural networks. Notice that, since these adopted datasets are very challenging and have been widely used, an absolute performance gain of 2% is generally considered as a significant improvement.

Comparing across the alternative approaches, early fusion tends to generate better results than late fusion. This observation is consistent with recent works, where the early fusion was more popularly used [1]. The neural network based approaches do not show significant gain over the SVM-based ones because the amount of training data in video classification is limited. With more training samples, the margin is expected to be significantly larger. In addition, for the contribution of the audio clues, we observed that for the classes with strong audio clues, such as "answering phone", adding audio features clearly improves the performance. On the contrary, for classes like "sitting down", using audio features may slightly degrade the result, which is easy to understand.

4.2.2 *Effect of Exploring Class Relationships*

Next we analyze the effect of solely imposing the regularization on the classification/output layer to explore the inter-class relationships, while disabling the regularization on the fusion layer. We compare our method, named as DNN-Classification Regularization (DNN-CR), with DMF and DASD and report the results in Table 2. Clearly, DNN-CR outperforms these two compared approaches, both of which use the outputs of the conventional DNN as inputs for context-based refinement. These results corroborate the effectiveness of the proposed regularization on the output layer. Note that the simple DMF is superior than the DASD because the latter requires pre-computed inter-class relationships, which are normally estimated based on the label co-occurrences in training data. However, some classes that share commonalities may not visually co-occur, resulting in

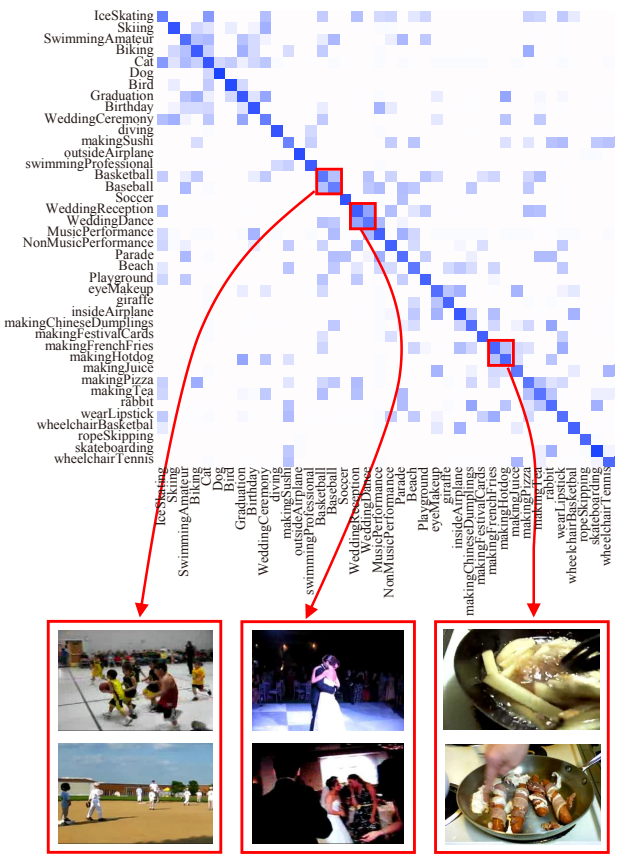


Figure 3: Class relationships in CCV+ indicated by the learned matrix $\mathbf{\Omega}$. Example video frames of a few found class groups are shown at the bottom.

a very sparse class relationship matrix that is insufficient for context-based learning. Several other alternative methods such as those based on the conditional random field [37] also rely on the relationship matrix and thus suffer from the same limitation.

To further show the power of our method in learning the class relationships, we visualize some results in Figure 3. As discussed in Section 3, values in the matrix $\mathbf{\Omega}$ can reflect the learned correlations among the classes. Hence, we can adopt spectral clustering on the matrix $\mathbf{\Omega}$ to group the video semantics, which are then re-ordered for the ease of visualization. We see that many classes sharing commonalities are grouped together, which confirms that our method can reveal the hidden class relationships.

4.2.3 *Results of the Entire Framework*

We now discuss the results of the entire framework, i.e., using regularizations on both the fusion layer and the output layer. In addition, to evaluate the performance using different amounts of training data, we plot the performance w.r.t. the number of training samples in Figure 4. Overall, substantial performance gains are observed from using the regularized DNN framework. Using regularizations on both layers also clearly achieves higher performance than solely imposing regularization on a single layer (i.e., only on $\mathbf{W_E}$) with clear margins. When there are less training samples, the improvement of our method is even more

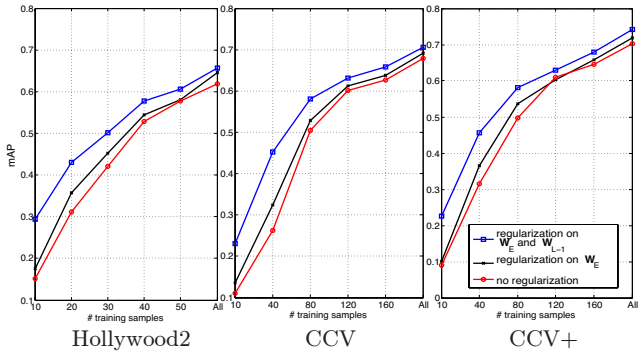


Figure 4: Performance on the three datasets using different number of training samples. We plot the results of DNN without regularization (red), DNN with regularization only on the fusion layer (black), and DNN with regularization on both the fusion and the output layers (blue). Consistent performance gains are obtained from imposing the proposed regularizations.

Hollywood2	mAP	CCV	mAP
Mathe et al. [28]	61.0%	Jiang et al. [20]	59.5%
Jain et al. [15]	62.5%	Jhuo et al. [16]	64.0%
Oneata et al. [34]	63.4%	Xu et al. [48]	60.3%
Wang et al. [46]	64.3%	Ye et al. [50]	64.0%
Regularized DNN	**65.7%**	Regularized DNN	**70.6%**

Table 3: Comparison with state-of-the-art results in the literature. Our method (Regularized DNN) achieves to-date the highest mAP on both the Hollywood2 and the CCV datasets.

Approaches	Training Time (s)
NN-EF	1.068±0.021
NN-LF	0.782±0.007
Regularized DNN	**0.640±0.002**

Table 4: Training time per epoch (seconds) of the neural network based approaches on the Hollywood2 dataset.

significant. An improvement of around 100% is obtained on all the three datasets when there are just 10 training samples per class. This demonstrates that the regularized DNN requires much less training data to achieve comparable performance to the non-regularized DNN. In addition, comparing the performance across the three datasets using all the training samples, the gain from exploring the class relationships is more significant on CCV+. This is because CCV+ has more classes and thus contains richer inter-class relationships that are helpful for classification.

Table 3 further compares our best results (from regularization on both layers) with several recently reported results on the Hollywood2 and the CCV datasets. On Hollywood2, our proposed method achieves the best mAP of 65.7%, outperforming all the recent results [15, 28, 46]. All these works are based on the dense trajectory features, and they performed classification using the simple early fusion method. Note that Wang et al. [46] and Oneata et al. [34] used the Fisher vector to encode the features, which has been shown to be more effective than the traditional bag-of-words representation [38]. However, the dimension of the Fisher vectors is too high to be used as inputs of the neural networks, since there is no sufficient training data to tune the numerous parameters. Therefore, we adopt the standard bag-of-words in this work, and it is very appealing to observe higher performance than the approaches using the Fisher vectors.

For the CCV dataset, several recent works have focused on the joint use of multiple audio-visual features. Xu et al. [48] and Ye et al. [50] adopted late fusion with specially designed methods to remove the noise of individually trained classifiers, and Jhuo et al. used a joint audio-visual codebook for classification [16]. Our approach is fundamentally different from these state-of-the-art methods in its design and produces significantly higher performance.

4.2.4 *Computational Efficiency*

Finally, we briefly compare and discuss the computational efficiency, using the Hollywood2 dataset. The average training time of each epoch for NN-EF, NN-LF and M-DBM are presented in Table 4, using a GPU-based implementation as mentioned in Section 4.1.3. Our proposed method is more efficient than NN-EF and NN-LF as our regularized DNN contains less parameters to be learned. Specifically, compared with the early fusion, our framework processes the features separately in the first two layers and thus avoids the parameters interacting among them. The late fusion method requires the training of separate networks, which is also more expensive. Note that we exclude the M-DBM approach in this comparison, because it requires significant additional time to pre-train the network for weight initialization. For all the methods, several hundreds of epochs are normally needed to finish the training process (several minutes in total). Once the training is done, all these neural network based methods are extremely fast in terms of performing predictions on testing videos.

5. CONCLUSION

We have introduced a novel DNN framework that explores both inter-feature and inter-class relationships to achieve better classifications on video semantics. By imposing trace-norm based regularizations on a specially designed fusion layer and an output layer in the neural network, our method can learn a fused representation of multiple feature inputs and utilize the commonalities among the semantic classes for improved classification performance. Extensive experiments on popular benchmarks of action and event recognition have shown that our method consistently outperforms the alternative approaches as well as the recent state-of-the-art works. In addition, the proposed method is also similar to even faster than the traditional approaches in terms of the model training, which is very important for large scale applications. One important future work is to add the function of learning feature representations directly from raw video data in the framework, which would require much more training data as discussed earlier but may lead to substantial further performance improvements. Therefore, it would also be interesting and valuable if this extension could be done with an effort of collecting and annotating a larger collection

of videos, like the Image-Net effort for image analysis [11], which is urgently needed to stimulate the research on large scale video classification.

Acknowledgement

This work was supported in part by a National 863 Program (#2014AA015101), a National Key Technologies Research and Development Program (#2013BAH09F01), a grant from NSF China (#61201387), and three grants from the Science and Technology Commission of Shanghai Municipality (#13PJ1400400, #13511504503, #12511501602).

6. REFERENCES

[1] R. Aly, R. Arandjelovic, K. Chatfield, and et al. The AXES submissions at TrecVid 2013. In *NIST TRECVID Workshop*, 2013.

[2] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012.

[3] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 2008.

[4] P. K. Atrey, M. A. Hossain, A. El Saddik, and M. S. Kankanhalli. Multimodal fusion for multimedia analysis: a survey. *Multimedia systems*, 2010.

[5] F. R. Bach, G. R. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *ICML*, 2004.

[6] S. Bengio, J. Dean, D. Erhan, E. Ie, Q. Le, A. Rabinovich, J. Shlens, and Y. Singer. Using web co-occurrence statistics for improving image categorization. *arXiv:1312.5697*, 2013.

[7] L. Cao, J. Luo, F. Liang, and T. S. Huang. Heterogeneous feature machines for visual recognition. In *ICCV*, 2009.

[8] R. Caruana. Multitask learning. *Machine learning*, 1997.

[9] J. Chen, J. Zhou, and J. Ye. Integrating low-rank and group-sparse structures for robust multi-task learning. In *ACM SIGKDD*, 2011.

[10] C. V. Cotton and D. P. Ellis. Subband autocorrelation features for video soundtrack classification. In *ICASSP*, 2013.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[12] H. Fei and J. Huan. Structured feature selection and task relationship inference for multi-task learning. *Knowledge and information systems*, 2013.

[13] J. Ghosn and Y. Bengio. Multi-task learning for stock selection. In *NIPS*, 1997.

[14] L. Jacob, F. Bach, J.-P. Vert, et al. Clustered multi-task learning: A convex formulation. In *NIPS*, 2008.

[15] M. Jain, H. Jégou, P. Bouthemy, et al. Better exploiting motion for better action recognition. In *CVPR*, 2013.

[16] I.-H. Jhuo, G. Ye, S. Gao, D. Liu, Y.-G. Jiang, D. T. Lee, and S.-F. Chang. Discovering joint audio-visual codewords for video event detection. *Machine Vision and Applications*, 2014.

[17] W. Jiang, C. Cotton, S.-F. Chang, D. Ellis, and A. Loui. Short-term audio-visual atoms for generic video concept classification. In *ACM Multimedia*, 2009.

[18] W. Jiang and A. C. Loui. Audio-visual grouplet: temporal audio-visual interactions for general video concept classification. In *ACM Multimedia*, 2011.

[19] Y.-G. Jiang, Q. Dai, J. Wang, C.-W. Ngo, X. Xue, and S.-F. Chang. Fast semantic diffusion for large-scale context-based image and video annotation. *IEEE TIP*, 2012.

[20] 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 *ACM ICMR*, 2011.

[21] Z. Kang, K. Grauman, and F. Sha. Learning with whom to share in multi-task feature learning. In *ICML*, 2011.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[23] Z.-Z. Lan, L. Jiang, S.-I. Yu, S. Rawat, Y. Cai, C. Gao, S. Xu, H. Shen, X. Li, Y. Wang, et al. Cmu-informedia@ trecvid 2013 multimedia event detection. In *NIST TRECVID Workshop*, 2013.

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

[25] D. Liu, K.-T. Lai, G. Ye, M.-S. Chen, and S.-F. Chang. Sample-specific late fusion for visual category recognition. In *CVPR*, 2013.

[26] J. Liu, S. Ji, and J. Ye. Multi-task feature learning via efficient l 2, 1-norm minimization. In *UAI*, 2009.

[27] S. Maji, A. C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR*, 2008.

[28] S. Mathe and C. Sminchisescu. Dynamic eye movement datasets and learnt saliency models for visual action recognition. In *ECCV*, 2012.

[29] M. Mazloom, X. Li, and C. G. M. Snoek. Few-example video event retrieval using tag propagation. In *ACM ICMR*, 2014.

[30] P. Natarajan, P. Natarajan, S. Wu, X. Zhuang, A. Vazquez-reina, S. N. Vitaladevuni, C. Andersen, R. Prasad, G. Ye, D. Liu, et al. Bbn viser trecvid 2012 multimedia event detection and multimedia event recounting systems. In *NIST TRECVID Workshop*, 2012.

[31] 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.

[32] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Ng. Multimodal deep learning. In *ICML*, 2011.

[33] T. Ohshiro, D. Angelaki, and G. DeAngelis. A normalization model of multisensory integration. *Nature Neuroscience*, 2011.

[34] D. Oneata, J. Verbeek, C. Schmid, et al. Action and event recognition with fisher vectors on a compact feature set. In *ICCV*, 2013.

[35] J. Pu, Y.-G. Jiang, J. Wang, and X. Xue. Multiple task learning using iteratively reweighted least square. In *IJCAI*, 2013.

[36] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, T. Mei, and H.-J. Zhang. Correlative multi-label video annotation. In *ACM Multimedia*, 2007.

[37] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *ICCV*, 2007.

[38] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *IJCV*, 2013.

[39] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep Fisher networks for large-scale image classification. In *NIPS*, 2013.

[40] J. R. Smith, M. Naphade, and A. Natsev. Multimedia semantic indexing using model vectors. In *ICME*, 2003.

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

[42] N. Srivastava and R. Salakhutdinov. Multimodal learning with deep boltzmann machines. In *NIPS*, 2012.

[43] B. E. Stein and T. R. Stanford. Multisensory integration: current issues from the perspective of the single neuron. *Nature Reviews Neuroscience*, 2008.

[44] A. Torralba. Contextual priming for object detection. *IJCV*, 2003.

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

[46] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[47] M.-F. Weng and Y.-Y. Chuang. Cross-domain multicue fusion for concept-based video indexing. *IEEE TPAMI*, 2012.

[48] Z. Xu, Y. Yang, I. Tsang, N. Sebe, and A. Hauptmann. Feature weighting via optimal thresholding for video analysis. In *ICCV*, 2013.

[49] R. Yan, M.-O. Fleury, M. Merler, A. Natsev, and J. R. Smith. Large-scale multimedia semantic concept modeling using robust subspace bagging and mapreduce. In *ACM Workshop on Large-scale Multimedia Retrieval and Mining*, 2009.

[50] G. Ye, D. Liu, I.-H. Jhuo, and S.-F. Chang. Robust late fusion with rank minimization. In *CVPR*, 2012.

[51] D. Zhang and D. Shen. Multi-modal multi-task learning for joint prediction of multiple regression and classification variables in alzheimer's disease. *Neuroimage*, 2012.

[52] Y. Zhang and D.-Y. Yeung. A convex formulation for learning task relationships in multi-task learning. In *UAI*, 2010.

[53] J. Zhou, L. Yuan, J. Liu, and J. Ye. A multi-task learning formulation for predicting disease progression. In *ACM SIGKDD*, 2011.

[54] B. Zhu, W. Li, Z. Wang, and X. Xue. A novel audio fingerprinting method robust to time scale modification and pitch shifting. In *ACM Multimedia*, 2010.