Multi-Stream Multi-Class Fusion of Deep Networks for Video Classification

Zuxuan Wu, Yu-Gang Jiang*, Xi Wang, Hao Ye, Xiangyang Xue
School of Computer Science, Shanghai Key Lab of Intelligent Information Processing,
Fudan University, Shanghai, China
{zxwu, ygj, xwang10, haoye10, xyxue}@fudan.edu.cn

ABSTRACT

This paper studies deep network architectures to address the problem of video classification. A multi-stream framework is proposed to fully utilize the rich multimodal information in videos. Specifically, we first train three Convolutional Neural Networks to model spatial, short-term motion and audio clues respectively. Long Short Term Memory networks are then adopted to explore long-term temporal dynamics. With the outputs of the individual streams on multiple classes, we propose to mine class relationships hidden in the data from the trained models. The automatically discovered relationships are then leveraged in the multi-stream multi-class fusion process as a *prior*, indicating *which* and *how much* information is needed from the remaining classes, to adaptively determine the optimal fusion weights for generating the final scores of each class. Our contributions are two-fold. First, the multi-stream framework is able to exploit multimodal features that are more comprehensive than those previously attempted. Second, our proposed fusion method not only learns the best weights of the multiple network streams for each class, but also takes class relationship into account, which is known as a helpful clue in multi-class visual classification tasks. Our framework produces significantly better results than the state of the arts on two popular benchmarks, 92.2% on UCF-101 (without using audio) and 84.9% on Columbia Consumer Videos.

Keywords

Video Classification; CNN; LSTM; Fusion.

1. INTRODUCTION

The sheer volume of video data nowadays demands robust video classification techniques that can effectively recognize human actions and complex events for applications like video search, summarization, intelligent surveillance and *etc.* However, it is a particularly challenging problem due to the complicated nature of videos, including large intra-class variations caused by different viewing conditions and multiple view points, noisy contents unrelated to the video topic, and complex temporal structures incurring understanding and computational difficulties. The fact that videos are intrinsically multimodal requires solutions that can explore not only static visual information, but also motion and auditory clues. Key to the development of video classification systems is the design of good features. Popular feature descriptors include the SIFT [33], the Mel-Frequency Cepstral Coefficients (MFCC) [66], the STIP [31] and the dense trajectories [56], which can be encoded into video-level representations by bag-of-words (BoW) [50, 68, 37] or Fisher vectors (FV) [40, 43, 30, 70].

In contrast to the hand-engineered descriptors, the deep neural networks that can learn features automatically from raw data have demonstrated strong performance in various domains. In particular, the convolutional neural networks (ConvNets) are very successful on image analysis tasks like object detection [14], object recognition [46, 51] and image segmentation [11]. However, for video classification,

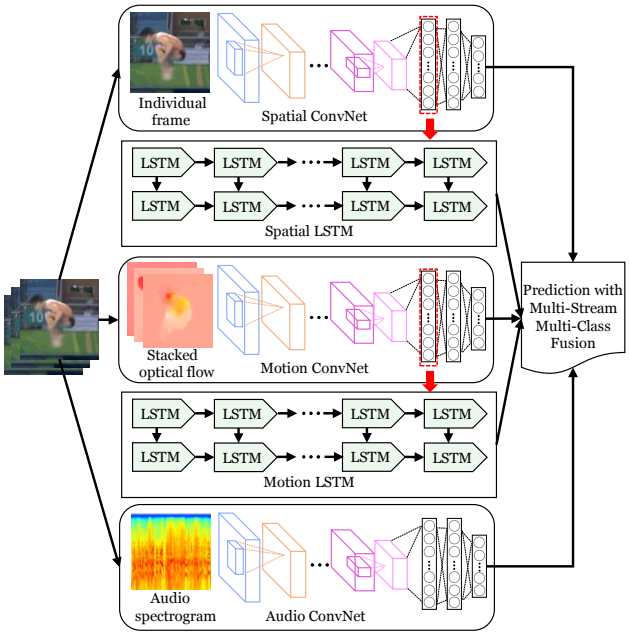

Figure 1: Illustration of the proposed framework.

*Corresponding author.

MM '16, October 15-19, 2016, Amsterdam, Netherlands

DOI: http://dx.doi.org/10.1145/2964284.2964328

most deep network based approaches [22, 26, 45, 62] demonstrated worse or similar results to the hand-engineered features [56]. This is largely due to the high complexity of the video data. Unlike images that only have static visual appearance information, videos also contain temporal motions and auditory soundtracks. For example, a "diving" action video usually involves a sequence of atoms, such as "jumping from a platform", "rotating in the air" and "falling into water", accompanied by cheering or clapping sounds. Some approaches [22, 26, 45] only focused on the static frames and short-term motion clues captured by a few adjacent frames, which are apparently not sufficient. A few very recent studies attempted to use recurrent neural networks (RNN) to model long-term temporal information and achieved competitive performance [39, 64]. Nevertheless, the audio information has rarely been exploited. Furthermore, most existing approaches fused the outputs of multiple networks in a very straightforward way using simple classifiers like logistic regression, which could lead to sub-optimal performance.

In addition, existing works for video categorization often assign single or multiple class labels to a video sample independently without considering the relationships among video semantics. However, humans do not recognize an object (concept) separately but rely on the interconnections of objects (concepts). The presence of related classes could help better categorize the class of interest. For example, "marathon" and "marching band" contain similar human motion patterns (at least when compared with unrelated class pairs like "marathon" and "fishing"), and the confidence of a video containing "marathon" could potentially help recognize "marching band". In other words, if a video receives extremely low score of "marathon", it is also unlikely to be "marching band". To leverage semantic relationships (*i.e.*, context knowledge), many existing methods rely on computational expensive models (*e.g.*, CRF), which are not feasible for large-scale applications.

Realizing the above limitations, in this paper, we propose a multi-stream framework of deep neural networks to exploit the multimodal clues for video classification. Figure 1 illustrates the diagram of our approach. Three ConvNets are trained to model the static spatial information, short-term motion and auditory clues, respectively. The motion stream is computed on stacked optical flows over a short temporal windows and thus can only capture short-term motion. In order to model the long-term temporal clues, we employ a Recurrent Neural Network (RNN) model, namely the Long Short Term Memory (LSTM), on the frame-level spatial and motion features extracted by the ConvNets. The LSTM encodes history information in memory units regulated with non-linear gates to discover temporal dependencies. To combine the outputs from different networks, we develop a simple yet effective fusion method to learn the optimal fusion weights adaptively for each class. Note that the deep models are trained using state-of-the-art networks, they possess high discriminative power and hence contain valuable knowledge on how classes are correlated. Therefore, we propose to leverage the class relationships hidden in the data to constrain the learning process, by informing the classifier *which* classes are related and *how much* information is needed from the each of these related classes. In other words, to generate the final prediction of a class of interest, the classifier also considers the predictions of other correlated classes.

Our contributions are summarized as follows:

1. We introduce a multi-stream framework that integrates spatial, short-term motion, long-term temporal and auditory clues in videos. We demonstrate the multi-stream networks are able to digest complementary clues to receive significantly improved performance.

2. We propose a multi-stream multi-class fusion method to combine the outputs of the individual networks. The method not only learns the weights of individual network streams adaptively for each class, but also harnesses class relationships that can further improve the performance.

3. We conduct extensive experiments to validate the performance of the proposed framework, and we achieved superior performance on two popular datasets.

The rest of this paper is organized as follows. Section 2 reviews and discusses related works. Section 3 describes the proposed multi-stream multi-class framework in detail. Experimental results and comparisons are discussed in Section 4, followed by conclusions in Section 5.

2. RELATED WORKS

As aforementioned, video classification has been extensively studied and significant efforts have been paid to design discriminative features or robust classifiers. We focus the review on recent works related to our proposed approach.

Hand-crafted Representations. There have been numerous works focusing on developing effective features that are expected to be robust to withstand intra-class variations and discriminative to separate different categories. For example, one can utilize image-based shape features, such as HOG and SIFT [33], to capture appearance information on individual frames. Different from frame-based features, motion features are designed to take the object movements into account, which is appealing since motion information is critical for understanding video contents. A popular way to obtain motion features is by extending frame-based local features into 3D space. For instance, Laptev *et al.* [31] extended the Harris detector into 3D space to find space-time interest points. Instead of locating interest points using 3D detectors, Wang *et al.* obtained better performance on video classification tasks by sampling patches densely [58]. In a later work, Wang *et al.* adopted the dense point trajectories, upon which several features are extracted from regions that are tracked with optical flow [56]. In addition, audio features are also adopted as a complement of the visual channel, among which the Mel-frequency cepstral coefficients (MFCC) is the most popular one.

CNN Representations. Motived by the promising results of deep networks (particularly the ConvNets) on image analysis tasks [51, 46, 14], several works have exploited deep architectures for video classification. Ji *et al.* extended CNN models into spatial-temporal space by operating on stacked video frames [22]. Karparthy *et al.* compared several architectures for action recognition [26]. Tran *et al.* proposed to learn generic spatial-temporal features which can be computed efficiently [53]. Xu *et al.* adopted advanced feature encoding strategies (*i.e.*, VLAD) to promote the generalization ability of CNN representations [65]. Zha *et al.* evaluated several options of using CNNs for event detection [70].

Simonyan and Zisserman [45] introduced an interesting two-stream approach, where two ConvNets are trained to explicitly capture spatial and short-term motion information using frames and stacked optical flows as inputs, respectively. Final predictions can be obtained by linearly averaging the prediction scores of the two ConvNets. A recent work by Wang *et al.* [59] combined the two-stream approach with the traditional dense trajectories [56] and reported strong results. In this paper, we also adopt two similar ConvNets as [45]. However, as the two-stream approach is not able to model the auditory and the long-term temporal clues, we adopt additional networks to build a more comprehensive framework. A novel fusion method is also proposed to combine the multi-stream outputs, which is better than the simple linear fusion used in [45].

Temporal Structure. Extensive works have been conducted to explore the temporal dynamics in videos. For example, Tang *et al.* introduced a HMM model to capture the changes of states for videos with variable durations [52]. Wang *et al.* combined feature templates with parts in a max-margin hidden CRF framework [61]. In addition to using graphical models, Fernando *et al.* proposed to train a linear ranking machine on the frames of a video, whose parameters will then be used to obtain a video-level representation [12]. Ramanathan introduced an unsupervised way to learn temporal embeddings using context information like word vectors [42]. More recently, RNN has been shown to be effective on many sequential modeling tasks, such as speech recognition [15] and image/video analysis [9, 67, 71]. For long-term temporal modeling of the video data, Srivastava *et al.* proposed an LSTM encoder-decoder framework to learn video representations in an unsupervised manner [48]. Donahua *et al.* [9] and Wu *et al.* [64] trained a two-layer LSTM network for action classification. Ng *et al.* [39] further demonstrated that a five-layer LSTM network is slightly better. Veeriah *et al.* proposed a differential gating scheme for LSTM to emphasize on the change in information gain [55]. More recently, Sharma *et al.* incorporated the attention mechanism into LSTM to identify the most relevant information (*e.g.*, objects) for recognizing actions [44].

Fusion. Videos naturally contain abundant clues, and hence a decent video categorization system often fuses multiple sources of information for improved recognition performance [69, 20]. The simplest fusion strategy is linear weighted fusion, which has been adopted in many recent approaches like [45]. Nandakumar *et al.* performed score fusion using a method called likelihood ratio test [36]. More recently, Xu *et al.* [66] and Ye *et al.* [68] proposed robust late fusion methods by seeking a low rank matrix to remove the noise of individually trained classifiers. Liu *et al.* [32] proposed to predict sample-specific weights in the fusion process. There are also a few works attempting to fuse multiple features with deep neural networks. Srivastava *et al.* proposed to combine features using deep Boltzmann Machines [49]. Neverova formulated the fusion problem as a modality dropping process [38]. Our paper relies on state-of-the-art deep models to characterize videos from different perspectives, and then performs a simple late fusion approach to further combine scores from multiple streams for final predictions.

Class Relationships. There are many studies using class relationships (a.k.a. context) to improve multi-class visual recognition performance. For instance, Rabinovich *et al.* utilized a Conditional Random Field (CRF) model to maximize object label agreement based on contextual relevance [41]. Jiang *et al.* used a semantic diffusion algorithm to incorporate class relationships [24]. Deng *et al.* proposed to jointly train a hierarchy and exclusion graph model with a ConvNet to learn class relations for image classification [8]. Assari *et al.* exploited class co-occurrences for improved video classification [2]. Wu *et al.* [63] proposed a regularized neural network to fuse features and explore class relationships, but used traditional hand-engineered features as inputs. Recently, Chen and Gupta utilized the class correlations in the form of confusion matrix to refine the classification scores from the softmax layer [6]. Note that Multi-Task Learning (MTL) also attempts to improve recognition performance by enforcing similar zero/nonzero patterns in the weight matrix through structural norms, which enables knowledge sharing among highly related tasks (classes). In our work, we adopt a data-driven approach, mining the knowledge learned by the models themselves, to borrow useful information from classes with relatedness.

3. THE PROPOSED APPROACH

In this section, we first describe the individual network streams and then introduce the proposed fusion method.

3.1 Multi-Stream ConvNets

Carrying abundant multimodal information, videos normally show the movements and interactions of objects under certain scenes over time, accompanied by human voices or background sounds. Therefore, video data can be naturally decomposed into spatial, motion and audio streams. The spatial stream consisting of individual frames depicts the static appearance information, while the motion stream captures object or scene movements demonstrated by continuous frames. In addition, sounds in the audio stream provide crucial clues that are often complementary to the visual counterpart. Motivated by the recent two-stream approach [45], we train three ConvNets to exploit the multimodal information, as described below.

In brief, the spatial ConvNet uses the raw frames as inputs, where we adopt a deep architecture with superior performance on image recognition tasks [46]. It can effectively recognize certain video semantics that have clear and discriminative appearance characteristics. For the motion stream, we train a ConvNet model operating on stacked optical flows following [45]. More specifically, through computing displacement vectors in both horizontal and vertical ways, the optical flows encode subtle motion patterns of objects between each pair of adjacent frames, which can be converted into two flow images as the inputs of the motion stream ConvNet. Previous studies have shown that further improvements can be obtained by stacking consecutive optical flow images in a short time window, owing to the inclusion of relatively more compact movements [45]. In order to leverage the audio information, we first apply the Short-Time Fourier Transformation to convert the 1-d soundtrack into a 2-D image (namely spectrogram) with the horizontal axis and vertical axis being time-scale and frequency-scale respectively. Then we employ a ConvNet to operate on the spectrograms as suggested in [54]. Notice that the ConvNet is well suited for modeling audio signals based on spectrograms with the weight sharing and max pooling mechanism to strive invariance of small frequency shifts [1].

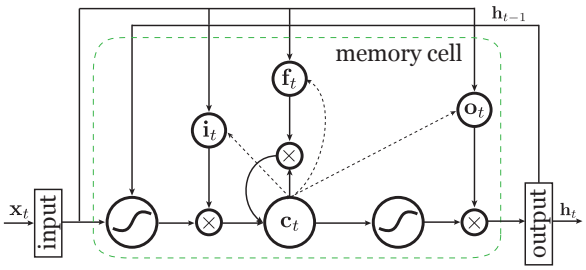

Figure 2: The structure of an LSTM unit.

3.2 Long Term Temporal Modeling

As the motion stream ConvNet only captures short-term motion patterns, we further employ LSTM [18] to model long-term temporal clues in the visual channel. LSTM is able to exploit temporal information of a data sequence with arbitrary length through recursively mapping the input sequence to output labels with hidden units. Each of the units maintains a built-in memory cell, which stores information over time guarded by several non-linear gate units to control the amount of changes and influence of the memory contents. To keep this paper self-contained, we briefly introduce LSTM as follows.

Figure 2 illustrates the typical structure of a hidden LSTM unit. In our framework, we denote $\mathbf{x}_t$ as the feature representation of a video frame or a stacked optical flow image at the t-th time step. Generally, an LSTM maps an input sequence $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$ to output labels $(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T)$ through computing activations of the units in the network recursively from $t = 1$ to $t = T$. At time t, the activation vectors of memory cell $\mathbf{c}_t$, output gate $\mathbf{o}_t$ and hidden state $\mathbf{h}_t$ are computed as:

$$\begin{aligned}
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c),\\
\mathbf{o}_t &= \sigma(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{W}_{co}\mathbf{c}_t + \mathbf{b}_o),\\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t),
\end{aligned} \tag{1}$$

where $\mathbf{W}_{xc}, \mathbf{W}_{hc}, \mathbf{W}_{xo}, \mathbf{W}_{ho}, \mathbf{W}_{co}$ are the weight matrices connecting two different units. $\mathbf{b}_c, \mathbf{b}_o$ are the bias terms, σ is the sigmoid function, and $\odot$ is an element-wise product operator. Notice that $\mathbf{i}_t$ and $\mathbf{f}_t$ are the activation vectors of input and forget gates, which are calculated with weight matrices as:

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{W}_{ci}\mathbf{c}_{t-1} + \mathbf{b}_i),\\
\mathbf{f}_t &= \sigma(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{W}_{cf}\mathbf{c}_{t-1} + \mathbf{b}_f).
\end{aligned} \tag{2}$$

From the above equations, the contents of the memory cell at the t-th time step $\mathbf{c}_t$ is computed as the weighted sum of the current inputs and the previous memory contents $\mathbf{c}_{t-1}$. The input and forget gates (*i.e.*, $\mathbf{i}_t$ and $\mathbf{f}_t$) impose regularization to determine whether to consider new information or forget old information. In addition, the output gate $\mathbf{o}_t$ controls the amount of information from the memory contents that is passed to the hidden state $\mathbf{h}_t$ to influence the computation in the next time step.

As a neural network, the LSTM model can be easily deepened by stacking the hidden states from a layer $l-1$ as inputs of the next layer l. In order to obtain the prediction scores for a total of C classes at a time step t, a softmax layer is placed on top of the last LSTM layer L to estimate the posterior probability p_c of the c-th class as:

$$p_c = \text{softmax}(\mathbf{h}_t^L) = \frac{\exp({\mathbf{u}_c}^T \mathbf{h}_t^L + b_c)}{\sum_{c' \in C} \exp({\mathbf{u}_{c'}}^T \mathbf{h}_t^L + b_{c'})}, \tag{3}$$

where $\mathbf{u}_c$ and b_c represent the corresponding weight vector and the bias term of the c-th class. Such an LSTM network can be trained using the Back-Propagation Through Time (BPTT) algorithm [16], which "unrolls" the model into a feed forward neural net and back-propagates to determine the optimal network parameters. We adopt the output from the last layer as the video-level prediction scores since this output is computed based on the information from the entire sequence. Our empirical results show that using the output of the last time step is better than pooling the predictions at all the time steps.

3.3 Multi-Stream Fusion

Given the prediction scores of the multiple deep network streams (each stream outputs scores of multiple classes), we are able to capture the video characteristics from different aspects. It is critical to effectively fuse the scores to generate the final predictions. Different semantic classes associate with the multiple streams with different strength. For example, some classes are strongly associated with particular objects which could be effectively recognized with the spatial stream, while others may contain dramatic movements so the short-term motion and the long-term temporal clues can contribute more significantly. Traditional fusion methods are usually performed in a uniform way without considering the class-specific preferences.

More formally, we denote the prediction scores from the m-th stream as $\mathbf{s}^m \in \mathbb{R}^C$ $(m = 1, \cdots, M)$ with C being the number of classes, and let $\hat{\mathbf{y}}$ be the final predicted labels. A straightforward way of late fusion is to compute the final prediction as $\hat{\mathbf{y}} = f(\mathbf{s}^1, \cdots, \mathbf{s}^M)$. Here f is a transition function, which can be a linear function, a logistic function, *etc.* However, such a late fusion approach treats all the classes uniformly and relies on the assumption that scores from multiple networks are explicitly complementary.

Different from the uniform fusion methods, we attempt to adaptively integrate the predictions from multiple streams to determine the optimal fusion weights for each class. To this end, we first stack the multiple score vectors of a training sample n as a coefficient vector:

$$\mathbf{s}_n = \left[{\mathbf{s}_n^1}^\top, \cdots, {\mathbf{s}_n^m}^\top, \cdots, {\mathbf{s}_n^M}^\top\right]^\top \in \mathbb{R}^{CM} \tag{4}$$

Then the best class-specific fusion weights can be learned with simple classifiers like logistic regression:

$$\mathbf{W} = \arg\min_{\mathbf{w}, \cdots, \mathbf{w}_C} \sum_{n,c} \log\left(1 + \exp\left[(1 - 2y_{n,c})\mathbf{s}_n^T \mathbf{w}_c\right]\right), \tag{5}$$

where $y_{n,c}$ is the ground-truth label of the n-th sample for class c, and $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_c, \cdots, \mathbf{w}_C] \in \mathbb{R}^{CM \times C}$. However, the final prediction score of class c in this process not only comes from its own scores of different streams, but also utilizes knowledge from other classes, which incurs extra parameters that will often lead to over-fitting.

3.4 Utilizing Class Relationships

To alleviate the over-fitting effect, one may use sparsity constraints (*e.g.*, ℓ_1 or ℓ_{21} norm) to force entries to be zero in

the weight matrix as a means of feature selection. For example, ℓ_1 norm penalizes non-zero weights, which will lead to the parameter vector to be sparse ignoring certain information in the input data. Nevertheless, instead of recognizing objects/concepts independently, humans can utilize the relations among concepts to derive a better understanding of the object of interest. Researchers also confirm that class relationships (a.k.a. semantic context) are known as helpful clues in multi-class visual recognition tasks and have been popularly used for improved performance [41, 4, 8]. Therefore, we believe that injecting such intrinsic class relationships in the learning process can improve the result.

Different from enforcing zero patterns in the weight matrix, in this paper, we tackle this problem by resorting to existing knowledge to determine the information from which classes and to what extent will be needed, and hence the parameters do not need to be learned from scratch. For example, we can adopt off-the-shelf WordNet/ConceptNet to obtain the relations among multiple concepts. However, these relations are handcrafted, which might be semantically similar but ignore the visual patterns. Note that each stream is a well-trained deep network, consisting of valuable information on data distribution. Therefore, we propose to mine knowledge (*i.e.*, class relationships) from the models themselves using the confusion matrix, which is a good indicator on how classes are related. More formally, we denote $\mathbf{V}^m \in \mathbb{R}^{C\times C}$ to be the similarity matrix of different classes for the m-th stream, and hence $\mathbf{V}^m_{ij}$ indicates the correlation between class C_i and C_j.

Confusion Matrix. We simply test the network on the validation set and adopt the confusion matrix to measure similarity among video classes:

$$\mathbf{V}^m_{ij} = \frac{1}{|C_i|}\sum_{n\in C_i} \mathbf{1}_{\arg\max_c(s^m_n)==C_j}. \tag{6}$$

where $\mathbf{1}(\cdot)$ is the indicator function, C_i is the collection of training samples that belongs to class i, and $|\cdot|$ is the cardinality function. Here each entry $\mathbf{V}_{ij}$ indicates the percentage of the samples with the ground-truth label of class i being wrongly classified into class j. We visualize the confusion matrix generated by the spatial ConvNet on CCV in Figure 3, which demonstrates how classes are correlated.

The reason of using a separate correlation matrix for each stream is that the captured class relationships in different streams are likely to be quite different. For instance, some classes are similar visually and some may share certain audio clues.

After obtaining the class relationship matrices of all the streams using the above equation, we stack the matrices $\mathbf{V} = \left[\mathbf{V}^1, \cdots, \mathbf{V}^m, \cdots, \mathbf{V}^M\right]^\top$ to constrain the weight learning process as:

$$\min_{\mathbf{W}} \quad L(\mathbf{S}, \mathbf{Y}; \mathbf{W}) + \lambda_1 \left\|\mathbf{W} - \mathbf{V}\right\|_F^2, \tag{7}$$

where the first term is the empirical loss defined in Equation 5. The second term constrains the fusion weights using the class correlation as a prior, with "F" indicating the Frobenius norm. For each similarity matrix $\mathbf{V}^m$, the non-diagonal entries demonstrate the similarities among different classes, which can be used to guide the weight learning process through borrowing information from highly related classes. Since the class relationship matrix $\mathbf{V}$ constrains which classes to be used and how much is needed, the best

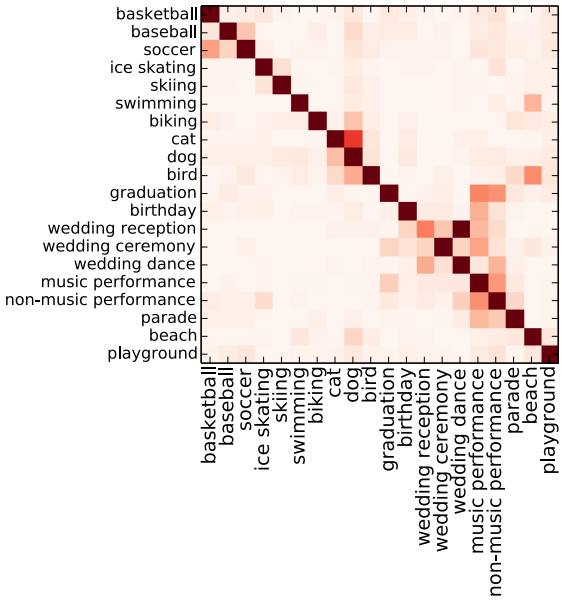

Figure 3: The confusion matrix of the spatial ConvNet on CCV.

optimal weights could be learned easier without suffering over-fitting. In addition, we also incorporate an ℓ_1 norm term to impose sparsity on the weight matrix, which, to some extent, can help avoid information sharing across irrelevant classes. Integrating all the terms, we have the following optimization problem:

$$\min_{\mathbf{W}} \quad L(\mathbf{S}, \mathbf{Y}; \mathbf{W}) + \lambda_1 \left\|\mathbf{W} - \mathbf{V}\right\|_F^2 + \lambda_2 \left\|\mathbf{W}\right\|_1. \tag{8}$$

Different from enforcing sparsity patterns as in standard MTL:

$$\min_{\mathbf{W}} \quad L(\mathbf{S}, \mathbf{Y}; \mathbf{W}) + \lambda_1 \left\|\mathbf{W}\right\|_p, \tag{9}$$

where the second term could be ℓ_1-norm, ℓ_2-norm or $\ell_{2,1}$-norm, our fusion approach relies on the class relationship matrix to constrain the weights for improved performance. As our approach not only fuses the multiple network streams but also utilizes class relationship, we name it as *multi-stream multi-class fusion*. The contribution from each component of the objective function will be evaluated later.

Although the loss function in Equation 8 is convex, it is non-trivial to solve it due to the non-smooth term. To tackle the optimization problem efficiently, we adopt the proximal gradient descent method that splits the objective function into a smooth part and a non-smooth part:

$$g = L(\mathbf{S}, \mathbf{Y}; \mathbf{W}) + \lambda_1 \left\|\mathbf{W} - \mathbf{V}\right\|_F^2, \tag{10}$$

$$h = \lambda_2 \left\|\mathbf{W}\right\|_1. \tag{11}$$

The update of $\mathbf{W}$ at the $k+1$ iteration can be simply computed as:

$$\mathbf{W}^{k+1} = \mathrm{Prox}_h(\mathbf{W}^k - \nabla g(\mathbf{W}^k)),$$

where Prox_h denotes the soft-thresholding operator for the ℓ_1 norm [10].

Note that the additional computational cost lies in the estimation of the proximal operator. Since it can be an-

alytically solved in linear time [3], the above optimization process is fairly efficient.

3.5 Implementation Details and Discussions

ConvNet Models. In this work, we adopt two ConvNet architectures, the CNN_M [45] model for capturing the short-term motion and the audio clues and a recent deeper VGG_19 architecture for the spatial stream [46]. The CNN_M is basically a variant of the AlexNet [28] with more filters included, which contains five convolutional layers followed by three fully connected layers. The VGG_19 not only reduces the size of the convolutional filters and the stride, but also extends the depth of the network to a total of 19 layers, equipping the architecture with the capacity of learning more robust representations. These two deep networks achieved 13.5% [45] and 7.5% [46] top-5 error rates on the ImageNet ILSVRC-2012 validation set, respectively. All the ConvNet models are trained using mini-batch stochastic gradient descent with a momentum fixed to 0.9. Our implementation is based on the publicly available Caffe toolbox [23] with some modifications. The input video frame is uniformly fixed to the size of 224×224. In addition, we also perform simple data augmentations like cropping and flipping following [45].

The spatial and the audio ConvNets are first pre-trained using the ILSVRC-2012 training set with 1.2 million images and then fine-tuned using the training video data. This strategy has been observed effective in [45] for the spatial stream, and we have observed it also helpful for the audio stream. To fine-tune the spatial and the audio ConvNets, we gradually decrease the learning rate from 10^{-3} to 10^{-4} after 14K iterations, then to 10^{-5} after 20K iterations. In addition, dropout is applied to the fully connected layers with a ratio of 0.5 to avoid over-fitting.

To train the motion ConvNet, we first compute optical flow using the GPU implementation of [5] and stack the optical flows in each 10-frame window to receive a 20-channel optical flow image as the input (one horizontal channel and one vertical channel for each frame pair). Unlike the spatial and the audio ConvNets, we train the motion ConvNet from scratch by adopting 0.7 dropout ratio and setting the learning rate to 10^{-2} initially, which is reduced to 10^{-3} after 100K iterations and then to 10^{-4} after 200K iterations. Note that we also tried to use the VGG_19 network to train the motion ConvNet, but observed worse results as the network contains much more parameters that cannot be well-tuned using the limited training video data.

LSTM. We adopt the two-layer LSTM model proposed by Graves [16] for temporal modeling. Two models are trained with features extracted from the first fully-connected layer of the spatial and the motion ConvNets respectively as inputs. Each LSTM has 1,024 hidden units in the first layer and 512 hidden units in the second layer. We utilize a parallel implementation of the BPTT algorithm with a mini-batch size of 10 to train the network weights, where the learning rate and momentum are set as 10^{-4} and 0.9. In addition, we set the maximal training iterations to be 150K. Note that, in this paper, we focus on a multi-stream framework by utilizing the audio signal as a single stream for video classification. Further decomposing the audio track into multiple segments to extract more detailed temporal audio dynamics is feasible.

Fusion. As shown in Equation 8, the proposed fusion method seeks a tradeoff among the three terms. We uniformly fix λ_2 to be 10^{-3} to encourage sparsity in the learned weight matrix. The parameter λ_1 is selected among $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$ using cross-validation.

Discussions. Our proposed framework has the capability of modeling video data comprehensively by adaptively fusing audio, static spatial, short-term motion and long-term temporal clues. As described above, such a framework consists of multiple separately trained deep networks. Although being feasible to jointly train the entire framework, it is complicated and computationally demanding. A recent work performing joint training of the LSTM with a ConvNet improves the results on the UCF-101 benchmark from 70.5% (separate network training) to 71.1% [9], which is not very significant. In addition, training multiple deep networks separately makes the approach more flexible, where a component may be replaced without the need of re-training the entire framework. For instance, one can utilize more discriminative ConvNet models like the GoogLeNet [51] and deeper RNN models [7] to replace the current ConvNet and LSTM parts respectively for better performance. Therefore, in this work, we focus on presenting a general framework for video classification. With the proposed multi-stream multi-class fusion method, the framework is empirically proved to be effective for the video classification task, as discussed in the following section.

4. EXPERIMENTS

In this section, we report results on two popular datasets. Experiments are designed to study the effectiveness of each individual stream and the proposed fusion method.

4.1 Experimental Setup

Datasets and Evaluation Measures. UCF-101 [47] is a widely adopted dataset for human action recognition, containing 13,320 video clips annotated into 101 action classes. All the video clips have a fixed frame rate of 25 fps with a spatial resolution of 320 × 240 pixels. This dataset is challenging because most videos were captured under uncontrolled environments with camera motion, cluttered backgrounds and large intra-class variations. We follow the suggested experimental protocol and report mean accuracy over the three training and test splits [19].

The Columbia Consumer Videos (CCV) dataset [25] contains 9,317 YouTube videos and 20 classes. Most of the classes are events like "basketball", "graduation ceremony" and "wedding dance". A few are scenes and objects like "beach" and "dog". Following [25], we adopt the suggested training and test split and compute the average precision (AP) for each class. Mean AP (mAP) is used to measure the overall performance on this dataset.

The two datasets possess very different characteristics. Besides the difference of the defined semantic classes, the average video duration of CCV is 80 seconds, which is around ten times longer than that of UCF-101. Testing on these two datasets is helpful for evaluating the effectiveness and the generalization capability of our multi-stream classification approach.

Compared Methods. To validate the effectiveness of our multi-stream multi-class fusion method, we compare with the following alternatives: (1) Average Fusion, where the mean scores of multiple network streams are used as the final prediction; (2) Weighted Fusion, where the scores are

fused linearly with weights estimated by cross-validation; (3) Kernel Average Fusion, where the scores are used as features and kernels computed from different network scores are averaged to train an SVM classifier; (4) Multiple Kernel Learning (MKL) Fusion, where the kernels are combined using the ℓ_p-norm MKL algorithm [27]; (5) Logistic Regression Fusion, where a logistic regression model is trained to estimate the fusion weights; (6) Domain Adaptive Semantic Diffusion (DASD) [24], which uses a graph diffusion formulation for context-based multi-class score fusion.

4.2 Results and Discussions

4.2.1 Multi-Stream Networks

We first report the performance of each individual stream on both datasets. After that, average fusion is adopted to study whether two or more streams are complementary. The proposed fusion method will be evaluated later.

Table 1 reports the results. Comparing the top two cells of results on UCF-101, it is interesting to observe that the spatial LSTM outperforms the spatial ConvNet and the motion LSTM is also comparable to the motion ConvNet. This is largely due to the fact that the long-term temporal clues are fully discarded in the ConvNet based classification, which can be exploited by the LSTM.

On the CCV dataset, the ConvNet achieves significantly better results than the LSTM on both spatial and motion streams. This is because the classes in CCV are either high-level events or objects/scenes. Compared with human actions, the temporal clues of these classes are more obscure and thus difficult to be captured. Also, the CCV videos are temporally untrimmed, which may contain significant portions of contents irrelevant to the classes, making the temporal modeling task even more difficult.

One may notice that our motion networks perform worse than spatial networks, which is not consistent with the observations in dense trajectory features [56]. We would like to underline that motion information extracted from optical flow images tends to be noisy, especially on temporally untrimmed long videos, as reported in [39]. In addition, the dense trajectory features are hand-crafted without the need of training, while the motion stream networks demand extensive training. Unfortunately, labeled training data in the video domain is still quite limited. We expect to obtain better results from the motion stream network once sufficient training data is available.

The audio ConvNets operated on spectrograms produce 16.2% on UCF-101 and 21.5% on CCV. Note that only 51 classes in UCF-101 have audio signals, and the performance on the 51-class subset is actually 32.1%. The audio stream is much worse than the spatial and the motion streams on both datasets, confirming that the visual channel are more informative than the audio counterpart.

Next, we evaluate the combinations of multiple networks to study whether fusion can compensate the limitations of a single stream in describing complex video data. The simple average fusion is adopted. Results are summarized in the bottom three groups of Table 1. We first assess the gain from integrating the spatial and the motion information modeled by ConvNet and LSTM respectively. On UCF-101, significant improvements (about 6% for ConvNet and 3% for LSTM) are observed over the best single stream results. The gain on CCV is consistent but not as significant as that on UCF-101, indicating that the short-term motion is more critical for human action analysis. Note that the average fusion of the spatial and the motion ConvNets follows the same idea of the two-stream approach proposed in [45]. Our implementation of this approach produces slightly worse performance than that originally reported in [45] (86.2% vs. 88.0%).

	UCF-101	CCV
Spatial ConvNet	80.1	75.0
Motion ConvNet	77.5	58.9
Spatial LSTM	83.3	43.3
Motion LSTM	76.6	54.7
Audio ConvNet	16.2*	21.5
ConvNet (spatial+motion)	86.2	75.8
LSTM (spatial+motion)	86.3	61.9
ConvNet+LSTM (spatial)	84.0	77.9
ConvNet+LSTM (motion)	81.4	70.9
ConvNet+LSTM (spatial+motion)	90.1	81.7
All the streams	90.3	82.4

Table 1: Performance of each individual stream and their average fusion (indicated by "+"). *Note that the videos of only 51 classes in UCF-101 contain audio soundtracks. The audio ConvNet can produce an accuracy of 32.1% on the 51-class subset.

We also fuse ConvNet with LSTM separately on both streams to investigate the contribution of the long-term temporal modeling. Overall, we observe very consistent improvements on both datasets. In particular, on CCV, although the individual LSTM model is worse than ConvNet, the combination of them leads to significant improvements. Especially, a gain of nearly 12% is obtained on the motion stream. These results show that the long-term temporal clues are highly complementary to the ConvNet-based predictions, even in the case of modeling complex contents in the long CCV videos.

Finally, the combination of ConvNet and LSTM on both streams, indicated by "ConvNet+LSTM (spatial+motion)", achieves 90.1% and 81.7% on UCF-101 and CCV respectively. Further adding the audio ConvNet ("all the streams") can improve the results particularly on CCV which contains many classes that can be partly revealed by auditory clues (*e.g.*, cheering sounds in the sports events). In summary, the fusion results clearly demonstrate that all the multimodal clues in our approach are useful and should be adopted in a successful video classification system.

4.2.2 Multi-Stream Multi-Class Fusion

In this subsection, we evaluate the proposed fusion method and compare it with the alternative methods. Table 2 gives the results. We see that all the methods produce better results than the individual streams. The simple average fusion and weighted fusion are slightly better than the learning based kernel fusion and logistic regression fusion, indicating that the learning based methods are prone to overfitting. Kernel average fusion shows slightly better results than MKL, which is consistent with the observations in several previous studies like [13]. DASD produces similar results to weighted fusion as it is essentially an iterative weighted fusion method.

	UCF-101	CCV
Average fusion	90.3	82.4
Weighted fusion	90.6	82.7
Kernel average fusion	90.2	82.1
MKL fusion	89.6	81.8
Logistic regression fusion	89.8	82.0
DASD	90.4	82.9
Multi-stream fusion (λ_1=0)	90.9	82.8
Multi-stream multi-class fusion (λ_2=0)	91.6	83.7
Multi-stream multi-class fusion (-A)	92.2	84.0
Multi-stream multi-class fusion	92.6	84.9

Table 2: Comparison of fusion methods. "-A" indicates that the audio stream ConvNet is not adopted. See texts for discussions.

Our proposed multi-stream multi-class fusion (the bottom group of results) outperforms all the alternatives with clear margins. To investigate the contributions of the class relationship term and the sparsity term in our approach, we set λ_1 and λ_2 to be zero respectively. As can be seen, both terms are very useful. Relatively, the class relationship (λ_1) plays a more important role than the sparsity term (λ_2). This corroborates the effectiveness of using the class relationship, even when it is roughly estimated based on prediction score correlations (see Section 3.3), which is very appealing. The two terms are complementary as the sparsity inducing norm further enhances robustness by alleviating incorrect information sharing. Note that when eliminating both terms, our fusion approach degenerates to the standard logistic regression fusion. In summary, these results show that it is helpful to fuse the outputs of both multiple network streams and multiple classes.

The contribution of the audio clues is similar on both datasets ("-A" indicates the same approach without using the audio ConvNet). Audio improves just 0.4% on UCF-101 because only half of the video clips contain soundtracks. Figure 4 further shows the per-class performance on CCV, where we can see that fusion leads to very consistent and significant improvements for all the classes.

Comparisons with different regularizers. Since sparsity regularizers are popular constraints to improve generalization ability as in standard MTL, we also compare the proposed approach with this line of work by replacing the proposed regularizers with alternative ones (see Eq. 9). The results are summarized in Table 3. As we can see from the table, the proposed regularization norm that utilizes class relationship as contextual information outperforms the competing regularizers. Different from forcing entries to be zero (*i.e.*, ℓ_1 norm) or selecting the same set of parameters for related tasks (*i.e.*, ℓ_{21} norm), the proposed norm leverages the class relationships in the models to guide the fusion process, through constraining which and how much information is required from other classes. This corroborates the fact the scores from the softmax layer contain meaningful information rather than simply indicating the predicted label [17].

4.2.3 Computational Efficiency

The proposed approach can achieve effective recognition efficiently. For a CCV video clip with an average duration of 80 seconds, extracting the improved dense trajectories requires 850 seconds, while it only takes 131 seconds for our method to finish the entire process, which is evaluated on a single NVIDIA Telsa K40 GPU.

	UCF-101	CCV
Fusion with ℓ_1 norm	90.9	82.8
Fusion with ℓ_{21} norm	91.0	82.4
Fusion with (ℓ_{21} + ℓ_1) norm	91.4	83.2
Multi-stream multi-class fusion	92.6	84.9

Table 3: Comparison with different regularization strategies.

4.2.4 Comparison with State of the Arts

UCF-101		CCV	
Donahue *et al.* [9]	82.9	Lai *et al.* [29]	43.6
Srivastava *et al.* [48]	84.3	Jiang *et al.* [25]	59.5
Wang *et al.* [57]	85.9	Xu *et al.* [66]	60.3
Tran *et al.* [53]	86.7	Ma *et al.* [34]	63.4
Simonyan *et al.* [45]	88.0	Jhuo *et al.* [21]	64.0
Ng *et al.* [39]	88.6	Ye *et al.* [68]	64.0
Lan *et al.* [30]	89.1	Liu *et al.* [32]	68.2
Zha *et al.* [70]	89.6	Wu *et al.* [64]	83.5
Wang *et al.* [59]	91.5	Nagel *et al.* [35]	71.7
Wang *et al.* [60]	92.4		
Ours (-A)	92.2	Ours (-A)	84.0
Ours	**92.6**	Ours	**84.9**

Table 4: Comparison with state-of-the-art results. Our approach produces to-date the highest reported results on both datasets. "Ours (-A)" indicates the same framework without using the audio stream ConvNet.

We compare our approach with the state of the arts on both datasets. Results are listed in Table 4. Our proposed multi-stream approach achieves the highest performance on both datasets. On UCF-101, many works with competitive results are based on the hand-engineered dense trajectory features [57, 30], while our approach fully relies on the deep networks. Compared with the original result of the two-stream approach [45], our approach captures a more comprehensive set of useful clues with a more effective fusion method. Zha *et al.* [70] combined the ConvNet features with the dense trajectories [56] to achieve competitive results. The previous best performance on UCF-101 is from Wang *et al.* [59], who combined the two-stream approach [45] and the dense trajectories. Note that a gain of 1% on the widely adopted UCF-101 dataset is generally considered as a significant progress.

In addition, the recent works in [9, 48, 64, 39] also adopted the LSTM to model the temporal clues for video classification and reported promising performance, but did not explore the audio stream nor employ advanced fusion strategies.

On the CCV dataset, all the recent approaches were developed based on multiple features, either the hand-engineered descriptors or the ConvNet-based representations. Our approach produces better results than all of them.

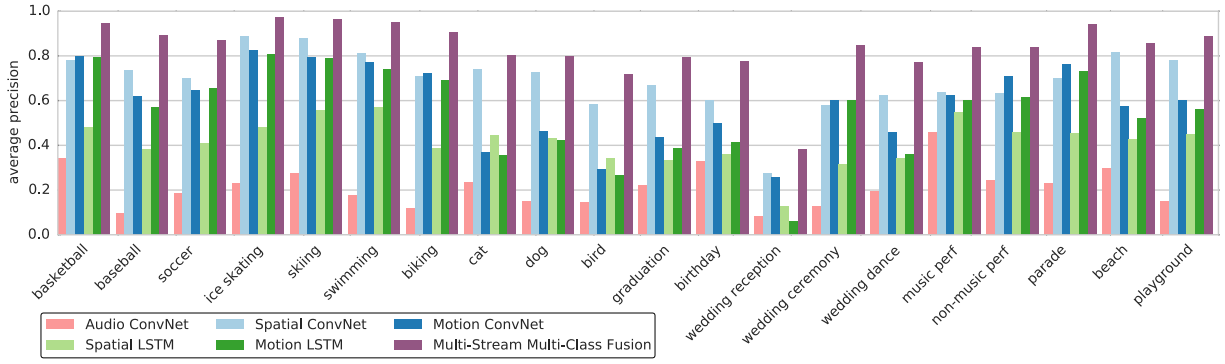

Figure 4: Per-class performance on CCV. Multi-stream multi-class fusion of the deep network outputs produces consistently better results than the individual streams on all the classes.

5. CONCLUSIONS

We have presented a multi-stream framework of deep networks for video classification. The framework harnesses multimodal features that are more comprehensive than those previously adopted. Specifically, standard ConvNets are applied to audio spectrograms, visual frames and stacked optical flows to exploit the audio, spatial and short-term motion clues in videos, respectively. LSTM is further adopted on the spatial and the short-term motion features from the ConvNets for long-term temporal modeling. The outputs from the different streams are then combined using a novel method called multi-stream multi-class fusion, which not only learns the best weights of the multi-stream networks for each class, but also considers class relationships for improved performance. Our results confirm that all the adopted streams are effective for modeling both simple human actions in short clips and complex events in temporally untrimmed Internet videos. Combining the multi-stream multi-class predictions by our proposed fusion method consistently outperforms peer approaches on two popular benchmarks.

This paper is among the limited number of studies showing strong video classification performance using deep networks. As aforementioned, unlike the spatial ConvNet that can be trained by fine-tuning a model pre-trained on the ImageNet dataset, the motion ConvNet has to be trained from scratch on videos. Therefore, one promising future direction is to pre-train the motion ConvNet using large video datasets like the Sports-1M [26], which may lead to much better results.

Acknowledgment

This work was supported by a China's National 863 Program (#2014AA015101), and two grants from National Natural Science Foundation of China (#61572138 and #U1509206).

6. REFERENCES

[1] O. Abdel-Hamid, L. Deng, and D. Yu. Exploring convolutional neural network structures and optimization techniques for speech recognition. In *INTERSPEECH*, 2013.

[2] S. M. Assari, A. R. Zamir, and M. Shah. Video classification using semantic concept co-occurrences. In *CVPR*, 2014.

[3] F. Bach, R. Jenatton, J. Mairal, G. Obozinski, et al. Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning*, 2011.

[4] 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. *CoRR*, 2013.

[5] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*. 2004.

[6] X. Chen and A. Gupta. Webly supervised learning of convolutional networks. In *ICCV*, 2015.

[7] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio. Gated feedback recurrent neural networks. *CoRR*, 2015.

[8] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam. Large-scale object classification using label relation graphs. In *ECCV*, 2014.

[9] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[10] D. L. Donoho and I. M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the american statistical association*, 1995.

[11] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE TPAMI*, 2013.

[12] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars. Modeling video evolution for action recognition. In *CVPR*, 2015.

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

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[15] A. Graves, A. Mohamed, and G. E. Hinton. Speech recognition with deep recurrent neural networks. In *ICASSP*, 2013.

[16] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 2005.

[17] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *CoRR*, 2015.

[18] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 1997.

[19] H. Idrees, A. R. Zamir, Y.-G. Jiang, A. Gorban, I. Laptev, R. Sukthankar, and M. Shah. The thumos challenge on action recognition for videos" in the wild". *arXiv preprint arXiv:1604.06182*, 2016.

[20] M. Jain and et al. University of amsterdam at thumos 2015. In *CVPR THUMOS Workshop*, 2015.

[21] 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.

[22] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. In *ICML*, 2010.

[23] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM Multimedia*, 2014.

[24] Y.-G. Jiang, J. Wang, S.-F. Chang, and C.-W. Ngo. Domain adaptive semantic diffusion for large scale context-based video annotation. In *ICCV*, 2009.

[25] 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.

[26] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[27] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien. Lp-norm multiple kernel learning. *The Journal of Machine Learning Research*, 2011.

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

[29] K.-T. Lai, F. X. Yu, M.-S. Chen, and S.-F. Chang. Video event detection by inferring temporal instance labels. In *CVPR*, 2014.

[30] Z. Lan, M. Lin, X. Li, A. G. Hauptmann, and B. Raj. Beyond gaussian pyramid: Multi-skip feature stacking for action recognition. *CoRR*, 2014.

[31] I. Laptev. On space-time interest points. *IJCV*, 2007.

[32] 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.

[33] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[34] A. J. Ma and P. C. Yuen. Reduced analytic dependency modeling: Robust fusion for visual recognition. *IJCV*, 2014.

[35] M. Nagel, T. Mensink, and C. G. M. Snoek. Event fisher vectors: Robust encoding visual diversity of visual streams. In *BMVC*, 2015.

[36] K. Nandakumar, Y. Chen, S. C. Dass, and A. K. Jain. Likelihood ratio-based biometric score fusion. *IEEE TPAMI*, 2008.

[37] 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.

[38] N. Neverova, C. Wolf, G. Taylor, and F. Nebout. Moddrop: adaptive multi-modal gesture recognition. *IEEE TPAMI*, 2014.

[39] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015.

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

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

[42] V. Ramanathan, K. Tang, G. Mori, and L. Fei-Fei. Learning temporal embeddings for complex video analysis. In *ICCV*, 2015.

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

[44] S. Sharma, R. Kiros, and R. Salakhutdinov. Action recognition using visual attention. *CoRR*, 2015.

[45] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

[46] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[47] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, 2012.

[48] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using LSTMs. In *ICML*, 2015.

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

[50] X. Sun, M. Chen, and A. Hauptmann. Action recognition via local descriptors and holistic features. In *CVPR*, 2009.

[51] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper with Convolutions. In *CVPR*, 2015.

[52] K. Tang, L. Fei-Fei, and D. Koller. Learning latent temporal structure for complex event detection. In *CVPR*, 2012.

[53] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. C3d: Generic features for video analysis. *CoRR*, 2014.

[54] A. Van den Oord, S. Dieleman, and B. Schrauwen. Deep content-based music recommendation. In *NIPS*, 2013.

[55] V. Veeriah, N. Zhuang, and G.-J. Qi. Differential recurrent neural networks for action recognition. In *ICCV*, 2015.

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

[57] H. Wang and C. Schmid. Lear-inria submission for the thumos workshop. In *ICCV THUMOS Workshop*, 2013.

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

[59] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, 2015.

[60] X. Wang, A. Farhadi, and A. Gupta. Actions ˜ transformations. In *CVPR*, 2016.

[61] Y. Wang and G. Mori. Max-margin hidden conditional random fields for human action recognition. In *CVPR*, 2009.

[62] Z. Wu, Y. Fu, Y.-G. Jiang, and L. Sigal. Harnessing object and scene semantics for large-scale video understanding. In *CVPR*, 2016.

[63] Z. Wu, Y.-G. Jiang, J. Wang, J. Pu, and X. Xue. Exploring inter-feature and inter-class relationships with deep neural networks for video classification. In *ACM Multimedia*, 2014.

[64] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In *ACM Multimedia*, 2015.

[65] Z. Xu, Y. Yang, and A. G. Hauptmann. A discriminative cnn video representation for event detection. In *CVPR*, 2015.

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

[67] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. In *ICCV*, 2015.

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

[69] S.-I. Yu, L. Jiang, and et al. Informedia@ trecvid 2014 med and mer. In *NIST TRECVID Video Retrieval Evaluation Workshop*, 2014.

[70] S. Zha, F. Luisier, W. Andrews, N. Srivastava, and R. Salakhutdinov. Exploiting image-trained cnn architectures for unconstrained video classification. In *BMVC*, 2015.

[71] H. Zhang, M. Mang, R. Hong, L. Nie, and T.-S. Chua. Play and rewind: Optimizing binary representations of videos by self-supervised temporal hashing. In *ACM Multimedia*, 2016.