Fusing Multi-Stream Deep Networks for Video Classification

Zuxuan Wu, Yu-Gang Jiang, Hao Ye, Xi Wang, Xiangyang Xue
Fudan University

zxwu, ygj, haoye10, xwang10, xyxue@fudan.edu.cn

Jun Wang
East China Normal University

wongjun@gmail.com

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, we propose a simple and effective fusion method to generate the final predictions, where the optimal fusion weights are learned adaptively for each class, and the learning process is regularized by automatically estimated class relationships. Our contributions are two-fold. First, the proposed multi-stream framework is able to exploit multimodal features that are more comprehensive than those previously attempted. Second, we demonstrate that the adaptive fusion method using the class relationship as a regularizer outperforms traditional alternatives that estimate the weights in a "free" fashion. 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.

1. Introduction

The problem of video classification based on semantic contents like human actions or complex events has been extensively studied in the computer vision community. The fact that videos are intrinsically multimodal demands 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 [28], the Mel-Frequency Cepstral Coefficients (MFCC) [47], the STIP [26] and the dense trajectories [44], which can be encoded into video-level representations by bag-of-words (BoW) [40, 49, 31] or Fisher vectors (FV) [33, 35, 25, 50].

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
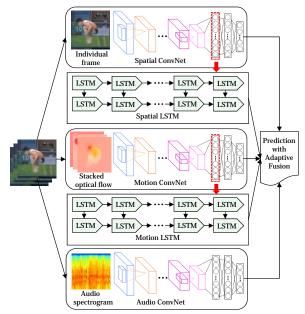

Figure 1. Illustration of the proposed multi-stream video classification framework.

domains. In particular, the convolutional neural networks (ConvNets) are very successful on image analysis tasks like object detection [13], object recognition [37, 41] and image segmentation [11]. However, for video classification, most deep network based approaches (*e.g.*, [18, 21, 36]) demonstrated worse or similar results to the hand-engineered features [44]. 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 [18, 21, 36] 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 [32, 46]. Nevertheless,

arXiv:1509.06086v2 [cs.CV] 11 Nov 2015

the audio information has rarely been exploited. In addition, most existing approaches fused the outputs of multiple networks in a very straightforward way [36], which could lead to sub-optimal performance.

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. We propose to regularize the weight learning process using class relationships estimated without using additional labels. This helps inject class context into the final predictions and thus can significantly improve the results. 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 that the multi-stream networks are able digest complementary information to receive significantly improved performance.

2. We propose a simple and effective fusion method to combine the outputs of the individual networks. The method learns fusion weights adaptively for each class and is able to harness class relationships in the weight learning process. We empirically show that the class relationship regularizer is very effective.

Incorporating the fusion method into the proposed multi-stream framework, we achieve superior performance on two popular benchmark datasets.

2. Related Works

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

Motived by the promising results of deep networks (particularly the ConvNets) on image analysis tasks [41, 37, 13], 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 [18]. Karparthy *et al.* compared several architectures for action recognition [21]. Tran *et al.* proposed to learn generic spatial-temporal features which can be computed efficiently [42]. Simonyan and Zisserman [36] 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. In this paper, we also adopt two similar ConvNets as [36]. 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 [36].

The RNN has been shown to be effective on many sequential modeling tasks, such as speech recognition [14] and image/video description [9, 48]. 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 [39]. Donahua *et al.* [9] and Wu *et al.* [46] trained a two-layer LSTM network for action classification. Ng *et al.* [32] further demonstrated that a five-layer LSTM network is slightly better.

Fusion is needed to combine the outputs of separate prediction models. The simplest solution is linear weighted fusion, which has been adopted in many recent approaches like [36]. Nandakumar *et al.* performed score fusion using a method called likelihood ratio test [30]. More recently, Xu *et al.* [47] and Ye *et al.* [49] proposed robust late fusion methods by seeking a low rank matrix to remove the noise of individually trained classifiers. Liu et al. [27] proposed to predict sample-specific weights in the fusion process.

There are many studies using context or class relationships to improve visual recognition performance. For instance, Rabinovich *et al.* utilized a Conditional Random Field (CRF) model to maximize object label agreement based on contextual relevance [34]. 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 video classification [2]. Different from these works, we use class relationship as a regularizer to learn fusion weights adaptively for each class.

3. Methodology

In this section, we first describe the individual streams and then introduce the proposed adaptive multi-stream fusion method, followed by implementation details.

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 [36], 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 [37]. 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 [36]. 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 [36]. 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 [43]. 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].

3.2. Long Term Temporal Modeling

As the motion stream ConvNet only captures short-term motion patterns, we further employ LSTM [16] to model long-term temporal clues in the visual channel. LSTM is a popular RNN model that incorporates memory cells with several gates to learn long-term dependencies without suffering from vanishing and exploding gradients as the traditional RNNs [5]. It is able to exploit temporal information of a data sequence with arbitrary length through recursively mapping the input sequence to output labels with hidden LSTM 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.
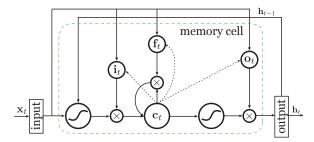

Figure 2. The structure of an LSTM unit.

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:

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

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:

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

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 [15], 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 last layer output is better than pooling the predictions at all the time steps.

3.3. Adaptive Multi-Stream Fusion

Given the prediction scores of multiple network streams (*i.e.*, the ConvNets and the LSTM), we are able to capture the video characteristics from different aspects. It is critical to effectively fuse the multi-stream 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 at the stream-level without considering the class-specific preferences. In addition, most existing studies on model fusion neglected the class relationships that can serve as complementary information for improved performance [34, 4, 8]. In the following we introduce the proposed adaptive multi-stream fusion method, which is able to determine the optimal fusion weights adaptively for each class. The highly correlated classes are also automatically identified and their relationships are utilized in the method.

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 without considering their different characteristics.

Different from the uniform fusion methods, we attempt to adaptively integrate the predictions from multiple streams for each class by not only combining scores across streams but also utilizing class knowledge as a prior to provide additional information. To this end, we first stack the multiple score vectors of a training sample n as a coefficient vector: $\mathbf{s}_n = \left[\mathbf{s}_n^{1\top}, \cdots, \mathbf{s}_n^{m\top}, \cdots, \mathbf{s}_n^{M\top} \right]^\top \in \mathbb{R}^{CM}$. Then the best class-specific fusion weights can be learned

with logistic regression as:

$$\mathbf{W} = \arg \min_{\mathbf{w}, \cdots, \mathbf{w}_C} \sum_{n,c} \log \left(1 + \exp \left[(1 - 2y_{n,c}) \mathbf{s}_n^T \mathbf{w}_c \right] \right), \tag{4}$$

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, direct optimization with the above formulation often leads to over-fitting and produces limited performance on the test set. In order to alleviate this and take the class relationships into account, we use the relationships as a prior to guide the learning of the weights. More precisely, we first compute a correlation matrix $\mathbf{V}^m \in \mathbb{R}^{C \times C}$ of the classes for the m-th stream using the corresponding prediction scores, where 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. The reason of using separate correlation matrix for each stream is that the captured class relationships in different streams are likely to be quite different. Next, we stack the similarity matrices of all the streams $\mathbf{V} = \left[\mathbf{V}^1, \cdots, \mathbf{V}^m, \cdots, \mathbf{V}^M \right]^\top$ to regularize the weight learning process as:

$$\min_{\mathbf{W}} \quad L(\mathbf{S}, \mathbf{Y}; \mathbf{W}) + \lambda_1 \|\mathbf{W} - \mathbf{V}\|_F^2, \tag{5}$$

where the first term is the empirical loss that measures the discrepancy between the ground-truth labels $\mathbf{Y}$ and the prediction scores $\mathbf{S}$, and the second term regularizes the fusion weights using the class correlation as a prior. 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.

In addition, we also incorporate an ℓ_1 norm regularization to impose sparsity on the weight matrix, which, to some extent, can help avoid information sharing from irrelevant classes. With both regularization terms, we have following optimization problem:

$$\min_{\mathbf{W}} \quad L(\mathbf{S}, \mathbf{Y}; \mathbf{W}) + \lambda_1 \|\mathbf{W} - \mathbf{V}\|_F^2 + \lambda_2 \|\mathbf{W}\|_1. \tag{6}$$

In summary, by treating the class correlation matrix as a prior, our fusion approach minimizes an empirical loss regularized by a sparsity constraint to effectively derive class adaptive fusion weights.

Although the loss function in Equation 6 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:

$$g = L(\mathbf{S}, \mathbf{Y}; \mathbf{W}) + \lambda_1 \|\mathbf{W} - \mathbf{V}\|_F^2, \tag{7}$$

$$h = \lambda_2 \|\mathbf{W}\|_1. \tag{8}$$

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

3.4. Implementation Details and Discussions

ConvNet Models. In this work, we adopt two ConvNet architectures, the CNN_M [36] model for capturing the short-term motion and the audio clues and a recent deeper VGG_19 [37] architecture for the spatial stream. The CNN_M is basically a variant of the AlexNet [23] 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% [36] and 7.5% [37] 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 [19] 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 [36].

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 [36] 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 [6] 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 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 [15] for temporal modeling. Two models are trained with features extracted respectively from the first fully-connected layer of the spatial and the motion ConvNets 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 6, the proposed adaptive fusion strategy seeks a tradeoff between the empirical loss and the two regularization terms. We uniformly fix λ_2 to be 10^{-3} to encourage sparsity in the weight matrix. The parameter λ_1 is selected among $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$ using cross-validation.

Discussions. The proposed multi-stream 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 [41] 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 multi-stream video classification. With the proposed adaptive fusion method, such a multi-stream 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 adaptive multi-stream fusion method.

4.1. Experimental Setup

Datasets and Evaluation Measures. UCF-101 [38] 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.

The Columbia Consumer Videos (CCV) dataset [20] 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 [20], 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.

Alternative Fusion Methods. To validate the effectiveness of our adaptive multi-stream fusion method, we compare with the following alternatives: (1) Average Fusion, where the mean scores of multiple networks 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 [22]; (5) Logistic Regression Fusion, where a logistic regression model is trained to estimate the fusion weights.

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 adaptive 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,

	UCF-101	CCV
Spatial ConvNet	80.4	75.0
Motion ConvNet	78.3	59.1
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 only the videos of 51 classes in UCF-101 contain audio soundtracks. The audio ConvNet can produce an accuracy of 32.1% on the 51-class subset.

which contain valuable information that 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.

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 [36]. Our implementation of this approach produces slightly worse performance than that originally reported in [36] (86.2% vs. 88.0%).

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, which is fairly appealing.

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 Adaptive Multi-Stream Fusion

In this subsection, we evaluate the proposed adaptive multi-stream fusion approach, 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 learning to fuse the prediction scores in a "free" manner is prone to over-fitting. Kernel average fusion shows slightly better results than MKL, which is consistent with the observations in several previous studies like [12].

Our proposed adaptive multi-stream fusion (the bottom row) outperforms all the alternatives with clear margins. To investigate the contributions of the two regularizers in our approach, we set λ_1 and λ_2 to be zero respectively. As can be seen, the class relationship regularizer ($\lambda_2 = 0$) plays a more important role than the sparsity regularizer ($\lambda_1 = 0$). This corroborates the effectiveness of using the class relationships, which not only brings in useful contextual information but also helps prevent over-fitting. The two regularizers are complementary as the sparsity inducing norm further enhances robustness by alleviating incorrect information sharing. Note that when eliminating both regularizers, our fusion approach degenerates to the standard logistic regression fusion.

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 sound-

	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
Adaptive multi-stream fusion (λ_1=0)	90.9	82.8
Adaptive multi-stream fusion (λ_2=0)	91.6	83.7
Adaptive multi-stream fusion (-A)	92.2	84.0
Adaptive multi-stream 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.

tracks. Overall, the gain from the adaptive multi-stream fusion is more significant on UCF-101 as it has more classes for semantic sharing. Figure 3 further shows the per-class performance on CCV, where we can see that the fusion leads to very consistent and significant improvements for all the classes.

4.2.3 Comparison with State of the Arts

We compare our approach with the state of the arts on both datasets. Results are listed in Table 3. 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 [45, 25], while our approach fully relies on the deep networks. Compared with the original result of the two-stream approach [36], our approach captures a more comprehensive set of useful clues with a more effective fusion strategy. Note that a gain of even just 1% on the widely adopted UCF-101 dataset is generally considered as a significant progress.

In addition, the recent works in [9, 39, 46, 32] also adopted the LSTM to model the temporal clues for video classification and reported promising performance, but did not explore the audio stream and employ advanced fusion strategies. Zha *et al.* [50] combined the ConvNet features with the dense trajectories [44] to achieve very competitive results.

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.

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
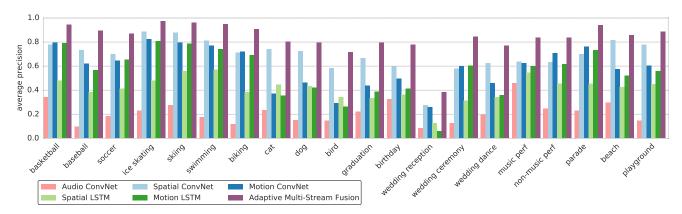
Figure 3. Per-class performance on CCV. Adaptive fusion of the multi-stream deep networks produces consistently better results than the individual streams on all the classes.

UCF-101		CCV	
Donahue *et al.* [9]	82.9	Lai *et al.* [24]	43.6
Srivastava *et al.* [39]	84.3	Jiang *et al.* [20]	59.5
Wang *et al.* [45]	85.9	Xu *et al.* [47]	60.3
Tran *et al.* [42]	86.7	Ma *et al.* [29]	63.4
Simonyan *et al.* [36]	88.0	Jhuo *et al.* [17]	64.0
Ng *et al.* [32]	88.6	Ye *et al.* [49]	64.0
Lan *et al.* [25]	89.1	Liu *et al.* [27]	68.2
Zha *et al.* [50]	89.6	Wu *et al.* [46]	83.5
Wu *et al.* [46]	91.3		
Ours (-A)	92.2	Ours (-A)	84.0
Ours	**92.6**	Ours	**84.9**

Table 3. 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.

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 fused using a novel method that adaptively learns the fusion weights for each class. Through imposing regularizations with the prior information and the sparsity, the weight learning process explores semantic class correlations, while suppressing inappropriate knowledge sharing among irrelevant classes. Our results confirm that all the adopted streams are effective for modeling not only simple human actions in short clips but also complex events in temporally untrimmed videos on the Internet. Combining all the streams by our proposed adaptive fusion method outperforms peer approaches with significant margins on two popular benchmarks.

The work in this paper is among the very few 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 [21], which may improve the results significantly.

References

[1] O. Abdel-Hamid, L. Deng, and D. Yu. Exploring convolutional neural network structures and optimization techniques for speech recognition. In *INTERSPEECH*, 2013. 3

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

[3] F. Bach, R. Jenatton, J. Mairal, G. Obozinski, et al. Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning*, 2011. 5

[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. 4

[5] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE TNN*, 1994. 3

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

[7] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio. Gated feedback recurrent neural networks. *CoRR*, 2015. 5

[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. 2, 4

[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. *CoRR*, 2014. 2, 5, 7, 8

[10] D. L. Donoho and I. M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the american statistical association*, 1995. 5

[11] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE TPAMI*, 2013. 1

[12] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *ICCV*, 2009. 7

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1, 2

[14] A. Graves, A. Mohamed, and G. E. Hinton. Speech recognition with deep recurrent neural networks. In *ICASSP*, 2013. 2

[15] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 2005. 4, 5

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 1997. 3

[17] 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. 8

[18] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. In *ICML*, 2010. 1, 2

[19] 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. 5

[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. 6, 8

[21] 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. 1, 2, 8

[22] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien. Lp-norm multiple kernel learning. *The Journal of Machine Learning Research*, 2011. 6

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 5

[24] K.-T. Lai, F. X. Yu, M.-S. Chen, and S.-F. Chang. Video event detection by inferring temporal instance labels. In *CVPR*, 2014. 8

[25] Z. Lan, M. Lin, X. Li, A. G. Hauptmann, and B. Raj. Beyond gaussian pyramid: Multi-skip feature stacking for action recognition. *CoRR*, 2014. 1, 7, 8

[26] I. Laptev. On space-time interest points. *IJCV*, 2007. 1

[27] 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. 2, 8

[28] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1

[29] A. J. Ma and P. C. Yuen. Reduced analytic dependency modeling: Robust fusion for visual recognition. *IJCV*, 2014. 8

[30] K. Nandakumar, Y. Chen, S. C. Dass, and A. K. Jain. Likelihood ratio-based biometric score fusion. *IEEE TPAMI*, 2008. 2

[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. 1

[32] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. *CoRRs*, 2015. 1, 2, 7, 8

[33] D. Oneata, J. Verbeek, C. Schmid, et al. Action and event recognition with fisher vectors on a compact feature set. In *ICCV*, 2013. 1

[34] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *ICCV*, 2007. 2, 4

[35] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *IJCV*, 2013. 1

[36] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 1, 2, 3, 5, 6, 7, 8

[37] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, 2014. 1, 2, 3, 5

[38] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, 2012. 6

[39] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using LSTMs. *CoRR*, 2015. 2, 7, 8

[40] X. Sun, M. Chen, and A. Hauptmann. Action recognition via local descriptors and holistic features. In *CVPR*, 2009. 1

[41] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper with Convolutions. *CoRR*, 2014. 1, 2, 5

[42] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. C3d: Generic features for video analysis. *CoRR*, 2014. 2, 8

[43] A. Van den Oord, S. Dieleman, and B. Schrauwen. Deep content-based music recommendation. In *NIPS*, 2013. 3

[44] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013. 1, 7

[45] H. Wang and C. Schmid. Lear-inria submission for the thumos workshop. *ICCV THUMOS Workshop*, 2013. 7, 8

[46] 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. *CoRR*, 2015. 1, 2, 7, 8

[47] Z. Xu, Y. Yang, I. Tsang, N. Sebe, and A. Hauptmann. Feature weighting via optimal thresholding for video analysis. In *ICCV*, 2013. 1, 2, 8

[48] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Video description generation incorporating spatio-temporal features and a soft-attention mechanism. *CoRR*, 2015. 2

[49] G. Ye, D. Liu, I.-H. Jhuo, and S.-F. Chang. Robust late fusion with rank minimization. In *CVPR*, 2012. 1, 2, 8

[50] S. Zha, F. Luisier, W. Andrews, N. Srivastava, and R. Salakhutdinov. Exploiting image-trained cnn architectures for unconstrained video classification. *CoRR*, 2015. 1, 7, 8