Modeling Multimodal Clues in a Hybrid Deep Learning Framework for Video Classification

Yu-Gang Jiang, Zuxuan Wu, Jinhui Tang, Zechao Li, Xiangyang Xue, and Shih-Fu Chang

***Abstract*—Videos are inherently multimodal. This paper studies the problem of exploiting the abundant multimodal clues for improved video classification performance. We introduce a novel hybrid deep learning framework that integrates useful clues from multiple modalities, including static spatial appearance information, motion patterns within a short time window, audio information, as well as long-range temporal dynamics. More specifically, we utilize three Convolutional Neural Networks (CNNs) operating on appearance, motion, and audio signals to extract their corresponding features. We then employ a feature fusion network to derive a unified representation with an aim to capture the relationships among features. Furthermore, to exploit the long-range temporal dynamics in videos, we apply two long short-term memory (LSTM) networks with extracted appearance and motion features as inputs. Finally, we also propose refining the prediction scores by leveraging contextual relationships among video semantics. The hybrid deep learning framework is able to exploit a comprehensive set of multimodal features for video classification. Through an extensive set of experiments, we demonstrate that: 1) LSTM networks that model sequences in an explicitly recurrent manner are highly complementary to the CNN models; 2) the feature fusion network that produces a fused representation through modeling feature relationships outperforms a large set of alternative fusion strategies; and 3) the semantic context of video classes can help further refine the predictions for improved performance. Experimental results on two challenging benchmarks—the UCF-101 and the Columbia Consumer Videos (CCV)—provide strong quantitative evidence that our framework can produce promising results: 93.1% on the UCF-101 and 84.5% on the CCV, outperforming several competing methods with clear margins.**

***Index Terms*—Deep learning, framework, fusion, video classification.**

I. Introduction

CLASSIFYING videos based on content semantics has been a hot research topic in multimedia for over a decade. Related techniques can be deployed in a wide range of applications such as video indexing and retrieval, smart advertising, etc. The key enabling factors behind the significant technical progress in recent years are discriminative and robust feature representations that can not only withstand large intra-class variations but also effectively differentiate multiple classes. Some popular feature descriptors such as SIFT [32] and HOG [5] model spatial clues like textures, while others such as HOF [6] and trajectory features [19], [40], [53] focus on motion information, a fundamental aspect of video depicting movements of objects among adjacent frames. Recently, deep neural networks, especially Convolutional Neural Networks (CNNs), have demonstrated great potentials for deriving robust features from raw data on a variety of tasks, including image classification [27], object detection [10], speech recognition [11], etc. Researchers have also attempted to apply deep learning techniques to the video domain. For instance, a straightforward extension is to stack multiple frames over time as inputs to CNNs for spatial-temporal feature learning [16], [24], [50]. Different from these works, Simonyan *et al.* [41] disentangled video feature learning with two independent CNNs operating on RGB frames and stacked optical flow images to capture spatial and motion information, respectively. Final predictions are derived by linear combination of the output scores of the two CNNs and the results are competitive against the hand-crafted trajectory features [53].

However, these works merely focus on appearance and motion information in videos, ignoring the long-range temporal clues therein because the training process of CNNs neglects the order of inputs (i.e., the order of the RGB frames or stacked optical flow images). In addition, the motion CNN can only account for object movements within very short time periods. We believe this is not sufficient for understanding video contents since different segments of videos usually correspond to different states of actions/events and their temporal order can assist recognition. For example, a "celebrating birthday" event could start with "making a wish", followed by "blowing out candles", and finally end with "eating cakes". Moreover, audio signal is an indispensable component of video data, providing complementary clues to visual information. In the case of a "celebrating birthday" event, a birthday song is typically associated with the video.

Further, video semantics usually do not occur in isolation, and recognizing a class of interest could benefit from its semantic contextual relationships. For example, similar human motion patterns can be observed in "running" and "playing tennis", and the likelihood of a video containing "running" could

Manuscript received April 29, 2017; revised September 13, 2017; accepted March 15, 2018. Date of publication April 12, 2018; date of current version October 15, 2018. This work was supported in part by the NSF China under Grants 61622204 and 61572134, and in part by the STCSM, Shanghai, China under 16QA1400500 and 16JC1420401. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Chang-Su Kim. *(Corresponding author: Jinhui Tang.)*

Y.-G. Jiang, Z. Wu, and X. Xue are with the School of Computer Science, Fudan University, Shanghai 200433, China (e-mail: ygj@fudan.edu.cn; zxwu@fudan.edu.cn; xyxue@fudan.edu.cn).

J. Tang and Z. Li are with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: jinhuitang@njust.edu.cn; zechao.li@njust.edu.cn).

S.-F. Chang is with the Department of Electrical Engineering, Columbia University, New York City, NY 10027 USA (e-mail: sfchang@ee.columbia.edu).

Digital Object Identifier 10.1109/TMM.2018.2823900

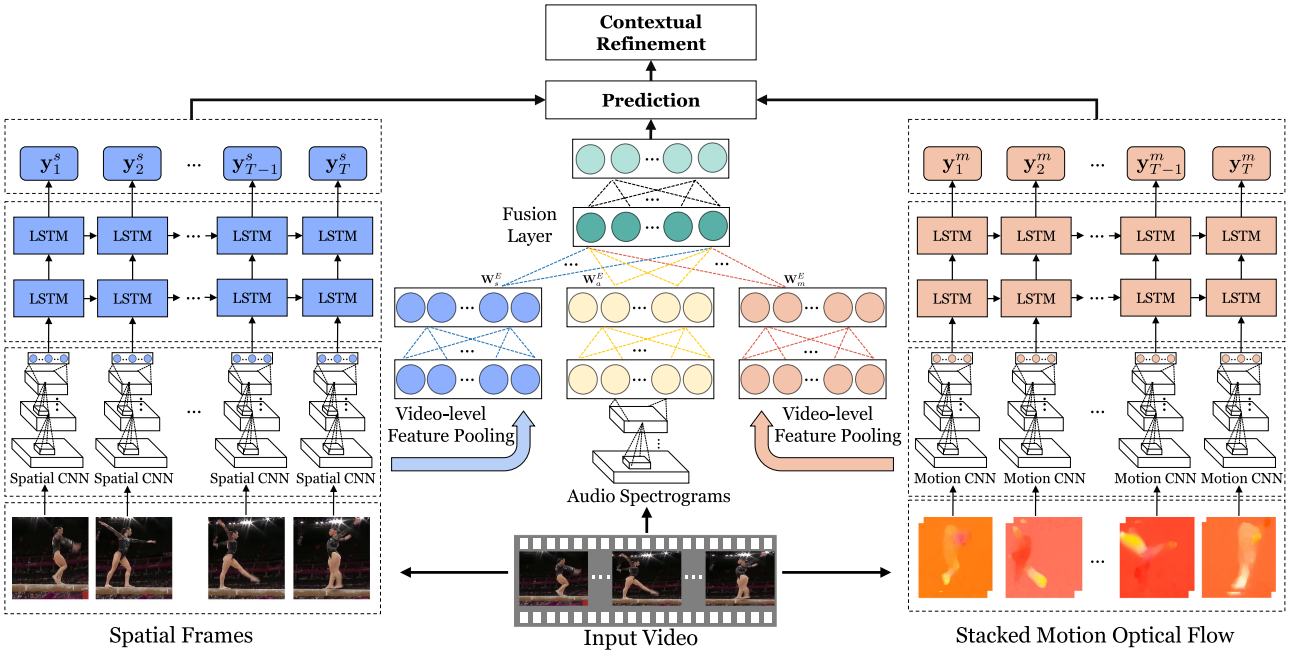


Fig. 1. The pipeline of the proposed hybrid deep learning framework. For a video clip, we first extract spatial, motion and audio features with three CNNs operating on video frames, stacked optical flow images and audio signals respectively. To capture long-range temporal dynamics in videos, we leverage two LSTM models with inputs of the extracted spatial and motion features. Further, we also utilize a feature fusion network to integrate multiple features into a unified representation to perform classification with carefully designed regularizations aiming to exploit feature relationships. Finally, we combine the outputs of the LSTM models and the feature fusion network with contextual refinement to generate the final prediction scores. See texts for more discussions.

potentially help recognize "playing tennis". These useful clues are either overlooked or modeled with complicated models that are infeasible to scale up in many existing works.

To mitigate these limitations, we propose a hybrid deep learning framework to exploit the abundant multimodal clues embedded in videos, including static spatial information, motion patterns, audio information and long-range temporal clues as well as the contextual relationships among multiple classes of video semantics. Motivated by the great success of Recurrent Neural Networks (RNNs) for sequence modeling tasks [11], [12], we leverage Long Short Term Memory (LSTM) for temporal information. Furthermore, different from existing methods that integrate features in a straightforward and heuristic way by either feature concatenation or score averaging, we are interested in exploring the feature correlations. To this end, we apply a deep neural network with carefully designed regularizations [22] to integrate the extracted static appearance, short-term motion and audio features. Then we combine the predictions from this network with the outputs of the LSTMs. Finally, we refine the prediction scores in consideration of contextual relationships among video semantics in a simple yet effective manner.

The proposed framework is illustrated in Fig. 1. In particular, we first compute spatial appearance, short-term motion (based on stacked optical flow images) and audio features with CNN models. The spatial and motion features are further utilized as inputs of LSTMs to capture the long-range temporal temporal clues. Then, a feature fusion network takes the video-level features (spatial, motion and audio) as inputs to derive a unified representation for predicting video semantics. The outputs of the feature fusion networks are further combined with scores from the LSTMs and then refined by taking advantage of the contextual relationships of video semantics. The main contributions are summarized as follows:

- We propose to exploit a comprehensive set of multimodal clues in a hybrid deep learning framework for video classification, including static spatial appearance, motion and audio information, long-range temporal coherence and contextual relationships among video semantics.
- We demonstrate that the LSTMs, modeling the long-range temporal information in video sequences through an explicitly recurrent manner, are highly complementary to the CNNs.
- We resort to the rich contextual relationships among video semantics in a simple yet effective way to further refine predictions for improved performance.
- We conduct experiments on two challenging benchmarks, and the experimental results provide strong quantitative evidence that our framework can produce promising results, outperforming a set of competing methods with clear margins.

This work extends from a conference paper [59] by further incorporating audio and semantic contextual relationships in the hybrid framework. New experiments are conducted to verify the effectiveness of the technical extensions and extra amplified discussions are provided throughout the paper. The remaining sections are organized as follows. We first review related works in Section II and elaborate the proposed hybrid deep learning framework in Section III. We then present and discuss the experimental results and comparisons in Section IV. Finally, Section V concludes this paper.

II. Related Works

We divide the discussions of related works into the following five subsections.

A. Hand-crafted Features

There is a large set of prior works on video classification in the multimedia and computer vision communities (see [18] for a survey). Among these works, designing powerful feature representations is an important topic due to the significant role of features in a typical video recognition pipeline. The success of image descriptors like SIFT and HoG has spurred the developments of video representations by considering the temporal feature of videos. For example, Harris corner detector is extended into 3D volumes to identify space-time interest points [30]. Similarly, based on HoG features, 3D spatial-temporal gradients are derived as local descriptors for action recognition [25]. Wang *et al.* proposed to track densely sampled local patches over time in an optical flow field to compute dense trajectory features, which achieved superior performance on a variety of benchmarks when coupled with quantization techniques like Bag-of-Words and Fisher Vector [13], [37]. However, these video representations focus on modeling local motion patterns within short time periods and the feature encoding methods while powerful totally discards the temporal information of videos.

B. CNN Representations

Different from hand-crafted features, recent advances on CNNs in image [10], [27] and speech domain [11] have encouraged works to learn features directly from raw video data. The most straightforward way to utilize CNN on video data is stacking frames as inputs with an aim to learn spatial-temporal features using 3D convolutions [16], [24], [50]. However, these works demonstrate worse performance than state-of-the-art trajectory features [53]. This might result from the difficulty to learn 3D features with insufficient training data. To effectively model 3D signals, Simonyan *et al.* proposed to utilize two independent CNNs to capture spatial and motion information operating on RGB frames and stacked optical flow images, separately. Based on this approach, Wang *et al.* proposed to learn the transformation between two states triggered by actions [56]. Feichtenhofer *et al.* experimented with different fusion approach to combine spatial and temporal features [9]. During the training process of CNNs, the temporal order of frames and stacked optical flow images is discarded and thus the temporal structures of videos are ignored.

C. Temporal Information

Graphical models, including Conditional Random Fields (CRF), Hidden Markov Models (HMM), etc., have been widely adopted to capture long-term temporal structures [49], [51], [57]. For example, Tang *et al.* proposed a variable duration HMM to model state changes in videos [49]. Instead of using graphical models, Fernando *et al.* utilized a ranking machine to account for the temporal order of frames. Wang *et al.* proposed the temporal segment networks, which used a consensus function to combine segment scores generated by two-stream networks [55].

Many works resort to LSTM to capture temporal dynamics in videos due to its great success in sequential modeling tasks like speech recognition [11] and video captioning [47]. Srivastava *et al.* proposed to learn video features using an auto-encoder framework [45] based on LSTMs. Donahue *et al.* utilized two LSTM models using spatial and motion features extracted from CNN models [8]. Ng *et al.* further deepened LSTM to five layers and experimented with several pooling strategies [35]. Our work leverages LSTMs for temporal modeling to explicitly complement the limitation of the frame-based CNN models.

D. Feature Fusion

Extensive works have been conducted on the fusion of multiple features, the complementarity of which is expected to promote classification accuracy. There are two popular fusion strategies, i.e., early fusion and late fusion performed at the feature level and the classification score level, respectively [43], [61]. Typically, early fusion integrates features by direct concatenation [53] or linear combination of their kernels [65] before classification. In addition, Multiple Kernel Learning (MKL) can also be applied to combine feature kernels, where the weights are automatically learned. Late fusion, on the other hand, combines prediction scores from multiple classifiers, each of which is independently trained with a single feature [31], [63]. Both fusion methods are popular due to their simplicity, however, they assume the features or prediction scores are explicitly complementary to one another and fail to consider potential hidden correlations among features. Recently, Srivastava *et al.* utilized Deep Boltzmann Machines (DBM) to derive an embedding of images and texts [45] and Ngiam *et al.* used deep auto-encoder to learn the relationships between different modalities [36]. Wu *et al.* proposed to explore feature and class relationships [58] by imposing trace norms. In this work, to alleviate computational complexity, we adopt a regularized neural network to automatically learn dimension-wise correlations of features extracted from state-of-the-art CNN models.

E. Contextual Relationships

As stated above, the co-occurrence of video semantics, serving as context, can provide useful information. For example, Rabinovich *et al.* proposed to incorporate the semantics context information with a CRF model [38]. Jiang *et al.* modeled the class relationships with a semantic diffusion algorithm [21]. Deng *et al.* leveraged a graphical model to encode label hierarchies for improved image classification performance [7]. Wu *et al.* proposed to capture the relationships of video semantics by regularizing the classification process [58]. Chen *et al.* utilized confusion matrix to predict the context of a category when training CNNs [3]. In our paper, we propose to utilize confusion matrix as contextual relationships derived from trained models, to refine the prediction scores as a post-processing step. Therefore, the recognition of a class of interest can benefit from its related classes.

III. Methodology

We now elaborate the proposed hybrid deep learning framework illustrated in Fig. 1. We first introduce the multimodal features extracted by CNN models and present the modeling of temporal dynamics in videos with LSTM models. Then we describe the feature fusion framework which is designed to model feature correlations. Finally, we introduce the detailed design of contextual refinement.

A. Spatial, Motion and Audio CNN Features

CNN models usually contain alternating convolutional and pooling layers to learn features from input images, followed by fully-connected (FC) layers for classification. In our framework, we first compute spatial and motion features based upon the two-stream approach [41], where two independent CNNs are trained with RGB frames and stacked optical flow images, respectively. More concretely, the spatial stream models static appearance information like texture from sampled video frames as in conventional CNNs for image classification. The motion CNN takes stacked optical flow images as inputs to capture object movements within a short time window. Optical flow is an explicit form of motion patterns derived by computing displacement vector fields between two adjacent frames, whose horizontal and vertical components are then used to generate two images. Multiple optical flow images are further stacked to represent motion information in a short period, upon which convolution is performed. Given a video at testing phase, each stream averages prediction scores produced by soft-max layer from 25 uniformly sampled frames (or stacked optical flow images) and the scores from the two streams are further linearly combined as the final prediction. In our work, we compute the outputs from the first FC layer of two CNNs, which are observed to be effective in many tasks [39], as the spatial and motion features to model long-term temporal structures and explore their correlations for improved performance.

In addition, we also utilize a CNN model to capture the acoustic information in videos as a compliment to visual information. Particularly, we convert the 1D soundtrack extracted from a video clip into a 2D spectrogram image with Short-Time Fourier Transformation, demonstrating changes of frequency-scale along with time. Then, inspired by [52], we take the spectrograms as inputs to a CNN network to capture the acoustic clues.

B. Temporal Modeling with LSTM

The two-stream approach focuses only on appearance and short-time motion information, which ignores the long-term temporal dynamics in videos. Therefore, we employ the LSTM model due to its great success in sequential modeling tasks [8], [11], [62]. Compared with conventional RNN models that map input data recursively to outputs through hidden states, an LSTM additionally incorporates a memory cell with multiple gates governing information into and out of the cell, enabling it to model long sequences without suffering from the "vanishing gradients" effect.

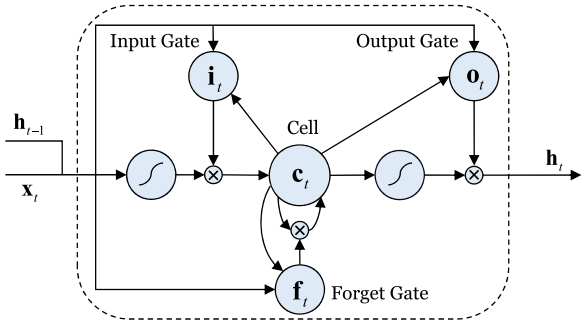


Fig. 2. An illustration of an LSTM unit.

Formally, an LSTM takes a sequence $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$ as inputs and maps it to an output sequence $(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T)$ by recursively computing activations of the units from $t = 1$ to $t = T$ as following:

$$\mathbf{i}_t = \sigma(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{W}_{ci}\mathbf{c}_{t-1} + \mathbf{b}_i),$$
$$\mathbf{f}_t = \sigma(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{W}_{cf}\mathbf{c}_{t-1} + \mathbf{b}_f),$$
$$\mathbf{c}_t = \mathbf{f}_t\mathbf{c}_{t-1} + \mathbf{i}_t \tanh(\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c),$$
$$\mathbf{o}_t = \sigma(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{W}_{co}\mathbf{c}_t + \mathbf{b}_\mathbf{o}),$$
$$\mathbf{h}_t = \mathbf{o}_t \tanh(\mathbf{c}_t).$$

Here, at the t-th time step, we denote the input features as $\mathbf{x}_t$ and the hidden states as $\mathbf{h}_t$. And $\mathbf{c}_t$ represents the contents of the memory unit. The activations of the input, forget and output gates are represented as $\mathbf{i}_t, \mathbf{f}_t, \mathbf{o}_t$, respectively. $\mathbf{W}_{\alpha\beta}$ represents the transition weights from component α to component β, and $\mathbf{b}_\alpha$ is the corresponding bias term. In addition, $\sigma(x) = \frac{1}{1+e^{-x}}$ is the non-linear sigmoid function. We present the structure of an LSTM unit in Fig. 2.

One can also stack hidden states to deepen the LSTM model aiming to increase its discriminative power. A softmax layer $\mathbf{y}_t = \texttt{softmax}(\mathbf{W}_c\mathbf{h}_t)$ can then be applied on top of the hidden states to obtain the prediction scores at each time-step, where $\mathbf{W}_c$ denotes the weights for classifiers. The training of LSTM is usually conducted with stochastic gradient descent using the Back-Propagation Through Time (BPTT) algorithm [12].

The memory cell regulated by different non-linear gates enables the LSTM model to store information progressively. More concretely, for the t-th time step, the current feature representation $\mathbf{x}_t$ together with information from the past $\mathbf{h}_{t-1}$ are fed into all gates and the memory cell. Past information stored in the memory cell $\mathbf{c}_{t-1}$ regulated by the activations of the forget gate $\mathbf{f}_t$ is linearly combined with the squashed inputs multiplied by the activation of the input gate $\mathbf{i}_t$ to generate the current "memory". This facilitates the LSTM model to learn when to utilize current information or forget previous contents. Furthermore, the information that will be used for future states is regulated by the output gate $\mathbf{o}_t$. The interactions between the memory units and these multiplicative gates allow LSTM to capture the temporal dynamics in long sequences, making it a natural fit for video classification.

In our framework (illustrated in Fig. 1), we model the temporal information in videos with two LSTMs, operating

on a spatial feature sequence $(\mathbf{x}_1^s, \mathbf{x}_2^s, \ldots, \mathbf{x}_T^s)$ and a motion feature sequence $(\mathbf{x}_1^m, \mathbf{x}_2^m, \ldots, \mathbf{x}_T^m)$, respectively. Once the model is trained, the two LSTM models will produce two sets of predictions: $(\mathbf{y}_1^s, \mathbf{y}_2^s, \ldots, \mathbf{y}_T^s)$ for the spatial stream and $(\mathbf{y}_1^m, \mathbf{y}_2^m, \ldots, \mathbf{y}_T^m)$ for the motion stream. We compute the prediction from the last time step $\mathbf{y}_T$ of a sequence as the score for the entire video, because it contains information from all previous steps.

C. Regularized Feature Fusion Network

The spatial, motion and audio features characterize the same video from different perspectives (i.e., person-related static appearance information, body motions and sound), and thus certain correlations between these features might exist. We posit that an ideal unified representation is expected to contain information shared by multiple features as well as the special aspect of each feature. This requires modeling feature relationships explicitly instead of uniform fusion approaches. To this end, we utilize a regularized feature fusion network [22] to fully exploit feature relationships (see Fig. 1). Given a video clip, we compute its video-level appearance and motion features by simply averaging descriptors from all frames. The spatial, motion and audio features are separately transformed into a higher space with one hidden layer. We then apply one hidden layer to absorb all the features to derive a unified representation, regularized by carefully designed norms to explore feature relationships.

We represent the n-th training video as a 4-tuple $(\mathbf{x}_n^s, \mathbf{x}_n^m, \mathbf{x}_n^a, \mathbf{l}_n)$, where $\mathbf{x}_n^s = \sum_{t=1}^T \mathbf{x}_{n,t}^s \in \mathbb{R}^{d_s}$ and $\mathbf{x}_n^m = \sum_{t=1}^T \mathbf{x}_{n,t}^m \in \mathbb{R}^{d_m}$ denote the video-level spatial and motion descriptors respectively, $\mathbf{x}_n^a \in \mathbb{R}^{d_a}$ as the audio feature derived from the audio CNN and $\mathbf{l}_n$ is the corresponding ground-truth label. We first consider the training of a neural network with a single feature as inputs. Let $g(\cdot)$ denote the non-linear function approximated by the neural network. To learn the optimal weights of the model, we minimize the following objective function:

$$\min_{\mathbf{W}} \sum_{i=1}^{N} \|g(\mathbf{x}_i) - \mathbf{l}_i\|^2 + \lambda_1 \Phi(\mathbf{W}). \tag{1}$$

Here N denotes the number of videos in the training set, $\mathbf{x}_i$ is the i-th training sample and $\mathbf{W}$ represents the weights of the network. The first item in the equation is the empirical loss, and the second term is a Frobenius norm on the weight matrices to prevent over-fitting.

We now introduce the fusion of multiple features in a regularized framework. Given three types of features, we first perform feature transformation independently and then integrate them to derive a fused representation with a fusion layer (denoted as the E-th layer). In the fusion process, we impose a structural ℓ_{21} norm to explore the relations of the features. The optimization problem now becomes:

$$\min_{\mathbf{W}} \mathcal{L} + \lambda_1 \Phi(\mathbf{W}) + \frac{\lambda_2}{2} \left\|\mathbf{W}^E\right\|_{2,1}. \tag{2}$$

Here $\mathcal{L} = \sum_{i=1}^N \|g(\mathbf{x}_i^s, \mathbf{x}_i^m, \mathbf{x}_i^a) - \mathbf{l}_i\|^2$, $\mathbf{W}^E = [\mathbf{W}_s^E, \mathbf{W}_a^E, \mathbf{W}_m^E] \in \mathbb{R}^{P \times D}$ represents the stacked weights for the E-th layer, where $D = d_s + d_m + d_a$ and P denotes dimension for the unified feature representation.

Compared with (1), an ℓ_{21} norm is appended to regularize the fusion process of the the E-th layer aiming to exploit feature relationships. The $\|\mathbf{W}\|_{2,1}$ is defined as $\sum_i \sqrt{\sum_j w_{ij}^2}$, and we can see that it first computes ℓ_2 norm for each row (weights of the three features), and then ℓ_1 norm for the resulting vector, which will force the matrix $\mathbf{W}^E$ to be row sparse and produce similar zero/nonzero patterns for the columns. In other words, the norm will be minimized when there are only a few non-zero rows in the weight matrix, which serve as the shared discriminative information of these features.

In addition, based on the idea that a good unified representation should not only exploit feature correlations but also preserve the unique information of each feature, we additionally regularize the fusion process with an ℓ_1 norm and rewrite (2) as:

$$\min_{\mathbf{W}} \mathcal{L} + \lambda_1 \Phi(\mathbf{W}) + \frac{\lambda_2}{2} \left\|\mathbf{W}^E\right\|_{2,1} + \lambda_3 \left\|\mathbf{W}^E\right\|_{1,1}. \tag{3}$$

The regularizer $\|\mathbf{W}^E\|_{1,1}$ complements the $\|\mathbf{W}^E\|_{2,1}$ norm to be robust by preventing it from sharing incorrect information, which enables different features to select different neurons (i.e., the unique information of these features).

We now move on to discuss the optimization in (3), which is nonconvex because of the multi-layer neural network. Therefore, we train the network using back-propagation with gradient descent method in two scenarios:

1) The E-th layer. Since the regularization is imposed only on the E-th layer, we treat it differently when performing gradient descent. The difficulty of the optimization here lies in the last two non-smooth terms, which are non-differentiable. Thus we cannot directly apply gradient descent. Instead, we utilize the proximal operation to evaluate their gradients. More specifically, we split the objective function into two components:

$$p = \mathcal{L} + \lambda_1 \Phi(\mathbf{W}),$$

$$q = \frac{\lambda_2}{2} \left\|\mathbf{W}^E\right\|_{2,1} + \lambda_3 \left\|\mathbf{W}^E\right\|_{1,1}.$$

Here p is a smooth function whose gradients are easy to obtain and q is a non-smooth function. We utilize a proximal operator to update the weights for the i-th iteration:

$$\left(\mathbf{W}^E\right)^{(i)} = \text{Prox}_q\left(\left(\mathbf{W}^E\right)^{(i)} - \nabla p\left(\left(\mathbf{W}^E\right)^{(i)}\right)\right),$$

where $\text{Prox}_q(\mathbf{W}) = \arg\min_{\mathbf{V}} \|\mathbf{W} - \mathbf{V}\| + q(V)$. Note that q here is a combination of ℓ_{21} and ℓ_{11} norms, and thus the proximal operator can be can be derived as:

$$\mathbf{W}_{r\cdot}^E = \left(1 - \frac{\lambda_2}{\|\mathbf{U}_{r\cdot}\|_2}\right) \mathbf{U}_{r\cdot}, \forall r = 1, \ldots, P, \tag{4}$$

where $\mathbf{U}_{r\cdot} = \max\{|\mathbf{V}_{r\cdot}| - \lambda_3, 0\} \cdot sign[\mathbf{V}_{r\cdot}]$, and $\mathbf{W}_{r\cdot}, \mathbf{U}_{r\cdot}, \mathbf{V}_{r\cdot}$ represents the r-th row of matrix $\mathbf{W}, \mathbf{U}$ and $\mathbf{V}$, respectively.

2) Other layers. Since there are no non-smooth regularizations for other layers, we compute their gradients directly and then update the weight matrix with gradient descent

Algorithm 1: Training algorithm of the regularized feature fusion network

Input : $\mathbf{x}_n^s$, $\mathbf{x}_n^m$ and $\mathbf{x}_n^a$: the video-level spatial, motion and audio CNN features of the n-th video;
$\mathbf{y}_n$: the corresponding ground-truth label;
randomly initialized weights $\mathbf{W}$;

```
begin
    for epoch ← 1 to M do
        Run a feed-forward pass through the network
        to obtain perdition error;
        for l ← L to 1 do
            Gradient descent with Eqn. (5);
            if l == E then
                Update the weights with proximal
                operation with Eqn. (4);
            end
        end
    end
end
```

as in [1]. Let $\mathbf{G}^l$ represent the gradients of $\mathbf{W}^l$, the weight matrix of the lth layer is updated as:

$$\mathbf{W}^l = \mathbf{W}^l - \eta \mathbf{G}^l. \tag{5}$$

Although the two regularization norms in function q incur extra computation cost, it is worth noting that the complexity of computing the proximal operator is $O(P \times D)$, which is fast to evaluate. The proposed method is also general for fusing more features at a linearly growing computational cost rather than cubic cost as in [22]. The overall training process of the feature fusion framework is presented in Algorithm 1.

D. Contextual Relationships

Given the classification scores from the two LSTMs and the regularized feature fusion network, accounting for spatial, motion, audio and long-term temporal clues in videos, we are interested in incorporating contextual relationships to further refine the outputs for improved performance. More specifically, for each video sample, we first linearly average the probabilities to obtain a compact prediction. Then we utilize a simple approach to refine the prediction with contextual relationships, which provide useful information of semantics co-occurrence. For example, "baseball" is more related to "soccer" than "diving", since "diving" contains totally different motion patterns. And if the likelihood for the video to be "soccer" is extremely low, then it is also unlikely to be "baseball". Existing works often resort to external knowledge like WordNet or word vectors to obtain class relationships, which are either hand-crafted or trained on text corpus and hence fail to consider visual patterns. In our work, we simply rely on the trained models to produce class relationships by computing the confusion matrix, which is a good indicator on how classes are related.

Formally, for a total of C classes, we denote $f(\cdot) \in \mathbb{R}^C$ as the mapping from the input to the linearly averaged prediction and then the confusion matrix $\mathbf{R} \in \mathbb{R}^{C \times C}$ is defined as following:

$$\mathbf{R}_{ij} = \frac{1}{|C_i|} \left| \{(\mathbf{x}, C_i) \in \mathcal{V} : \arg\max f(\mathbf{x}) = C_j\} \right|. \tag{6}$$

Here, $\mathcal{V}$ is the validation set and $|\cdot|$ is the cardinality function. When $i \neq j$, $\mathbf{R}_{ij}$ measures the number of samples originally belongs to the C_i class but are misclassified into C_j. It is easy to understand that if C_i and C_j are close, the value $\mathbf{R}_{ij}$ will be large since they are difficult to separate. Then for the i-th video sample, we refine its prediction score by:

$$\mathbf{p}_i = \mathbf{R} f(\mathbf{x}_i), \tag{7}$$

where $\mathbf{p}_i$ is the final probability for the i-th video sample. The recognition of a class of interest can benefit from the contextual relationships in that information from its related classes is utilized to adjust its confidence based on semantic co-occurrence. Note that researchers also employ multi-label loss functions like hinge loss or ranking loss [2] to consider context in an explicit way but they are not suitable for single-label recognition tasks. Our approach models contextual relationships among classes by analyzing their appearance and motion patterns, and thus it is general to both multi-label and single-label scenarios.

E. Discussion

The proposed framework is able to model a comprehensive set of multimodal features, including static appearance, motion patterns in a short time window, long-range temporal dynamics and acoustic clues, which are all critical for understanding video contents since they describe videos from different perspectives. In our framework, we train different components independently rather than jointly in an end-to-end manner. Although training jointly is theoretically feasible, it would require extra training samples to prevent under-fitting in the complicated process and it is observed in [8] the performance gain of joint training is rather marginal. In addition, separate training ensures flexibility in the framework, since a component can be replaced easily without incurring the re-training of the whole complex framework. For example, one can easily update the framework with more powerful CNN models like GoogleNet [48] and ResNet [14] or better RNN models [4]. The main purpose of this paper is to demonstrate that a comprehensive set of features are demanded for improved video classification. In addition, in this work, we mainly demonstrate audio information captured by a CNN model can serve as an effective complement to visual information, and thus we do not investigate modeling temporal audio dynamics with LSTMs.

IV. Experiments

In this section, we first introduce the experimental settings and then discuss the results of the proposed hybrid deep learning framework on two popular benchmarks.

A. Experimental Setup

1) Datasets: To investigate the effectiveness of the proposed hybrid deep learning framework, we utilize the following two benchmarks:

- UCF-101 [44]. The UCF-101 benchmark is a widely adopted dataset for human action recognition, which contains 13,320 video clips manually annotated into 101 human actions, totaling 27 hours. We conduct experiments using three training and testing splits following the protocol defined in [20]. Performance is measured by the average classification accuracy of all three splits.
- Columbia Consumer Videos (CCV) [23]. It consists of 9,317 videos collected from YouTube belonging to 20 categories, including "basketball", "wedding dance", "soccer", etc. Following [23], we utilize a training set of 4,659 videos and a testing set of 4,658 videos. We compute average precision for each class and report the mean AP over all classes.

2) Implementation Details: We utilize the VGG_19 network [42] to extract spatial features and the CNN_M model [41] to compute motion and audio features, due to their expressive performance on the ImageNet ILSVRC-2012 validation set: a 7.5% and 13.5% top-5 error rates, respectively. We first pre-train the spatial and audio CNN with ImageNet data and then fine-tune the network on video frames and spectrograms respectively. Note that for the audio CNN we observe better performance with pre-training though the images are spectrograms. Due to the lack of existing models trained on 20 channels (the input data format for the motion CNN), we train the motion CNN from scratch. To further promote the performance, we also employ simple data augmentation methods like cropping and flipping as in [41].

We apply stochastic gradient descent using back-propagation to train the CNN models. We adopt a batch size of 256 and fix the momentum to be 0.9. To fine-tune the spatial and audio CNN, we first set the initial learning rate to 10^{-3} and decay it by a factor of 10 after every 14K iterations. Different from [41], we begin with a smaller rate rather than 10^{-2}. To train the motion network, we set the initial learning rate to 10^{-2}, and then decay it by a factor of 10 after every 100K iterations. We adopt the popular Caffe [17] toolbox with modifications to support parallel training on multiple GPUs for implementations.

To capture the long-range temporal dynamics, we utilize two two-layer LSTMs operating on spatial and motion CNN features respectively. Both LSTMs contain 1,024 hidden neurons for the first layer and 512 units for the second layer. We train the network with a parallel implementation of Back-Propagation Through Time (BPTT) algorithm. The mini-batch size is set to 10 and the maximal iterations to 150K. We also fix the learning rate and momentum to 10^{-4} and 0.9 respectively.

Finally, to learn the optimal weights for the feature fusion network, we follow the procedures described in Alg. 1. The network contains four hidden layers shown in Fig. 1. More concretely, we first employ a layer with 200 neurons for each of the spatial, motion and audio feature for independent feature transformation, followed by one layer with 200 neurons to perform feature fusion. The derived unified feature representations are further trained to categorize videos into semantic classes. We utilize a learning rate of 0.7 and fix λ_1 to 3×10^{-5} to prevent over-fitting. λ_2 and λ_3 are selected using cross-validation.

3) Compared Approaches: To evaluate the proposed framework, we compare with the following alternative competing methods: (1) **Spatial CNN**, **Motion CNN** and **Audio CNN**, which are independently trained with raw RGB frames, stacked optical flow images and audio spectrograms; (2) **Spatial LSTM** and **Motion LSTM**, which denote LSTM models operating on extracted spatial and motion CNN features respectively; (3) **SVM-based Early Fusion (SVM-EF)**, which averages three χ^2-kernels derived from spatial, motion and audio features for classification with an SVM; (4) **SVM-based Late Fusion (SVM-LF)**, which employs a separate SVM for each feature and then linearly average their prediction scores; (5) **Multiple Kernel Learning (SVM-MKL)**, which integrates three features using the ℓ_p-norm MKL [26] with $p = 2$; (6) **Early Fusion with Neural Networks (NN-EF)**, which performs classification with a 4-layer neural network operating on the concatenated features; (7) **Late Fusion with Neural Networks (NN-LF)**, which combines predictions from three individual neural networks trained on three types of features respectively; (8) **Multimodal Deep Boltzmann Machines (M-DBM)** [36], [46], which performs feature fusion in a DBM without regularizations; (9) **RDNN** [58], which utilizes a different regularization scheme with higher computational complexity.

TABLE I
PERFORMANCE OF THE LSTM AND THE CNN MODELS ON UCF-101 AND CCV

	UCF-101	CCV
Spatial CNN	80.1%	75.0%
Spatial LSTM	83.3%	43.3%
Motion CNN	77.5%	58.9%
Motion LSTM	76.6%	54.7%
Audio CNN	16.2%	21.5%
CNN + LSTM (Spatial)	84.0%	77.9%
CNN + LSTM (Motion)	81.4%	70.9%
CNN + LSTM (Spatial & Motion)	90.1%	81.7%
CNN + LSTM (Spatial & Motion) + Audio	90.3%	82.4%

"+" indicates model fusion, which simply uses the average prediction scores of different models.

Notice that the first two classes of methods are components of the proposed framework and we report their performance independently to better analyze their contribution in the overall framework. The remaining seven methods aim to integrate the spatial, motion and audio features to improve classification performance.

B. Results and Discussions

1) Multimodal Representations:

a) Temporal Modeling: In this section, we investigate the effectiveness of LSTMs on modeling the long-range temporal dynamics in video sequences. Table I presents the results of different methods on UCF-101 and CCV. We first compare the performance of LSTM models with CNNs as shown in the top two groups. Since CNN models fail to take the temporal order of frames into consideration, we expect the performance of CNN models is worse than LSTMs. On UCF-101, we can see that Spatial LSTM slightly outperforms Spatial CNN, but the Motion LSTM is marginally worse than Motion CNN. Since the motion LSTM takes stacked optical flow images as inputs, we posit this might result from the lack of training data to learn

the optimal weights unlike the training of Spatial LSTM, where a large number of redundant frames could be utilized.

For CCV, CNN models perform consistently better than LSTM models on both spatial and motion streams. Compared to UCF-101, CCV contains more diversified and noisy videos without post-editing, whose duration are also significantly longer than those in UCF-101 (in average, 80 seconds *vs.* 8 seconds). Therefore, the noises in such videos could significantly degrade the performance of LSTM models. The noisy nature of CCV videos can also be reflected by the relatively low performance of motion streams operating on optical flow images, which are sensitive to camera motions and cluttered backgrounds.

b) Audio Modeling: The performance of audio CNN is presented in the middle of Table I. Audio CNN operating on spectrograms achieves 16.2% and 21.5% on UCF-101 and CCV respectively. Note that the performance on UCF-101 is measured by mean accuracy over 101 classes, however only 51 categories contain soundtracks and thus the actual accuracy is 32.1%. Audio signals are usually not robust and discriminative as visual clues due to the noises in video backgrounds.

c) Feature Complementarity: We now study whether the extracted multimodal representations are complementary through linearly averaging the outputs of the trained models. Here we only adopt simple late fusion and we will experiment with different fusion strategies in Sec. 4.2.2.

Results are summarized in the bottom two groups of Table I. We first combine CNN and LSTM models for both spatial and motion streams, and the fusion offers significant performance gains on both benchmarks. The combination of CNN and LSTM on the spatial stream offers 0.7% and 2.9% improvements over the best single model on UCF-101 and CCV, respectively. On the motion stream, the performance gains of fusion are more noticeable, 3.9% and 12% on UCF-101 and CCV. The consistent trend when fusing CNN with LSTM models on both streams confirms the complementarity of these features. Further, we also combine all spatial and motion models, offering 90.1% and 81.7% on UCF-101 and CCV respectively. This clearly verifies that spatial and motion features are very complementary. In addition, we also incorporate audio clues to complement the visual information, and this entire set of features attains the highest performance on both datasets: 90.3% and 82.4%. Therefore, we believe a successful video classification system should integrate all these features.

2) Feature Fusion: We now move on to evaluate the proposed regularized feature fusion network and compare with competing methods. Table II presents the results and comparisons. In particular, the first group compares the results of the spatial, motion and audio features using SVMs. This set of experiments serves as baselines to better understand the improvements of fusion using SVM classifiers (summarized in the second group of Table II). See Table I for results that are directly obtained from CNN models. We also compare with alternative neural network based fusion methods as summarized in the third group in Table II. Finally, we report the results of our method in the bottom row.

Based on the results, we have the following observations: (1) the fusion of multiple features offers performance gains on both UCF-101 and CCV and the improvements on UCF-101 are more significant than those on CCV; (2) the proposed feature fusion approach outperforms other neural network based methods; (3) the performance gain over the regularizer-free M-DBM network confirms modeling feature relationships is important during fusion; (4) our framework also outperforms RDNN slightly at a much lower cost as previously mentioned. Note that compared with the last row of Table I, the Regularized Fusion Network here performs slightly worse since it does not incorporate temporal information modeled by LSTMs. The fusion with temporal clues will be studied in the next subsection.

TABLE II
PERFORMANCE COMPARISON ON UCF-101 AND CCV, USING VARIOUS FUSION APPROACHES TO COMBINE THE MULTIMODAL CLUES

	UCF-101	CCV
Spatial SVM	78.6%	74.4%
Motion SVM	78.2%	57.9%
Audio SVM	16.7%	22.1%
SVM-EF	86.9%	75.9%
SVM-LF	85.4%	75.1%
SVM-MKL	87.1%	75.6%
NN-EF	86.6%	76.1%
NN-LF	85.4%	75.4%
M-DBM	87.0%	76.0%
RDNN	88.4%	76.2%
Non-regularized Fusion Network	87.2%	75.8%
Regularized Fusion Network	**88.7%**	**76.7%**

TABLE III
COMPARISON WITH STATE-OF-THE-ART RESULTS

UCF-101		CCV	
Donahue *et al.* [8]	82.9%	Lai *et al.* [28]	43.6%
Srivastava *et al.* [45]	84.3%	Jiang *et al.* [23]	59.5%
Wang *et al.* [53]	85.9%	Xu *et al.* [60]	60.3%
Tran *et al.* [50]	86.7%	Ma *et al.* [33]	63.4%
Simonyan *et al.* [41]	88.0%	Jhuo *et al.* [15]	64.0%
Ng *et al.* [35]	88.6%	Ye *et al.* [63]	64.0%
Lan *et al.* [29]	89.1%	Liu *et al.* [31]	68.2%
Zha *et al.* [64]	89.6%	Wu *et al.* [59]	83.5%
Wang *et al.* [54]	91.5%	Nagel *et al.* [34]	71.7%
Wang *et al.* [56]	92.4%		
Hybrid Framework	92.1%	Hybrid Framework	84.0%
Hybrid Framework-DASD	92.4%	Hybrid Framework-DASD	84.2%
Contextual Refinement	**93.1%**	Contextual Refinement	**84.5%**

To evaluate the contribution of norms in the objective function, we perform an ablation study and report the performance of the same network without any regularizers. Compared with the full model, the performance of the regularizer-free network drops 1.5% on UCF-101 and 0.9% on CCV (the second last row of Table II).

3) The Hybrid Framework: We now discuss the effectiveness of the entire hybrid deep learning framework. In particular, we linearly average classification scores computed from the two LSTM models and the feature fusion network, which offers promising results, a mean accuracy of 92.1% on UCF-101 and an mAP of 84.0% on CCV (shown in Table III), outperforming alternative methods by clear margins. The entire hybrid framework improves 3.4 and 7.3 percentage points over the regularized fusion network (in Table II) on UCF-101 and CCV respectively, which stems from the combination with temporal

Fig. 3. Two example videos of class "cat" in the CCV dataset with similar temporal clues over time.

clues captured by LSTM models. It is worth noting that our framework also achieves better performance than simple late fusion method (last row in Table I), which performs fusion with the same set of features.

For categories like "graduation" and "birthday" party in CCV, it is easy to understand that the fusion with temporal clues could assist recognition. We also examine other categories like "cat" and "dog" to see if there are certain temporal patterns. Interestingly, as illustrated in Fig. 3, we found many "cat" videos depicting a cat chasing objects or laser on the floor. Though the temporal order is not explicit, it could be captured by LSTM model for improved performance.

Finally, we refine the prediction scores from the hybrid framework using semantics context. The result are summarized in the last row of Table III. The contextual refinement is easy to perform but very effective, offering 1.0% and 0.5% performance gain over the original prediction scores. This confirms our assumption that related classes can assist the recognition of a class of interest. In addition, we also compare with DASD [21], which utilizes context in a graph diffusion framework. Our context modeling method outperforms DASD by 0.7 and 0.3 percentage points with much lower computational complexity on UCF-101 and CCV, respectively.

We further demonstrate per-class average precision after contextual refinement on CCV in Fig. 4. As can be seen from the figure, contextual refinement improves over the original model for nearly all classes. In addition, for classes with lower performance like "bird" and "wedding reception", the performance gains are more significant, resulting from the useful information borrowed from related classes.

4) Speed Efficiency: To investigate the efficiency of our framework, we report the average time to classify a UCF-101 video clip using a single NVIDIA Telsa K40 GPU once the network is trained. Given a video clip, it takes around 4.5 seconds to compute RGB frames, optical flow images and audio spectrograms. The extraction of spatial, motion and audio CNN features takes 12 seconds. Finally, computing and refining the prediction scores from the LSTM and the feature fusion network can be finished in 4.3 seconds.

5) Comparison with State of the Arts: We also compare with several state-of-the-art results on both datasets. Results are summarized in in Table III. We can see from the table that the proposed hybrid deep learning framework produces strong performance on both datasets. Different from works that obtain competitive results on UCF-101 using dense trajectory features

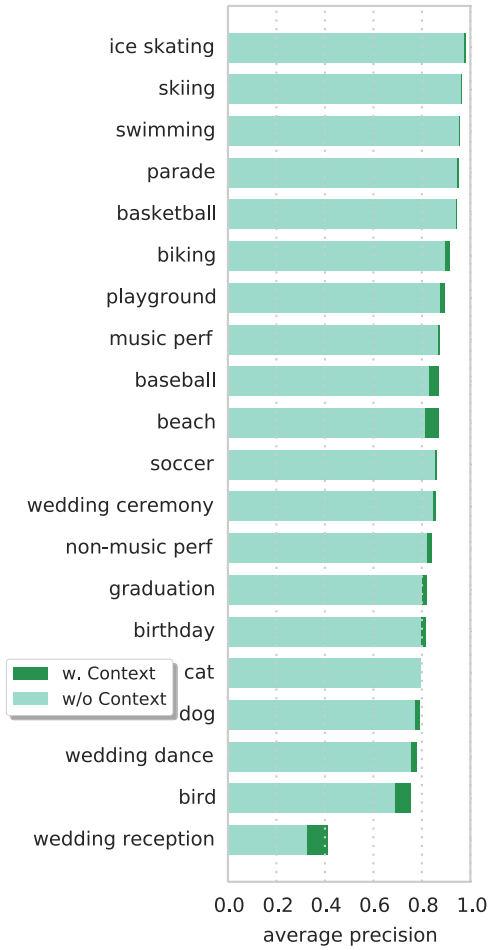


Fig. 4. Per-class average precision with and without contextual refinement on CCV.

[53], [64], our framework is built upon neural networks with an aim to learn feature representations. Our proposed approach improves the original twos stream CNN by incorporating temporal and audio modeling as well as better fusion methods. Notice that a few recent approaches also leverage temporal information with LSTMs [8], [45]; they utilized different CNN models to compute features, and hence the results are not directly comparable. Notice that we expect further performance improvements with more advanced neural networks like ResNet on UCF-101 [9], [55]. On the CCV dataset, the proposed framework outperforms all the recent approaches that are designed to perform fusion by clear margins [15], [31], [33], [58], [60], [63].

V. Conclusion

In this paper, we have proposed a novel hybrid deep learning framework to integrate a comprehensive set of multimodal clues for video categorization. More specifically, we utilize three independent CNN models operating on static frames, stacked optical flow images and audio spectrograms to compute spatial, motion and audio features, respectively. In order to utilize the long-range temporal clues in videos, we apply two LSTM models with the spatial and motion features as inputs. Since different features characterize the same video from different perspectives, we

employ a regularized feature fusion network that derives a unified feature representation for recognizing video semantics. Finally, we also refine the classification scores, the linear combination of LSTM models and feature fusion network, with semantic contextual relationships.

Through an extensive set of experiments on two challenging benchmarks, we demonstrate that (1) the LSTMs, modeling the long-range temporal information in video sequences through an explicitly recurrent manner, are highly complementary with CNNs; (2) the rich contextual relationships among video semantics in a simple yet effective way to further refine predictions for improved performance. The experimental results provide strong quantitative evidence that our framework achieves promising results, outperforming competing methods with clear margins.

REFERENCES

[1] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Proc. Neural Netw., Tricks Trade.*, 2012, pp. 437–478.

[2] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. BMVC*, 2014, pp. 54.1–54.12.

[3] X. Chen and A. Gupta, "Webly supervised learning of convolutional networks," in *Proc. Int. Conf. Comput. Vis*, 2015, pp. 1431–1439.

[4] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Gated feedback recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2067–2075.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 886–893.

[6] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 428–441.

[7] J. Deng *et al.*, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[8] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2625–2634.

[9] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1933–1941.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.

[11] A. Graves, A. Mohamed, and G. E. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Conf. Acoust., Speech Signal Process.*, 2013, pp. 6645–6649.

[12] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *in Proc Int. Conf. Neural Netw.*, 2005, pp. 2047–2052.

[13] X. Han, B. Singh, V. Morariu, and L. S. Davis, "VRFP: On-the-fly video retrieval using web images and fast fisher vector products," *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1583–1595, Apr. 2017.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[15] I.-H. Jhuom *et al.*, "Discovering joint audio-visual codewords for video event detection," *Mach. Vis. Appl.*, vol. 25, no. 1, pp. 33–47, 2014.

[16] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[17] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.

[18] Y.-G. Jiang, S. Bhattacharya, S.-F. Chang, and M. Shah, "High-level event recognition in unconstrained videos," *Int. J. Multimedia Inf. Retrieval*, vol. 2, no. 2, pp. 73–101, 2013.

[19] Y.-G. Jiang, Q. Dai, T. Mei, Y. Rui, and S.-F. Chang, "Super fast event recognition in internet videos," *IEEE Trans. Multimedia*, vol. 17, no. 8, pp. 1174–1184, Aug. 2015.

[20] Y.-G. Jiang *et al.*, "THUMOS challenge: Action recognition with a large number of classes," 2014. Online. Available: http://crcv.ucf.edu/THUMOS14/

[21] Y.-G. Jiang, J. Wang, S.-F. Chang, and C.-W. Ngo, "Domain adaptive semantic diffusion for large scale context-based video annotation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 1420–1427.

[22] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, and S.-F. Chang, "Exploiting feature and class relationships in video categorization with regularized deep neural networks," *IEEE J. Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 352–364, Feb. 2018.

[23] 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 *Proc. 1st ACM Int. Conf. Multimedia Retrieval.*, 2011, Art no. 29.

[24] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1725–1732.

[25] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3D-gradients," in *Proc. BMVC*, 2008, pp. 99.1–99.10.

[26] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien, "Lp-norm multiple kernel learning," *J. Mach. Learn. Res.*, vol. 12, pp. 953–997, 2011.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. 25th Int Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[28] K.-T. Lai, F. X. Yu, M.-S. Chen, and S.-F. Chang, "Video event detection by inferring temporal instance labels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2251–2258.

[29] Z. Lan, M. Lin, X. Li, A. G. Hauptmann, and B. Raj, "Beyond gaussian pyramid: Multi-skip feature stacking for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 204–212.

[30] I. Laptev, "On space-time interest points," *Int. J. Comput. Vis.*, vol. 64, no. 2–3, pp. 107–123, 2005.

[31] D. Liu, K.-T. Lai, G. Ye, M.-S. Chen, and S.-F. Chang, "Sample-specific late fusion for visual category recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 803–810.

[32] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[33] A. J. Ma and P. C. Yuen, "Reduced analytic dependency modeling: Robust fusion for visual recognition," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2014.

[34] M. Nagel, T. Mensink, and C. G. M. Snoek, "Event fisher vectors: Robust encoding visual diversity of visual streams," in *Proc. BMVC*, 2015, pp. 178.1–178.12.

[35] J. Y.-H. Ng *et al.*, "Beyond short snippets: Deep networks for video classification," In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4694–4702.

[36] J. Ngiam *et al.*, "Multimodal deep learning," in *Proc. 28th Int. Conf. Mach Learn.*, 2011, pp. 689–696.

[37] D. Oneata, J. Verbeek, and C. Schmid, "Action and event recognition with fisher vectors on a compact feature set," in *Proc. IEEE Int. Conf. Comput. Vis.* , 2013, pp. 1817–1824.

[38] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie, "Objects in context," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[39] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop*, 2014, pp. 512–519.

[40] Y. Shi, Y. Tian, Y. Wang, and T. Huang, "Sequential deep trajectory descriptor for action recognition with three-stream CNN," *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1510–1520, Jul. 2017.

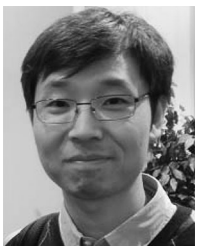
[41] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 568–576.

[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015.

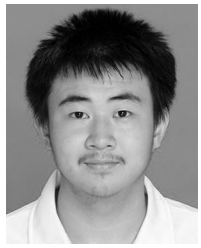
[43] C. G. Snoek, M. Worring, and A. W. Smeulders, "Early versus late fusion in semantic video analysis," in *Proc. 13th Annu. ACM Conf. Multimedia*, 2005, pp. 399–402.

[44] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," in *CoRR*, arXiv.org, vol. cs.CV., pp. 1–7, 2012.

[45] N. Srivastava, E. Mansimov, and R. Salakhutdinov, "Unsupervised learning of video representations using LSTMs," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 843–852.
[46] N. Srivastava and R. Salakhutdinov, "Multimodal learning with deep boltzmann machines," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 2222–2230.
[47] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.
[48] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.
[49] K. Tang, L. Fei-Fei, and D. Koller, "Learning latent temporal structure for complex event detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1250–1257.
[50] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "C3D: Generic features for video analysis," in *Proc. Int Conf. Comput. Vis.*, 2015, pp. 4489–4497.
[51] D. L. Vail, M. M. Veloso, and J. D. Lafferty, "Conditional random fields for activity recognition," in *Proc. 6th Int. Joint Conf. Auton. Agents Multiagent Syst.*, 2007, pp. 235:1–235:8.
[52] A. Van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in *Proc. 26th Int. Conf. Neural Inf. Process Syst.*, 2013, pp. 2643–2651.
[53] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 3551–3558.
[54] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4305–4314.
[55] L. Wang *et al.*, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. Eur Conf. Comput. Vis.*, 2016, pp. 20–36.
[56] X. Wang, A. Farhadi, and A. Gupta, "Actions transformations," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2658–2667.
[57] Y. Wang and G. Mori, "Max-margin hidden conditional random fields for human action recognition," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 872–879.
[58] 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 *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 167–176.
[59] 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 *Proc. 23rd ACM Int. Conf. Multimedia*, 2015, pp. 461–470.
[60] Z. Xu, Y. Yang, I. Tsang, N. Sebe, and A. Hauptmann, "Feature weighting via optimal thresholding for video analysis," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 3440–3447.
[61] Y. Yang *et al.*, "Multi-feature fusion via hierarchical regression for multimedia analysis," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 572–581, Apr. 2013.
[62] L. Yao *et al.*, "Describing videos by exploiting temporal structure," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4507–4515.
[63] G. Ye, D. Liu, I.-H. Jhuo, and S.-F. Chang, "Robust late fusion with rank minimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3021–3028.
[64] S. Zha, F. Luisier, W. Andrews, N. Srivastava, and R. Salakhutdinov, "Exploiting image-trained CNN architectures for unconstrained video classification," in *Proc. BMVC*, 2015, pp. 60.1–60.13.
[65] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *Int. J. Comput. Vis.*, vol. 73, no. 2, pp. 213–238, 2007.



Yu-Gang Jiang received the Ph.D. degree in computer science from the City University of Hong Kong, Kowloon, Hong Kong, in 2009. During 2008–2011, he was with the Department of Electrical Engineering, Columbia University, New York, NY, USA. He is currently a Professor of Computer Science with Fudan University, Shanghai, China. His research interests include computer vision and multimedia.



Zuxuan Wu received the B.E. and M.Sc. degrees from East China Normal University, Shanghai, China, and Fudan University, Shanghai, China, in 2013 and 2016, respectively. He is currently working toward the Ph.D. degree in computer science at the University of Maryland, College Park, College Park, MD, USA. His research interests include computer vision, multimedia, and deep learning.

Jinhui Tang received the B.E. and Ph.D. degrees from the University of Science and Technology of China Hefei, China, in 2003 and 2008, respectively. He is currently a Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. From 2008 to 2010, he was a Research Fellow with the School of Computing, National University of Singapore. His current research interests include large scale multimedia search.

Zechao Li received the B.E. degree from the University of Science and Technology of China, Hefei, China, in 2008, and the Ph.D. degree from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2013. He is currently a Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. His research interests include big multimedia data analysis and computer vision.

Xiangyang Xue received the B.S., M.S., and Ph.D. degrees in communication engineering from Xidian University, Xi'an, China, in 1989, 1992, and 1995, respectively. He is currently a Professor of Computer Science with Fudan University, Shanghai, China. His research interests include multimedia and machine learning.

Shih-Fu Chang is the Richard Dicker Professor, the Director of the Digital Video and Multimedia Laboratory, and a Senior Vice Dean with the Engineering School, Columbia University, New York, NY, USA. Prof. Chang is a Fellow of the American Association for the Advancement of Science and ACM.