IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 30, 2020, accepted June 17, 2020, date of publication June 22, 2020, date of current version July 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3003939

Recurrent Compressed Convolutional Networks for Short Video Event Detection

PING LI, (Member, IEEE), AND XIANGHUA XU
School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China

Corresponding author: Ping Li (patriclouis.lee@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61872122 and Grant 61502131, in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LY18F020015, and in part by the Open Project Program of the State Key Laboratory of CAD and CG, Zhejiang University, under Grant A1802.

ABSTRACT Short videos are popular information carriers on the Internet, and detecting events from them can well benefit widespread applications, e.g., video browsing, management, retrieval and recommendation. Existing video analysis methods always require decoding all frames of videos in advance, which is very costly in time and computation power. These short videos are often untrimmed, noisy and even incomplete, adding much difficulty to event analysis. Unlike previous works focusing on actions, we target short video event detection and propose *Recurrent Compressed Convolutional Networks* (RCCN) for discovering the underlying event patterns within short videos possibly including a large proportion of non-event videos. Instead of using the whole videos, RCCN performs representation learning at much lower cost within the compressed domain where the encoded motion information reflecting the spatial relations among frames can be easily obtained to capture dynamic tendency of event videos. This alleviates the information incompleteness problem that frequently emerges in user-generated short videos. In particular, RCCN leverages convolutional networks as the backbone and the Long Short-Term Memory components to model the variable-range temporal dependency among untrimmed video frames. RCCN not only learns the common representation shared by the short videos of the same event, but also obtains the discriminative ability to detect dissimilar videos. We benchmark the model performance on a set of short videos generated from publicly available event detection database YLIMED, and compare RCCN with several baselines and state-of-the-art alternatives. Empirical studies have verified the preferable performance of RCCN.

INDEX TERMS Compressed domain, event analysis, recurrent neural networks, short video event detection, temporal dependency.

I. INTRODUCTION

In the era of big data, the popularity of portable devices including smart mobile phones makes it quite easy to share videos on the Internet, resulting in a massive increase of video data. It is interesting and important to know what has happened in the videos. Some of these videos may contain only simple gestures like clapping, running, smiling, which are regarded as *actions*. The other videos may involve special occasions or complex activities, such as birthday party, parade and wedding ceremony, which are referred to as *events*, while a large proportion of the videos do not convey any important information such as cluttered background, which are regarded as *nonevent* videos. As a matter of fact, events are more complex compared with actions, since different actions may belong to the same event. For instance, as shown in Figure 1 which selects three scenes in a birthday party video, we can see a series of different actions including blowing out candles, eating cake, and smiling, which have almost the same background. Previous works mostly focus on actions [1]–[3] while neglecting events which are usually composed of sequential actions with both spatial and temporal relations. In this work, we primarily concentrate on the event analysis in the short videos.

Generally speaking, people are more interested in discovering event videos which are drown in a large number of nonevent videos. Moreover, in most social media networks, e.g., Instagram, Twitter and WeChat, users are allowed to upload only short videos (say less than 30 seconds) which may occasionally include incomplete event scenes. Based on

The associate editor coordinating the review of this manuscript and approving it for publication was Li He.

FIGURE 1. Illustration of a happy *birthday* event in a short video. left: blowing out candle; middle: eating cake; right: smiling.

these observations, we are inspired to explore a novel topic, i.e., *Short Video Event Detection* (SVED), which aims to detect event videos and identify their high-level event labels within short video length.

To this end, there are several difficulties in learning a model to detect event videos from a huge number of short video sequences. First, it is costly to process videos due to the large number of continuous frames. Second, these user-generated videos are often untrimmed and unconstrained, which means some frames or segments that are irrelevant to one specific event will mislead and harm learning effective models. Third, it is much more difficult to capture the spatial and temporal relations for complex events within short videos, compared to long videos. These above issues collectively make short video event detection a rather challenging and tough task in both academic and industrial communities.

Most of video analysis methods require decoding the frames from the video as pre-processing which is very time-consuming. Actually, videos as one kind of media with low information density are essentially stored in compressed form, and thus can be directly used. This motivates us to explore the efficient way of using compressed videos for short video event detection, largely reducing the cost of processing a great many frames. Modern video compression techniques [4] like MPEG-4, H.264 and HEVC take advantage of a fact that there exists much redundancy among successive frames which share similar visual information. Hence, it makes sense to retain only a small number of complete images which can reconstruct the remaining by using the offsets [5]. Those offsets usually refer to motion vectors and residual differences which are available at very low computational cost. While motion vectors that indicate pixel block movements are invariant in the spatial dimension, they are robust to spatial variation for the same event, e.g., wedding ceremony in light-varying conditions. This is beneficial for recognizing unconstrained videos containing various scenes. Residuals, reflecting the differences between the consecutive *P-frames* (predicted frames) and the reference *I-frame* (intra-coded frames), can well encode the contour of some motion regions of interest, which plays an important role in short video event analysis. Especially, short videos only have limited frames, in which some are not that relevant to the event and even incomplete in disadvantageous situations. To alleviate this problem, we adopt an accumulated strategy to fuse the information embedded in motion vectors and residuals since such a kind of accumulation is able to capture the

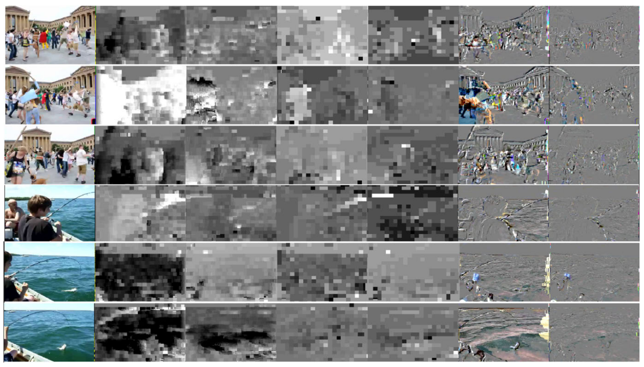


FIGURE 2. Illustration of short video events. 1st col: P-frames; 2nd/4th cols: x/y-axis accumulated motion vectors; 3rd/5th cols: x/y-axis motion vectors; 6th col: accumulated residuals; 7th col: residuals. The top 3 rows show *flash mob* and the bottom 3 rows show *person landing a fish*.

temporal structures as well as appearance changes of event frames. Thus the dynamic information encoded by motion vectors can be leveraged for reflecting the motion tendency of pre-defined events in user-generated short videos. Therefore, we propose to detect events within short videos in the compressed domain.

To achieve the above goal, we develop a **R**ecurrent **C**ompressed **C**onvolutional **N**etworks (RCCN) architecture to directly learn effective representation of short videos. This architecture leverages two kinds of encoded frames including I-frames and P-frames in the compressed video, where I-frames are regular frame images while P-frames encode the change information by referencing the previous frames. One part of the change information denotes the movements of pixel blocks from one frame to another, named motion vectors, and the other part is the residuals which are used to compute the error between the referenced frame and the predicted frames. The two kinds of P-frames are illustrated in Figure 2, where the motion vectors reveal the pixel block changes and the accumulated residuals reflect motion boundaries and coarse object contours. To learn the higher-level representations of these I-frames and P-frames, they are respectively fed into recurrent convolutional neural networks which are composed of residual Convolutional Neural Network (CNN) module and Recurrent Neural Network (RNN) module using Long Short-Term Memory (LSTM) [6], [7] units. Our architecture takes advantage of the CNN module to capture the spatial relations among consecutive frames, and meanwhile makes use of the RNN module to preserve the temporal structure of those compressed short videos. This allows us to effectively model the spatiotemporal dependency among continuous frames for a given short video, resulting in more satisfying performance of event detection. Since this is a pioneering work on short video event detection and there is no available database, we use a data set generated from a publicly available multimedia event detection database YLIMED [8], [9]. Comprehensive experiments are conducted to demonstrate the superiority of the proposed architecture.

In this work we make the following main contributions:

- We introduce an interesting and important topic, i.e., Short Video Event Detection (SVED), which targets at detecting events from the videos with only a few seconds, sometimes noisy and incomplete. Since short videos increase sharply in a large volume in our daily life, especially in social media, this brings us a big challenge to effectively manage and analyze them. SVED can provide a promising way of helping to better sense what reveals in those short videos.
- We design an efficient architecture named Recurrent Compressed Convolutional Networks (RCCN) for the proposed SVED task, which directly handles short videos in the compressed domain, allowing it to capture the dynamic tendency of events and behave as one scalable approach for real-time deployment. Unlike traditional methods requiring decoding all frames in the video, RCCN can directly leverage some compressed information including I-frame, motion vectors, and residuals for learning satisfactory representation in the data space of short videos.
- For the compressed I-frames and P-frames of short videos, RCCN can utilize both CNN module and RNN module to intrinsically capture their spatiotemporal structure as accurate as possible. Therefore, the common representation shared by the short videos within the same event can be well captured and also the discriminative ability can be learned better, thus promoting better understanding of the contents indicated in the videos.
- We newly establish a data set for the SVED task from an open multimedia database, and evaluate several primary backbone deep learning models as baselines. Several state-of-the-art 3D models for video analysis are also investigated on the new database, and experimental results have verified the advantages of RCCN, which can better detect events from short videos in an efficient way.

II. RELATED WORK

The SVED task is closely related to traditional multimedia event detection [10], [11] and action recognition [3], [12], [13], which have attracted lots of attention in recent years. The former focuses on the videos lasting for several minutes or even hours while the latter tackles video clips including simple actions. In contrast, our SVED targets at short videos with the limited duration (e.g., less than 30 seconds) where there exist sequential actions likely involving quite a few objects across various scenes which are defined as *events*. These short videos uploaded onto social media networks by users are often captured in uncontrolled environments, possibly resulting in noisy, blurry, jittering and even incomplete sequences, further adding the difficulty of dealing with the task. These problems still remain unsolved and open to the community, and this work attempts to handle this challenging problem. To this end, many existing video understanding methods have paved a way to tackling SVED tasks, e.g., learning spatiotemporal representation using 3D convolutional networks [14] or pseudo-3D residual networks [15], utilizing the rich multimodal information in videos to train CNNs and using LSTMs to explore long-term temporal dynamics [16], and interpreting temporal dependencies between video frames at multiple time scales [17].

Multimedia event detection aims to identify the likelihood of a given video belonging to some event of interest by ranking the scores, which is similar to event recognition [18], [19] that classifies a given video to pre-defined event categories. This work considers the SVED task in the presence of a large proportion of non-event videos, which is usually the case in practical applications. A typical event detection system consists of feature extraction and classification components by using hand-crafted features (e.g., improved dense trajectory [20]) and a classifier, respectively. Recently, deep learning [21] becomes prevalent in computer vision and has exhibited great success in video understanding [22] since deep features are able to learn much better representation of the data compared to low-level features. For example, [23] proposed a discriminative CNN video representation method to boost event detection. To enhance the discriminative video representation, [24] developed a semantic pooling method for event analysis in long untrimmed Internet videos. Reference [25] also tried to learn a bi-level semantic representation from different multimedia archives at source level while reducing the negative influence of noisy or irrelevant concepts at concept level for event detection. To further leverage semantic information of videos, [26] proposed a hierarchical video event detection model that unifies the processes of underlying semantics discovery and event modeling from video data. Reference [27] attempted to learn semantic attributes from external videos using their semantic labels. Besides, some works unify the complex event detection and evidence recounting for video events. For instance, [28] proposed a joint framework to detect high-level events while meanwhile localizing the indicative concepts of the events. Reference [29] utilized key frames of a video as inputs to simultaneously detect pre-defined events and provide key spatial-temporal evidences. In addition, there exist some works addressing the problem of zero-shot event recognition in consumer videos. For example, [30] provided a fully automatic way to select representative and reliable concepts for event queries by discovering event composition knowledge from web images. While most previous works consider visual information, some works utilize the audio information of videos to detect events. In [31], discriminative and compact audio representation was proposed to respect the structure of audio signals for event detection.

The other topic related to our work is action recognition which has also received much attention in recent years. Most popular approaches are based on CNNs. One of them is two-stream CNNs with 2D convolutional kernels. For example, [32] fed both RGB and stacked optical flow frames into respective CNN for classification, but their work can not

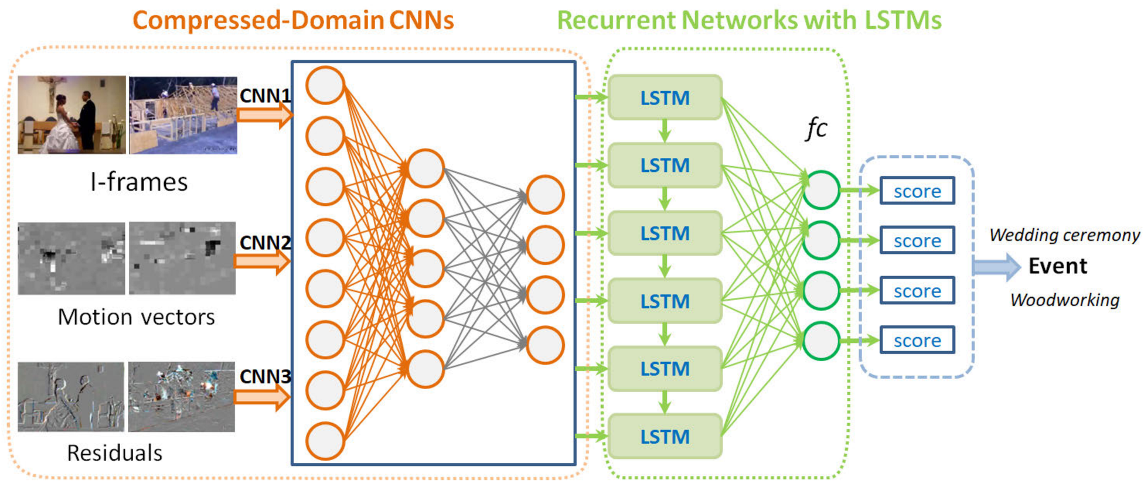


FIGURE 3. The RCCN architecture. RCCN consists of two components, i.e., compressed-domain CNNs in the left part and recurrent networks with LSTMs in the right part. Unlike traditional methods requiring decoding all the frames in the video, RCCN directly utilizes I-frames, motions vectors, and residuals in the compressed domain as the inputs of CNNs, which hugely reduces the computational cost of video processing, making it possible to deploy in highly demanding applications. Moreover, as mentioned earlier, events are more complex than actions as an event often consists of continuous actions, which makes it more difficult to model the temporal dependency in the short videos. Thus, recurrent networks with LSTMs are introduced to alleviate this problem by encoding the correlations of those frames within the same event video along the temporal dimension. Here, CNN1 accepts the I-frams, CNN2 accepts motion vectors, CNN3 accepts residuals, '*fc*' denotes fully-connected layer.

satisfy the demanding requirement of real-time process on videos because of the intensive computation of optical flow. For real-time application, [5], [33] adopted motion vectors instead of optical flow to accelerate training of the model. Since 2D methods fail to fully consider the temporal relations among frames, it is a heuristic way to take advantage of RNN module such as typical LSTM units to reflect the long-range temporal dependencies among continuous frames in videos. For instance, [34] developed a long-term recurrent neural network using a deep hierarchical feature extractor with LSTM networks to synthesize temporal dynamics for visual recognition and description; [13] learned video representations using neural networks with long-term temporal convolutions to model actions at full temporal extent; [35] tried to adaptively identify key features of actions in videos for every time-step prediction of RNN by reinforcing LSTM with a spatial-temporal attention module; [7] proposed an attention-based bidirectional LSTM method for video analysis. Moreover, Wang *et al.* [36] modeled long-range temporal structure with segment-based sampling and aggregation strategy; Kim and Won [2] employed stacked gray-scale 3-channel image to fine-tune the pre-trained 2D CNN for the temporal stream in videos. Furthermore, there exist successful attempts of directly applying 3D CNN convolutional networks to action recognition, since 3D filters can learn spatiotemporal representation from raw videos [14], [37]–[39]. In the tests, we examined several typical 3D models on our task.

III. THE PROPOSED APPROACH

This section introduces our RCCN architecture built upon two critical blocks: compressed-domain CNNs and recurrent networks with LSTMs, as illustrated in Figure 3.

A. COMPRESSED-DOMAIN CNNs

Efficient video processing is of vital importance in many demanding applications, such as social media content analysis. RCCN directly handles raw videos in the compressed domain, which allows it to be deployed in real-time situations. In the video coding field, consecutive frames are often organized as groups of pictures named GOPs. One typical GOP usually consists of intra-coded I-frames, predictive P-frames and bi-directional B-frames, among which the former two sorts of frames are employed here. I-frames are actually regular images while P-frames encode the changes by referencing previous frames. One kind of changes called *motion vectors* $\mathcal{T}^{(t)}$ can reveal the movement patterns of a pixel block from the source frame to the target frame at time t while the other kind is *residuals* $\Delta^{(t)}$ that encode the differences between the original frame and the predicted frame. Motion vectors exploit the temporal relation in neighboring frames by recording how the pixel block moves. Since the information of motion vectors is coarse and thus insufficient for analyzing short videos, it is natural to utilize the residuals of frames to compensate for such deficiency. Mathematically, P-frames can be reconstructed by recurrently using motion vectors and residuals as [5]

$$I_i^{(t)} = I_{i-\mathcal{T}_i^{(t)}}^{(t-1)} + \Delta_i^{(t)}, \quad t = 1, 2, \ldots, T, \tag{1}$$

where i indexes the pixel position of one frame, T is the time duration length, $I_i^{(t)}$ represents the i-th pixel value of image at time t, $\mathcal{T}_i^{(t)}$ indicates the movement of the i-th pixel position for a pixel block of the image, and $\Delta_i^{(t)}$ specifies the residual of pixel i of the image.

Our task is to detect events from short videos and it is challenging to utilize the compressed video frames directly.

Since I-frames, motion vectors and residuals are characterized by different properties and structures, we feed them into three different convolutional neural networks to learn spatial representations. The P-frame depends on the reference frame which may again reference another until encountering a preceding I-frame, and thus we adopt the *back-tracking* technique [5] to decouple individual P-frames so that all motion vectors can be traced back to the reference I-frame and meanwhile the residuals are accumulated during the process. In this way, every P-frame is relevant with only one I-frame rather than other P-frames. Details about this technique are elaborated in the following.

Given the i-th pixel of a frame at time t, assume $\mu_{\mathcal{T}^{(t)}(i)} = i - \mathcal{T}_i^{(t)}$ is its reference location in the previous frame. Then the updated location tracing back to those frames $I^{(k)}$ $(k < t)$ is $\mathcal{J}_i^{(t,k)} = \mu_{\mathcal{T}^{(k+1)}} \circ \cdots \circ \mu_{\mathcal{T}^{(t)}}(i)$. At time t, the accumulated motion vector $\mathcal{D}^{(t)} \in \mathbb{R}^{H\times W\times 2}$ and the accumulated residuals $\mathcal{R}^{(t)} \in \mathbb{R}^{H\times W\times 3}$ can be respectively expressed as

$$\mathcal{D}_i^{(t)} = i - \mathcal{J}_i^{(t,k)}, \tag{2}$$

$$\mathcal{R}_i^{(t)} = \Delta^{(k+1)}_{\mathcal{J}_i^{(t,k+1)}} + \cdots + \Delta^{(t-1)}_{\mathcal{J}_i^{(t,t-1)}} + \Delta_i^{(t)}. \tag{3}$$

The accumulation vectors are calculated at a very low cost via a light feed-forward in the process of decoding the video. P-frames in (1) are updated as

$$I_i^{(t)} = I^{(0)}_{\mathcal{J}_i^{(t,k)}} + \mathcal{R}_i^{(t)}, \quad t = 1, 2, \ldots, T. \tag{4}$$

For short videos with noise or jittering, the back-tracing can bring robustness since the accumulated signals can model the spatial structure of frames in a long range, leading to promising performance on the SVED task.

The I-frames and P-frames (e.g., motion vectors and residuals) decoded from compressed short videos are passed through the feature learning module, i.e., convolutional neural networks here, by data transformation $\phi_\Gamma(\cdot)$ with visual frames as inputs and Γ as parameters. For compressed videos in our architecture, these frames would reside in some GOP, which consists of one leading I-frame and subsequent P-frames with the length of T, i.e.,

$$I_{GOP} := \{I^{(0)}, \mathcal{D}^{(t)}, \mathcal{R}^{(t)}\}, \quad t = 1, 2, \ldots, T. \tag{5}$$

To equip the feature of video frames with more representative and discriminating power, we feed these frames stacked as vectors into CNNs with recurrent networks in a supervised manner. As mentioned earlier, the three kinds of frames have diverse characteristics that respect the spatiotemporal structure of the data in different ways, so they will be sent to three CNNs, respectively. In particular, the learned features $\mathbf{x} \in \mathbb{R}^m$ (m is the feature dimension) of the I-frame, accumulated motion vectors and accumulated residuals can be obtained by

$$\mathbf{x}^{(0)}_{iframe} = \phi_{\Gamma_{cnn}}(I^{(0)}), \tag{6}$$

$$\mathbf{x}^{(t)}_{motion} = \phi_{\Gamma_{cnn}}(\mathcal{D}^{(t)}), \tag{7}$$

$$\mathbf{x}^{(t)}_{residual} = \phi_{\Gamma_{cnn}}(\mathcal{R}^{(t)}), \tag{8}$$

where $\phi_{\Gamma_{cnn}}(\cdot)$ serves as a feature mapping function of CNNs. These produced features are good at modeling the spatial relations of frames in videos, but they are still insufficient for capturing the temporal dependencies among frames, which are essential for event video understanding. Hence, we introduce recurrent networks into our architecture to strengthen the temporal dynamics modeling for the features learned through convolutional neural networks.

B. RECURRENT NETWORKS WITH LSTMs

To effectively model the temporal structure of the representations learned from CNNs, we employ one typical RNN that uses Long-Short Term Memory (LSTM) [6] units to accept these representations as input sequences for mapping to hidden states, which are then regarded as the output after data transformations. The crucial property of LSTMs is that they can incorporate memory units, which explicitly allow them to learn how much information from previous hidden states should be stored and to control the time to update hidden states when new information arrives. The successful applications of LSTMs in various tasks such as audio analysis, machine translation, and video captioning have shown its powerful ability to model sequential data of variable lengths, so it is expected to enable well modeling of the enhanced temporal structure of short video sequences for event detection.

The most important component of LSTM networks is memory cell modulated by nonlinear sigmoid gate function $\sigma(\mathbf{x}) = \frac{1}{1+e^{-\mathbf{x}}}$ which squashes real-valued inputs to a range from 0 to 1. The gates can decide how much information is kept at every time step when taking in newly arrived feature vectors. For a basic LSTM module in our architecture, it is composed of three main gates, one memory cell and one hidden unit. The three gates are the input gate $\mathbf{i}_t \in \mathbb{R}^d$ that controls whether to use its current input $\mathbf{x}^{(t)} \in \mathbb{R}^m$, the forget gate $\mathbf{f}_t \in \mathbb{R}^d$ that selectively forgets its previous memory, and the output gate $\mathbf{o}_t \in \mathbb{R}^d$ that learns how much of the memory cell $\mathbf{c}_t \in \mathbb{R}^d$ is transferred to the hidden unit $\mathbf{h}_t \in \mathbb{R}^d$. As a result, the history and the current information can be both selectively incorporated for modeling time-varying temporal dynamics in sequence learning. To update the gates, memory cell and hidden states at time t, we assume the weight matrix is $\mathbf{W} \in \mathbb{R}^{d\times m}$ while the bias vector is $\mathbf{b} \in \mathbb{R}^d$, and then use the equations below:

$$\mathbf{i}_t = \sigma(\mathbf{W}_{xi}\mathbf{x}^{(t)} + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{b}_i), \tag{9}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{xf}\mathbf{x}^{(t)} + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{b}_f), \tag{10}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{xo}\mathbf{x}^{(t)} + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{b}_o), \tag{11}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{t}_t \odot \tanh(\mathbf{W}_{xc}\mathbf{x}^{(t)} + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c), \tag{12}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \tag{13}$$

where $\tanh(\cdot) = 2\sigma(\cdot) - 1$ is the hyperbolic tangent nonlinearity that squashes its inputs to a range from -1 to 1.

The number of LSTM units *a.k.a.* the layer depth can be varied by stacking them on top of each other, i.e., the hidden state of the previous LSTM unit is the input of the next unit. A series of continuous frames passing one LSTM unit sequentially means they are fed into one LSTM layer, and LSTM networks can have several layers as desired in real-world applications. For our SVED task, the inputs of the LSTM layer are the learned representations from previous CNNs, and the outputs of LSTM networks can be expressed as

$$\mathbf{x}'^{(0)}_{iframe} = \phi_{\Gamma_{lstm}}(\mathbf{x}^{(0)}_{iframe}), \tag{14}$$

$$\mathbf{x}'^{(t)}_{motion} = \phi_{\Gamma_{lstm}}(\mathbf{x}^{(t)}_{motion}), \tag{15}$$

$$\mathbf{x}'^{(t)}_{residual} = \phi_{\Gamma_{lstm}}(\mathbf{x}^{(t)}_{residual}), \tag{16}$$

where $\phi_{\Gamma_{lstm}}(\cdot)$ is a feature mapping function whose parameters include weights and bias of LSTM networks.

C. THE RCCN ARCHITECTURE

Our RCCN architecture incorporates the above two neural network modules together to model the spatiotemporal structures and dependencies of the short videos for event detection in a principled way.

Given a set of n short videos $\mathbb{S}_V = \{V_i\}_{i=1}^n$ where each video is denoted by a tensor $V_i \in \mathbb{R}^{w\times h\times c}$, we first extract the compressed I-frames and P-frames directly, where I-frames can be regarded as regular RGB images $I^{(0)}$ while P-frames are used to compute the accumulated motion vectors $\mathcal{D}^{(t)}$ and the accumulated residuals $\mathcal{R}^{(t)}$ at time t. Note that here these frames are readily available from the compressed videos at a very low cost, which are suitable for processing a large number of short videos. Having obtained the source inputs, these three kinds of visual frames are reshaped in the tensor form and fed into thee different CNNs respectively, because they have diverse properties in reflecting the spatiotemporal relations embedded in videos. From the convolutional neural networks, our model can learn relatively high-level representation of these compressed frames through convolutional kernels (e.g., 3×3 filters), activation functions (e.g., ReLU), batch normalization, and pooling layers (e.g., max-pooling), resulting in corresponding transformed features $\mathbf{x}^{(0)}_{iframe}$, $\mathbf{x}^{(t)}_{motion}$, and $\mathbf{x}^{(t)}_{residual}$. Moreover, CNNs fail to fully consider the temporal relations among frames in videos, and we thus introduce LSTMs based recurrent networks to model varying-range temporal dependencies of consecutive frames. In this way, the LSTMs module accepts the learned representations from CNNs module and yields the spatiotemporal structured representations $\mathbf{x}'^{(0)}_{iframe}$, $\mathbf{x}'^{(t)}_{motion}$, and $\mathbf{x}'^{(t)}_{residual}$.

In the implementation, the last fully-connected layer of CNNs is removed while the LSTM units are added subsequently, and the outputs of LSTM layers are fed into a fully-connected layer to estimate the distribution of predicted scores. In the training phase, the ground-truth event labels of videos are used for guiding the gradient back-propagating process through LSTMs module and CNNs module in sequence. In the testing phase, a compressed video is fed into the RCCN model which produces the score distribution of event classes by the forward process, and the top-ranked score indicates whether this video contains an event or not as well as which category it belongs to. The main procedures of our method are summarized in Algorithm 1.

Algorithm 1 RCCN for Short Video Event Detection

```
Input:
    Training short videos $\{V_1^{tr}, V_2^{tr}, \ldots, V_n^{tr}\}$.
    Training video event labels $\{y_1^{tr}, y_2^{tr}, \ldots, y_n^{tr}\}$.
    Testing short videos $\{V_1^{te}, V_2^{te}, \ldots, V_q^{te}\}$.
    Learning rate $lr$ and its decay $\eta$, weight decay $\zeta$.
    Batch size $n_{batchsize}$ and the number of epochs $N$.
Output:
    Predicted event labels $\{\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_q\}$ of test videos.
1:  Training phase:
2:  Extract I-frames and P-frames directly from compressed
    short videos $\{V_i^{tr}\}_{i=1}^n$.
3:  Compute accumulated motion vectors $\mathcal{D}$ using (2).
4:  Compute accumulated residuals $\mathcal{R}$ using (3).
5:  for $epoch = 1$ to $N$ do
6:      for $idx = 1$ to $\frac{n}{n_{batchsize}}$ do
7:          Feed I-frames, $\mathcal{D}$, and $\mathcal{R}$ to different CNNs.
8:          Obtain $\mathbf{x}^{(0)}_{iframe}$, $\mathbf{x}^{(t)}_{motion}$, and $\mathbf{x}^{(t)}_{residual}$ by the forward
            pass through convolutional layers.
9:          Obtain $\mathbf{x}'^{(0)}_{iframe}$, $\mathbf{x}'^{(t)}_{motion}$, and $\mathbf{x}'^{(t)}_{residual}$ by the forward
            pass through LSTM layers.
10:         Compute the loss and back-propagate the gradient
            as well as updating the parameters at each layer.
11:     end for
12: end for
13: Testing phase:
14: for $j = 1$ to $q$ do
15:     Repeat Steps 2 to 4 for testing short videos $V_j$.
16:     Repeat Steps 7 to 9 to yield the estimated ranking
        scores for each video.
17:     Top-ranked score induces predicted event label $\tilde{y}_j$.
18: end for
```

IV. EXPERIMENTS

This section first describes the short video data set, and then reports the evaluation results of several basic deep learning models, state-of-the-art 3D convolution models, attention-based model, bidirectional LSTM model, and compressed model, which we compare with our proposed RCCN approach. All the tests are carried out using *FFmpeg* package and *PyTorch* platform on Ubuntu system with Nvidia Titan Xp graphic cards.

A. DATASET

Since this is the first work targeting at short video event detection and there is no available dataset, we establish a short video dataset from one publicly available multimedia event detection data corpus YLIMED [9], where the videos are mostly of several minutes duration. YLIMED is drawn from the Yahoo Flickr Creative Commons 100 Million (YFCC100M) [8] dataset for video event understanding.

TABLE 1. The rule of generating short videos from the long videos with the length of *L* seconds.

Range	[1,10]	(10,20]	(20,30]	(30,60]	(60,∼]
split ratio	L	L/2	L/3	L/4	L/6

TABLE 2. Statistics (number) of short video database.

Sets	Training	Validation	Testing	Total
videos	6,384	739	5,870	12,993
frames	2,008,064	239,594	1,829,057	4,076,715

The events defined here are similar to those in TRECVID MED, and less than 2,000 videos depict one of ten target events while the remaining 48,000 videos belong to none of them. We download all the available videos from the available URLs, and there are 1,822 event videos distributed in ten event categories. The frame rates are diverse among these videos in the set of {4, 5, 10, 12, 12.5, 15, 20, 24, 25, 30}. For our task, we combine all positive event videos and the same number of randomly selected non-event videos together, i.e., 3,644 long videos, and split them into short videos according to the rules in Table 1. In this way, we obtain totally 12,993 short videos, of which 6,384 videos form the training set, 739 videos (randomly sampling 10% from original training split) form the validation set, and 5,870 videos form the test set. The 88.9% of the short videos last less than 15 seconds(s) and the overall average length is 11.28 s. The shortest length is 1.04s and the longest is 38.74 s(very few). The statistics of the newly generated short video database is shown in Table 2.

The event videos are categorized into 10 classes, i.e., Birthday Party (Ev101), Flash Mob (Ev102), Getting a Vehicle Unstuck (Ev103), Parade (Ev104), Person Attempting a Board Trick (Ev105), Person Grooming an Animal (Ev106), Person Hand-Feeding an Animal (Ev107), Person Landing a Fish (Ev108), Wedding Ceremony (Ev109), and Working on a Woodworking Project (Ev110). There a large number of non-event videos, which agrees with the real situation, and these generated short videos are mostly incomplete and noisy, which makes it quite challenging to detect and recognize these event videos.

There are two evaluation criteria for video event detection, i.e., the average event accuracy (ACC) and the overall detection rate (ODR). The former reflects the performance of the model on each event category, i.e., the average result of the accuracy values obtained from all event classes, and the latter is the ratio of successfully detected event videos in all event videos excluding non-event ones.

B. BASELINE COMPARISON

To investigate the performance of basic deep learning models on detecting events from short videos, we conduct several experiments on some base nets including AlexNet [40], VGG [41], ResNet [42], DenseNet [43], and SqueezeNet [44]. In the tests, AlexNet has 5 convolution

TABLE 3. Performance comparisons of typical baseline methods.

Events	AlexNet	VGG	ResNet	DenseNet	SqueezeNet
Ev101	0.5117	**0.5820**	0.5604	0.5730	0.4018
Ev102	0.5401	**0.6257**	0.5829	0.5989	0.5187
Ev103	**0.4744**	**0.4744**	0.4359	0.4423	0.4359
Ev104	0.4254	**0.5033**	0.4900	0.4521	0.3207
Ev105	0.2945	**0.4036**	0.4000	0.3055	0.3018
Ev106	0.4156	0.5130	0.6039	**0.6104**	0.4610
Ev107	0.1600	0.2471	0.2824	**0.2847**	0.1482
Ev108	0.6626	**0.7117**	0.6871	0.6442	0.5828
Ev109	0.5000	0.5921	**0.6360**	0.6653	0.4038
Ev110	0.3085	**0.4876**	0.4577	0.4527	0.2338
ACC	0.4293	**0.5140**	0.5136	0.5029	0.3809
ODR	0.4180	0.5035	**0.5058**	0.4975	0.3562

TABLE 4. Parameter settings of the proposed RCCN architecture.

Parameter	I-frame	Motion vectors	Residuals
lr_{cnn}	3e-4	5e-3	1e-3
lr_{lstm}	3e-4	5e-4	1e-3
hidden units	1024	256	512
step size	[40, 80]	[80, 120]	[80, 110]
epochs	90	160	130
batch size	25	45	45
backbone	ResNet152	ResNet101	ResNet101

(*conv*) layers and 3 full-connected (*fc*) layers; VGG has 16 *conv* layers and 3 *fc* layers; ResNet has 50 layers including four blocks; DenseNet has 121 layers and it is a logical extension of ResNet; SqueezeNet is composed of building bricks called fire modules, each containing two layers: a squeeze layer and an expand layer. Since there exists much redundancy in videos, we randomly select 3 frames from every 25 frames, resulting in 231,678 training images, 14,811 validation images, and 211,245 test images. To obtain the results, we fine-tune basic nets on a pre-trained model using ImageNet. Regarding parameters settings, learning rate is empirically set to 0.001 for AlexNet, SqueezeNet and 0.0001 for the rest; learning rate decay is 0.1; weight decay is 0.0005; momentum is 0.9; epochs are 25 with step size [10, 20] except SqueezeNet using [30, 40] for 50 epochs; the batch size is set to fully filling 2 GPU cards.

The results of baseline models are shown in Table 3. It can be observed that VGG and ResNet perform relatively better than the rest, which indicates that the deeper convolutional layers and residual learning units can enhance the event detection performance compared to others. However, SqueezeNet performs the worst although the number of epochs is twice as many as the remaining base nets, which shows its network structure might be improper for short video event detection.

C. STATE-OF-THE-ART ALTERNATIVES COMPARISON

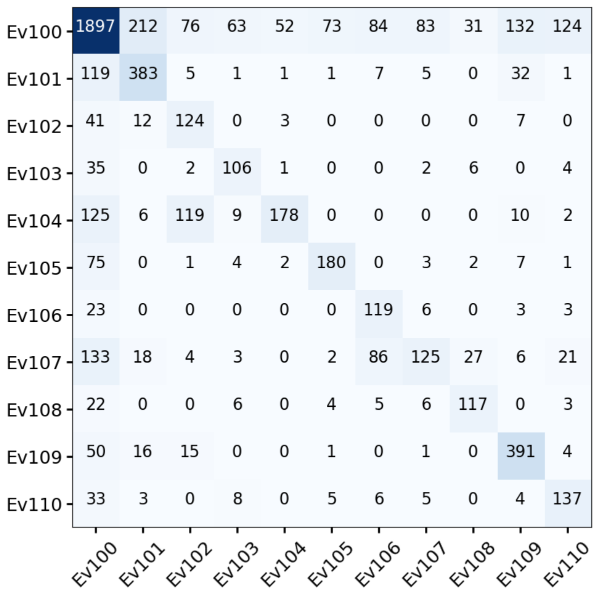
For video understanding, convolutional 3D models have been proved effective in practice. In the tests, we examine C3D [14], P3D [15], and 3D ResNet [39]. The pre-trained model used for C3D is derived from training Sports-1M, and the other two use the pre-trained model on Kinetics. Parameters are set as suggested in original papers or github web sites, i.e., learning rate 0.001, momentum 0.9, learning

TABLE 5. Performance comparisons of RCCN and state-of-the-art alternatives on the short video database ('mv' denotes motion vectors).

Events	ABiLSTM	SG3I	C3D	P3D	3DResNet	CoViAR			RCCN		
						iframe	mv	residual	iframe	mv	residual
Ev101	0.5821	0.6188	0.5405	0.5441	0.5748	0.5946	0.1063	0.4667	**0.6900**	0.3135	0.5495
Ev102	0.5790	0.6234	0.5348	0.5615	0.5829	0.4492	0.2941	0.3904	**0.6631**	0.5026	0.4973
Ev103	0.5265	0.6119	0.6090	0.5513	0.4615	0.4872	0.1154	0.4167	**0.6795**	0.2307	0.4807
Ev104	0.5032	0.5787	0.5813	0.5590	0.5880	**0.6058**	0.3474	0.5056	0.3964	0.5278	0.5880
Ev105	0.6456	0.6222	**0.6873**	0.5818	0.5564	0.5491	0.1455	0.2073	0.6545	0.3381	0.5200
Ev106	0.7086	0.7516	0.4481	0.6429	0.6234	0.7468	0.1364	0.4091	**0.7727**	0.3246	0.6168
Ev107	**0.3692**	0.2904	0.2918	0.2729	0.3576	0.2565	0.0282	0.1553	0.2941	0.1176	0.1505
Ev108	0.6551	0.7027	0.6380	0.6564	0.6933	0.6933	0.3190	0.5215	**0.7178**	0.4907	0.6748
Ev109	0.7224	0.7433	0.6485	0.5356	0.6653	0.6172	0.2197	0.5251	**0.8180**	0.3682	0.5125
Ev110	0.5702	0.6109	0.3731	0.4776	0.5572	0.4478	0.0498	0.1692	**0.6816**	0.2338	0.3681
ACC	0.5862	0.6154	0.5352	0.5383	0.5660	0.5447	0.1762	0.3767	**0.6368**	0.3448	0.4958
ODR	0.5687	0.5793	0.5347	0.5186	0.5613	0.5373	0.1735	0.3878	**0.6112**	0.3407	0.4824

patience 10, 5, 10 respectively, learning rate decay 0.1, weight decay 0.001, 0.0001, 0.001 respectively, sample duration 16, the backbone net being ResNet152 except C3D. Moreover, recently proposed ABi-LSTM [7], SG3I [2], and the compressed method CoViAR [5] originally used for action recognition are introduced for comparison; their parameters are set to default as indicated in their papers and github web site. For our RCCN method, we use temporal segments to capture variable-length dependencies among frames: during training segment size is 5 and set to 25 in testing; the other parameters are shown in Table 4. The performance comparison results of RCCN and state-of-the-art alternatives are recorded in Table 5.

From Table 5, it can be seen that RCCN using I-frames enjoys the most satisfying event detection performance compared to other competing alternatives. This demonstrates RCCN is able to capture the spatial relations among frames better and also good at modeling the temporal dynamics in the short videos, suggesting that it can discover the visual knowledge of events embedded from the short videos in a more sensible way. Meanwhile, we find that RCCN using I-frames can achieve the training speed of 5.2 ms per frame and the testing speed of 4.36 ms per frame, which completely meets the real-time application requirement. Moreover, RCCN using accumulated residuals can achieve almost the same performance of DenseNet at much lower cost, verifying the merits of applying residuals in short video event detection. Furthermore, C3D on Ev105, ABiLSTM on Ev107, and CoViAR on Ev104 rank first, indicating convolutional 3D models are appropriate for handling events like *person attempting a board trick*, attention-based bidirectional LSTM is suited to detect events like *hand-feeding animal* while CoViAR can be used for processing videos of crowd scenes, e.g., *parade*. In addition, the overall performance of SG3I is the best among those alternatives, which demonstrates that the stacked gray-scale 3-channel image possibly substituting optical flows is effective in capturing motion information in the videos.



	Ev100	Ev101	Ev102	Ev103	Ev104	Ev105	Ev106	Ev107	Ev108	Ev109	Ev110
Ev100	1897	212	76	63	52	73	84	83	31	132	124
Ev101	119	383	5	1	1	1	7	5	0	32	1
Ev102	41	12	124	0	3	0	0	0	0	7	0
Ev103	35	0	2	106	1	0	0	2	6	0	4
Ev104	125	6	119	9	178	0	0	0	0	10	2
Ev105	75	0	1	4	2	180	0	3	2	7	1
Ev106	23	0	0	0	0	0	119	6	0	3	3
Ev107	133	18	4	3	0	2	86	125	27	6	21
Ev108	22	0	0	6	0	4	5	6	117	0	3
Ev109	50	16	15	0	0	1	0	1	0	391	4
Ev110	33	3	0	8	0	5	6	5	0	4	137

FIGURE 4. Confusion matrix of I-frames using RCCN.

For RCCN, we also draw the confusion matrix of all samples in the test set including both event and non-event videos in Figure 4, where the digits on the diagonal line show the number of correctly detected event videos. From this figure, it is easily to see that a large proportion of the videos belong to Ev100, i.e., containing no event, where there are some videos similar to the pre-defined events, which would hinder RCCN from learning accurate models for some events. This is actually the data imbalance problem, which remains to be explored in future.

To have an intuitive view on the role of recurrent networks, we extract the feature representations from compressed-domain CNNs (left figure) and LSTMs (right figure) on the validation set, and visualize its low-dimensional representation in t-SNE [45] spaces as depicted in Figure 5. As can be seen in the figures, the left scatter points are

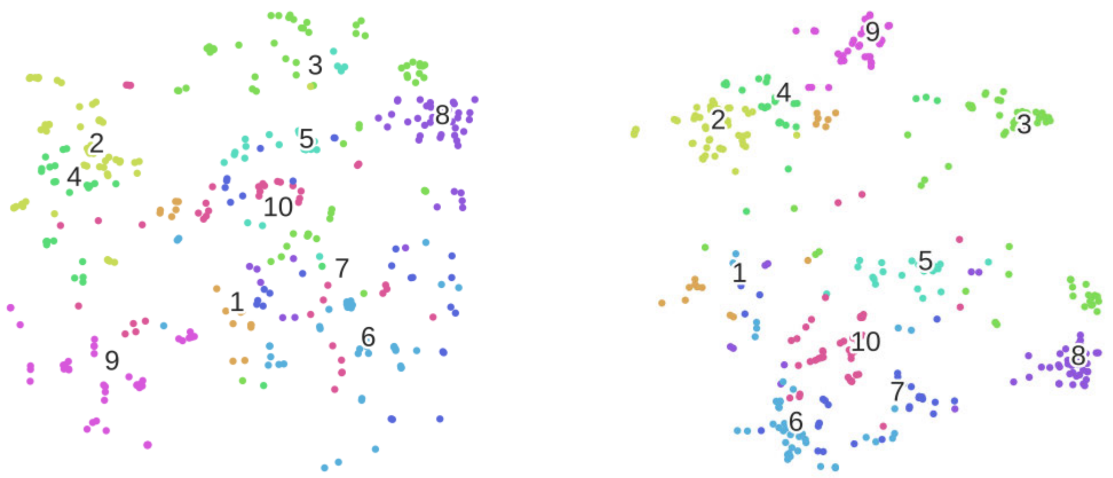


FIGURE 5. Feature visualization for RCCN in t-SNE space. Left: only compressed-domain CNNs; right: after adding LSTMs.

much more separated in comparison with the right ones and meanwhile the right figure has more closely gathered clusters than the left. This validates that the LSTMs module contributes to improving the performance of the RCCN architecture by modeling the temporal structure of the short video sequences.

V. CONCLUSIONS

This work focuses on a new topic, i.e., short video event detection, which still remains untouched but is of vital importance in intelligent video analysis. The biggest challenge of this problem is that the widely existing short videos are usually untrimmed, noisy and sometimes incomplete. To overcome this problem, we proposed a Recurrent Compressed Neural Network (RCCN) architecture to understand the contents of short videos.

The major contributions of this work can be summarized as follows: 1) Short video event detection as an important and emerging topic was explored for the first time in intelligent video sensing; 2) A novel architecture named RCCN consisting of two components, i.e., compressed-domain CNNs and recurrent networks with LSTMs, was developed for detecting events in a vast amount of short videos; 3) RCCN processes video frames in a compressed domain and particularly I-frames as well as P-frames can be directly decoded from the video at a very low computational cost, which makes it be appropriate to deploy in a real-time video analysis system; 4) Variable-range temporal dynamics among decoded video frames can be well modeled by RCCN while common representation can be shared by the short videos of the same event; 5) The representation learned from RCCN model was empowered with discriminative ability for identifying various categories of short videos; 6) One benchmark database was newly made from a public data corpus for the task of short video event detection; 7) Extensive experiments were carried out to investigate the performance of several baseline models and to show the advantages of RCCN compared with several state-of-the-art alternative competitors.

In the future, it is a promising direction of utilizing reinforcement learning or generative adversarial learning to further boost the overall performance of short video event detection models.

REFERENCES

[1] Y. Kong, Z. Tao, and Y. Fu, "Adversarial action prediction networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 539–553, Mar. 2020.

[2] J.-H. Kim and C. S. Won, "Action recognition in videos using pre-trained 2D convolutional neural networks," *IEEE Access*, vol. 8, pp. 60179–60188, 2020.

[3] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, "An attention enhanced graph convolutional LSTM network for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1227–1236.

[4] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[5] C.-Y. Wu, M. Zaheer, H. Hu, R. Manmatha, A. J. Smola, and P. Krahenbuhl, "Compressed video action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6026–6035.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] Y. Cao, F. Yang, Q. Tang, and X. Lu, "An attention enhanced bidirectional LSTM for early forest fire smoke recognition," *IEEE Access*, vol. 7, pp. 154732–154742, 2019.

[8] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, "YFCC100M: The new data in multimedia research," *Commun. ACM*, vol. 59, no. 2, pp. 64–73, Jan. 2016.

[9] J. Bernd, D. Borth, B. Elizalde, G. Friedland, H. Gallagher, L. Gottlieb, A. Janin, S. Karabashlieva, J. Takahashi, and J. Won, "The YLI-MED corpus: Characteristics, procedures, and plans," 2015, *arXiv:1503.04250*. [Online]. Available: http://arxiv.org/abs/1503.04250

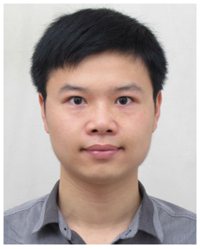
[10] Z. Ma, Y. Yang, Z. Xu, S. Yan, N. Sebe, and A. G. Hauptmann, "Complex event detection via multi-source video attributes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2627–2633.

[11] K.-T. Lai, D. Liu, M.-S. Chen, and S.-F. Chang, "Recognizing complex events in videos by learning key static-dynamic evidences," in *Proc. ECCV*, Sep. 2014, pp. 675–688.

[12] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action recognition in video sequences using deep bi-directional LSTM with CNN features," *IEEE Access*, vol. 6, pp. 1155–1166, 2018.

[13] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1510–1517, Jun. 2018.

[14] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497.
[15] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3D residual networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5534–5542.
[16] Z. Wu, Y.-G. Jiang, X. Wang, H. Ye, and X. Xue, "Multi-stream multi-class fusion of deep networks for video classification," in *Proc. ACM Multimedia Conf. MM*, 2016, pp. 791–800.
[17] B. Zhou, A. Andonian, and A. Torralba, "Temporal relational reasoning in videos," in *Proc. ECCV*, Sep. 2018, pp. 803–818.
[18] S. Bhattacharya, M. M. Kalayeh, R. Sukthankar, and M. Shah, "Recognition of complex events: Exploiting temporal dynamics between underlying concepts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2235–2242.
[19] C. Sun and R. Nevatia, "ACTIVE: Activity concept transitions in video event classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 913–920.
[20] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3551–3558.
[21] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
[22] 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.*, Jun. 2014, pp. 1725–1732.
[23] Z. Xu, Y. Yang, and A. G. Hauptmann, "A discriminative CNN video representation for event detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1798–1807.
[24] X. Chang, Y.-L. Yu, Y. Yang, and E. P. Xing, "Semantic pooling for complex event analysis in untrimmed videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1617–1632, Aug. 2017.
[25] X. Chang, Z. Ma, Y. Yang, Z. Zeng, and A. G. Hauptmann, "Bi-level semantic representation analysis for multimedia event detection," *IEEE Trans. Cybern.*, vol. 47, no. 5, pp. 1180–1197, May 2017.
[26] C. Li, Z. Huang, Y. Yang, J. Cao, X. Sun, and H. T. Shen, "Hierarchical latent concept discovery for video event detection," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2149–2162, May 2017.
[27] Z. Ma, X. Chang, Z. Xu, N. Sebe, and A. G. Hauptmann, "Joint attributes and event analysis for multimedia event detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 2921–2930, Jul. 2018.
[28] X. Chang, Y.-L. Yu, Y. Yang, and A. G. Hauptmann, "Searching persuasively: Joint event detection and evidence recounting with limited supervision," in *Proc. 23rd ACM Int. Conf. Multimedia MM*, 2015, pp. 581–590.
[29] C. Gan, N. Wang, Y. Yang, D.-Y. Yeung, and A. G. Hauptmann, "DevNet: A deep event network for multimedia event detection and evidence recounting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2568–2577.
[30] C. Gan, C. Sun, and R. Nevatia, "Deck: Discovering event composition knowledge from Web images for zero-shot event detection and recounting in videos," in *Proc. AAAI*, Feb. 2017, pp. 4032–4038.
[31] L. Jing, B. Liu, J. Choi, A. Janin, J. Bernd, M. W. Mahoney, and G. Friedland, "A discriminative and compact audio representation for event detection," in *Proc. ACM Multimedia Conf. MM*, 2016, pp. 57–61.
[32] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. NIPS*, 2014, pp. 568–576.
[33] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, "Real-time action recognition with enhanced motion vector CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2718–2726.
[34] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2625–2634.
[35] W. Du, Y. Wang, and Y. Qiao, "Recurrent spatial-temporal attention network for action recognition in videos," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1347–1360, Mar. 2018.
[36] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks for action recognition in videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2740–2755, Nov. 2019.
[37] 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.
[38] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4724–4733.
[39] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet?" in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 18–22.
[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.
[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556
[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[43] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.
[44] L. Su, L. Ma, N. Qin, D. Huang, and A. H. Kemp, "Fault diagnosis of high-speed train bogie by residual-squeeze net," *IEEE Trans. Ind. Informat.*, vol. 15, no. 7, pp. 3856–3863, Jul. 2019.
[45] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.



PING LI (Member, IEEE) received the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China. He was a Postdoctoral Researcher with the National University of Singapore, Singapore, from 2016 to 2017. He is currently an Associate Professor with the School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou. His current research interests include machine learning, computer vision, and intelligent media computing.

XIANGHUA XU received the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China. He is currently a Full Professor with the School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou. His research interests include data mining, the Internet of Things, and artificial intelligence.