# Semantic Feature Mining for Video Event Understanding

XIAOSHAN YANG, TIANZHU ZHANG, and CHANGSHENG XU,
Institute of Automation, Chinese Academy of Sciences

Content-based video understanding is extremely difficult due to the semantic gap between low-level vision signals and the various semantic concepts (object, action, and scene) in videos. Though feature extraction from videos has achieved significant progress, most of the previous methods rely only on low-level features, such as the appearance and motion features. Recently, visual-feature extraction has been improved significantly with machine-learning algorithms, especially deep learning. However, there is still not enough work focusing on extracting semantic features from videos directly. The goal of this article is to adopt unlabeled videos with the help of text descriptions to learn an embedding function, which can be used to extract more effective semantic features from videos when only a few labeled samples are available for video recognition. To achieve this goal, we propose a novel embedding convolutional neural network (ECNN). We evaluate our algorithm by comparing its performance on three challenging benchmarks with several popular state-of-the-art methods. Extensive experimental results show that the proposed ECNN consistently and significantly outperforms the existing methods.

CCS Concepts: • **Computing methodologies** → **Image representations**; *Lexical semantics*; • **Information systems** → *Multimedia streaming;*

Additional Key Words and Phrases: Video recognition, event

**ACM Reference Format:**
Xiaoshan Yang, Tianzhu Zhang, and Changsheng Xu. 2016. Semantic feature mining for video event understanding. ACM Trans. Multimedia Comput. Commun. Appl. 12, 4, Article 55 (August 2016), 22 pages.
DOI: http://dx.doi.org/10.1145/2962719

## 1. INTRODUCTION

With the impressive progress of mobile Internet availability, an increasing number of smartphones and digital cameras have been connected directly or indirectly to the Internet, which successfully facilitates video sharing and propagation. Take the YouTube site as an example: 300 hours of video are uploaded every minute, more than 1 billion users generate billions of views every day, and the number goes up 50 percent year after year. Moreover, this video-sharing site is available in 75 countries with 61 different languages. More than 60 percent of a creator's views come from outside the creator's

This work is supported in part by the National Natural Science Foundation of China (Grant Nos. 61225009, 61303173, 61432019, 61572498, 61532009, 61472379, 61572296), the National Basic Research Program of China (Grant No. 2012CB316304), and the Importation and Development of High-Caliber Talents Project of Beijing Municipal Institutions under Grant IDHT20140224.

Changsheng Xu is the corresponding author.
Authors' addresses: X. Yang, T. Zhang, and C. Xu are with National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, P. R. China; email: xiaoshan.yang@nlpr.ia.ac.cn, tzzhang10@gmail.com, csxu@nlpr.ia.ac.cn.

DOI: http://dx.doi.org/10.1145/2962719

home country[1]. Note that all these numbers are statistics only for YouTube; there are many other well-known sites, such as Facebook, Instagram, and Vine, also including millions of videos, which have recorded everything happening around us and around the world in our daily life. Due to the huge number of videos, it is quite challenging to design an effective algorithm to organize, browse, or retrieve these videos.

To automatically find some interesting videos, most of the present social-network sites provide video retrieval and recommendations through metadata, such as geographical location, timestamp, tag, title, and text description. However, the majority of the existing videos have no metadata, and it is extremely difficult to process them. To overcome this problem, in the research communities of multimedia and computer vision, action recognition and event recognition in videos have been widely studied in recent years for video content understanding. Action recognition mainly focuses on human motions, such as walking, jogging, and hand waving. By contrast, event recognition is more complex because more objects, scenes, and humans may be related to a specific event, such as "Birthday party", "Making a sandwich", or "Rock climbing". A number of algorithms have been proposed to detect and recognize general categories of events in the popular multimedia event detection (MED) dataset from NIST [Ramanathan et al. 2013; Ma et al. 2013; Over et al. 2013]. Even though promising performance for event recognition in videos has been achieved in the past few years, the present technologies cannot meet the requirements of video indexing and retrieval in real applications. There exists significant room for improvement, especially in how to extract semantic features for video content understanding.

In Figure 1, we give an example of the video feature extraction scheme using dense trajectories [Wang et al. 2011; Wang and Schmid 2013], which achieves promising performance and is one of the most representative methods for video recognition recently. This method is proposed for action recognition, in which dense trajectories are obtained by tracking densely sampled points using optical flow fields. The shape of a trajectory mainly encodes local motion patterns. There are three kinds of local features: histograms of oriented gradients (HOGs), which focus on static appearance information; histograms of optical flow (HOFs), which capture local motion information; and motion boundary histograms (MBHs), which encode the relative motion between pixels along the dense trajectories, which are combined to describe the trajectories. Recently, dense trajectories are also adopted for event recognition in videos [Habibian et al. 2014] due to their promising performance. However, there are several problems with using these kinds of features for event recognition. (1) Since motion is the most informative cue for action recognition, the dense trajectories mainly focus on the moving targets in videos. However, contextual information, such as the scene and the background behind the moving objects, is omitted. For example, as shown in Figure 1, by using dense trajectories, the final video features will probably focus on describing the objects covered with green trajectories. The room and the street regions will be omitted, though these areas contain very important contextual cues for event recognition. (2) In traditional methods, video features are extracted based on local visual and motion features. Thus, the global correlation between two humans or objects located at different frames and the global change of the scenes or backgrounds will probably be omitted. For example, as shown in Figure 1, we can see that the man runs out of the room with guns and flees through the street, which shows strong evidence that the video is about a robbery. All these cues can only be parsed from the video by considering the relations between humans in different frames and the change of scenes. (3) Most existing video descriptions are based on hand-designed features. These features capture only low-level information and it has

[1] https://www.youtube.com/yt/press/statistics.html.

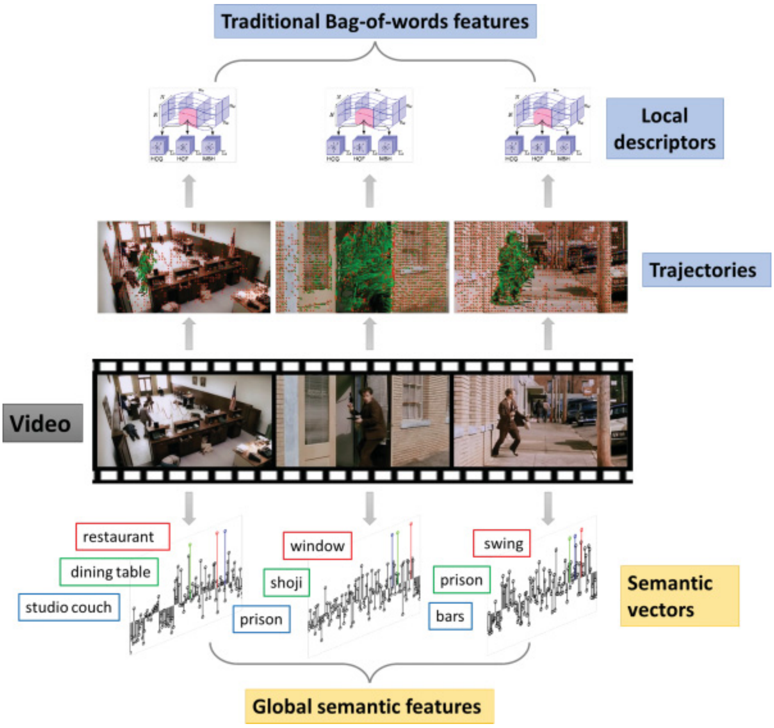

Fig. 1. The traditional features using dense trajectories [Wang et al. 2011; Wang and Schmid 2013] versus the proposed global semantic features for video description.

proven difficult to design features that effectively capture mid-level cues or high-level representations. As a result, the existing features inevitably have a semantic gap with high-level concepts or categories. There are some methods, such as the one in Habibian et al. [2014], which can transform low-level features into more effective semantic representations. However, these projections are carried out based on low-level features, and omitted information while extracting low-level features cannot be recovered.

To deal with these issues, we propose a novel video embedding method based on convolutional neural networks (CNNs) to learn semantic representations for event recognition in videos when only a few labeled samples are available for video recognition. The proposed model has three advantages. (1) To fully consider all the visual concepts contained in videos, we adopt the pretrained CNN model to detect the concepts in each frame. With this scheme, each frame can be represented with a concept vector for which the value of each element corresponds to the appearance probability of the concept. In Figure 1, for each semantic concept vector, we show three concepts with the highest values. From these words, such as “studio couch”, “prison”, and “swing”, and their relations in time, we can see this video is about “robbery”. (2) To explore the global relation between humans or objects located at different frames and the global change of the scenes or backgrounds in video, we convolve the concept vectors of the frames with multiple convolution filters. Here, the filters are optimized with the video–text pairs on the training dataset. (3) To extract the semantic features for video, we adopt the textual information with the constraint that the learned visual descriptions in the semantic space should be similar to the semantic vectors of the related texts. In contrast to the traditional bag-of-words model, which is widely used to represent text, we adopt the word2vector [Mikolov et al. 2013] to obtain the high-quality distributed

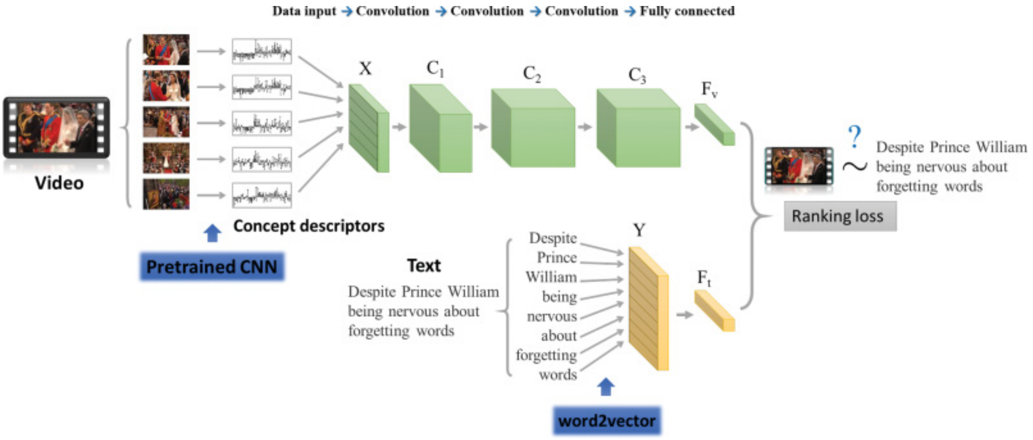
Data input → Convolution → Convolution → Convolution → Fully connected
Video
X
$C_1$
$C_2$
$C_3$
$F_v$
Concept descriptors
Pretrained CNN
?
~
Despite Prince William being nervous about forgetting words
Ranking loss
Text
Despite Prince William being nervous about forgetting words
Despite
Prince
William
being
nervous
about
forgetting
words
Y
$F_t$
word2vector

Fig. 2. Architecture of the proposed convolutional neural networks for video embedding. Several convolutional layers and a fully connected layer are adopted to transform videos into the semantic space, in which the distances between the semantic representation vectors of the video–text pairs are minimized. Refer to the text for details.

vector representations that capture a large number of precise syntactic and semantic word relationships.

The architecture of the proposed embedding convolutional neural networks (ECNN) model for video description is shown in Figure 2. Several convolutional layers and a fully connected layer are adopted to transform videos into the semantic space. In the semantic space, a ranking loss function is adopted to make relevant video and text have consistent semantic features while the non-relevant video and text have different semantic features. Note that, to deal with the convolution operations for videos that always have different amounts of frames, we adopt a dynamical convolution in which the layer size of the network will change according to the input video–text pairs. Compared with the existing methods, the proposed model has four major contributions:

1. We propose an ECNN model trained on top of the pretrained classification neural nets.
2. The proposed ECNNs can capture the temporal semantic changes in video by the convolution operation on the concept vectors of frames.
3. A ranking loss function using only textual supervision is adopted to make visual and textual information have consistent semantic features.
4. By introducing the one-dimensional convolution and dynamical k-max pooling scheme, the proposed ECNN method can process videos with any length without cropping or padding them to a fixed size, as in widely used convolutional neural networks.

The rest of this article is organized as follows. In Section 2, we summarize related work. In Section 3, we introduce local semantic feature extraction. Our method is introduced in Section 4 and the optimization is shown in Section 5. The implementation details are illustrated in Section 6. Experimental results are reported and analyzed in Section 7. Finally, we present our conclusions and discuss future work in Section 8.

## 2. RELATED WORK

In this section, we review the work that is most related to our method, including visual content description, event recognition, and deep learning.

**Visual content description:** For the image description, the early work [Duygulu et al. 2002; Barnard et al. 2003] formulates image recognition as machine translation. There is also some work on image and text embedding [Frome et al. 2013; Kiros et al. 2014; Gong et al. 2014; Karpathy et al. 2014a]. Recently, a number of works describe an image with a single sentence. In Farhadi et al. [2010], images and sentences are projected into an intermediate meaning space for matching. In Chen and Zitnick [2015], a recurrent visual hidden layer is added to reconstruct the visual features from the previous words. A multimodal RNN [Karpathy and Li 2015] is proposed to generate novel descriptions of image regions based on their inferred alignments. In Kiros et al. [2014], an encoder-decoder pipeline is proposed, which unifies joint image–text embedding with multimodal language models. In Kuznetsova et al. [2014], a new description is composed by selectively combining the extracted tree fragments based on the expressive phrases. In Mao et al. [2014] and Vinyals et al. [2015], RNNs are adopted to model the probability distribution of generating a word given an image and previous words. In Lebret et al. [2015], a simple language model based on the syntax of the descriptions is proposed. Socher et al. [2014] mainly focuses on the action and agents in a sentence.

For video description, several previous methods [Guadarrama et al. 2013; Krishnamoorthy et al. 2013; Rohrbach et al. 2013; Thomason et al. 2014] generate a sentence based on mined template knowledge (subject, verb, object). More recently, video description is directly framed as a machine translation problem. In Venugopalan et al. [2015b], knowledge from images with category labels and images with captions is transferred to translate videos to sentences. In Yao et al. [2015], both the local and global temporal structure of videos are explored to produce descriptions. In Venugopalan et al. [2015a], an LSTM-based model to associate a sequence of video frames to a sequence of words is proposed. VideoStory [Habibian et al. 2014] has the most similar idea to the proposed method. In Habibian et al. [2014], the mapping from the video feature to the text feature is learned through two linear transformations. In contrast to this method, the proposed ECNNs connect the frame images of the video with features of the text through convolution neural nets.

There are also some methods for solving both image and the video descriptions [Donahue et al. 2015]. Some approaches address the problem of aligning video and text. In Tapaswi et al. [2015], aligning the chapters of a book to scenes of a video is modeled as finding the shortest path in a sparse directed acyclic graph (DAG). In Bojanowski et al. [2015], given vectorial features for both video and text, the alignment is modeled as a temporal assignment problem, with an implicit linear mapping between the two feature modalities. A novel dataset that contains transcribed audio descriptions temporally aligned to full length HD movies is proposed in Rohrbach et al. [2015].

**Event Recognition:** Recently, many methods have been proposed for vision-based event recognition/detection from videos [Jiang et al. 2012; Yang et al. 2013, 2015b] and event recognition from images [Luo et al. 2008; Imran et al. 2009; Rothe et al. 2015; Wang et al. 2015a, 2015b; Liu et al. 2015; Yang et al. 2015a; Qian et al. 2015, 2016, 2014; Zhang and Xu 2014]. There are two methods related to ours by leveraging on the auxiliary dataset [Duan et al. 2012; Ramanathan et al. 2013]. In Duan et al. [2012], the Domain Selection Machine (DSM) is proposed for event recognition by leveraging on web images. Compared with our algorithm, this method is based only on visual features without considering the contextual information in videos. Moreover, our method uses visual concepts for video descriptions. In Ramanathan et al. [2013], the visual features and textual descriptions are used to represent a video event by learning an "atomic event". In contrast to Ramanathan et al. [2013], our method learns the projection from videos to texts directly to obtain semantic features.

VideoStory embedding [Habibian et al. 2014] is most related to the proposed method. In Habibian et al. [2014], it learns two linear projections between the videos and the texts. Given a video without description, the embedding representation can be obtained by the visual projection. Then, the embedding feature can be further translated into text through textual projection. There are two main differences between the VideoStory embedding and the proposed ECNN. First, VideoStory practically learns two linear projections to translate videos into texts, while the proposed ECNN learns multiple layers of convolutional kernels to find the relations between the videos and their text descriptions. Second, to represent videos, VideoStory still depends on the conventional motion and visual descriptors, while the proposed ECNN adopts global semantic features that are extracted through multiple layers of learned convolution kernels based on local semantic features. Local semantic features are obtained by pretrained concept/semantic classifiers. In this article, the words “semantic” and “concept” are mutually used without distinguishing.

**Deep Learning:** In recent years, deep models including deep belief networks (DBNs) [Hinton et al. 2006], deep Boltzmann machines (DBMs) [Salakhutdinov and Hinton 2009], stacked auto-encoders (SAEs) [Bengio et al. 2006; Vincent et al. 2008] and CNNs [LeCun et al. 1998; Krizhevsky et al. 2012] have drawn much attention due to their encouraging performances compared with the existing shallow models. As an effective feature learning method, the cCNNs has been widely used in computer vision for a lot of applications, such as large-scale object recognition [Krizhevsky et al. 2012; Russakovsky et al. 2015; Zeiler and Fergus 2014; Simonyan and Zisserman 2014; Szegedy et al. 2014], human action recognition [Ji et al. 2013], and face point detection [Sun et al. 2013].

In the literature, two methods [Hu et al. 2014; Wang et al. 2014] are related to the proposed ECNN model. In Hu et al. [2014], deep metric learning is adopted for face verification. This method learns image representations using a deep fully collected neural network such that Euclidean distance can be used to describe sample distances. The inputs to the networks are still low-level visual features, including DSIFT, LBP, and SSIFT. In Wang et al. [2014], the deep ranking model is proposed for fine-grained image similarity learning. This method learns a ranking function by a triplet-based network architecture, and each network is a combination of the CNNs. In contrast to these methods, the proposed model is used mainly to explore the correlations between videos and their text descriptions. For video without text descriptions, we can improve its semantic representation through the learned embedding model.

## 3. LOCAL SEMANTIC FEATURE EXTRACTION

In this section, we first introduce how to sample frames in video. Then, concept description extraction for video based on the sampled frames is illustrated. Next, we will introduce the semantic feature extraction of texts.

### 3.1. Video Sampling

As shown in the left column of Figure 3, there are several frames sampled with a fixed step from a video with 2831 frames. Practically, even with these 6 frames, we can quickly recognize what is happening in this video. At first glance, we can recognize “bicycles”, “man with cycling wear”, “man bending over a bicycle”, “road”, and “grass”. All these pictures lead us to the conclusion that this video is about “a man repairing the disabled bicycle”. Without considering the motion cues, we can classify the video using only several static frames. Based on this observation, we believe that it is an effective way to achieve event recognition in videos by considering the interactions or relations of different objects located at a different timestamp within a video.

On the right side of Figure 3, we show the extracted semantic features of all frames. We can see that a large number of adjacent frames have similar semantic features. Considering that most of the videos are captured by 24 frames per second, most of

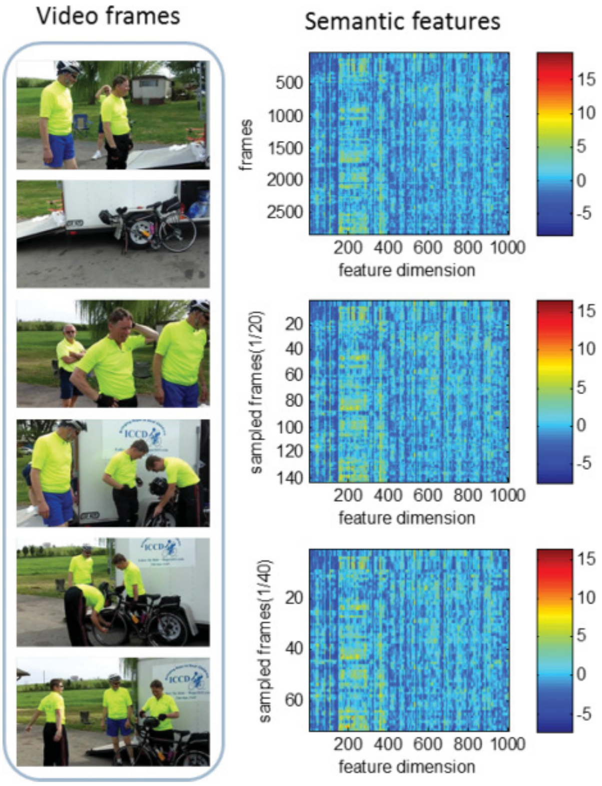

Fig. 3. Left: Six images sampled with a fixed step from a video with 2831 frames. Right: Semantic features of the sampled frames with different steps.

the objects and scenes will not change with such a high speed. Since we mainly focus on extracting global semantic features, the subsampled frames are enough for us to obtain the global semantic pattern. As shown in Figure 3, even with 40 times the original frame rate, the learned semantic features can retrain the global patterns. To reduce the computational cost without decreasing too much performance, we sample frames from the videos with a fixed step (20 in the experiment). This means that we only fetch a single frame per 20 frames in the video.

### 3.2. Video Description

All sampled frames in a given video need to be transformed into concept descriptors that will be imported into the ECNNs. In contrast to the traditional feature extraction methods in which low-level visual features and motion features are combined [Wang et al. 2011], we extract high-level semantic concept features for video frames through concept classifiers. Here, these concept classifiers are pretrained using the image classification model based on CNNs [Krizhevsky et al. 2012]. For convenience, we use the reference CaffeNet model [Jia et al. 2014], which is trained on 1 million photographs with 1000 object categories for ImageNet large-scale visual recognition challenge [Russakovsky et al. 2015].

### 3.3. Text Description

The text descriptions of the videos are always short. For example, in the VideoStory dataset, each text description of the video has only about 7.7 individual terms, on average. It is ineffective to obtain semantic features of these text descriptions using traditional methods, such as the TFIDF or Topic Model. In this work, the semantic features of the text descriptions are extracted according to the pretrained Google News word2vector model. The public word2vector model is pretrained on about 100 billion words in the Google News dataset. The model provides 300-dimensional vectors for 3 million words and phrases. Among the 19159 unique terms in the text descriptions of all videos in the VideoStory dataset, 12261 appear in the Google News words archive. We call these 12261 terms the valid words. Each text description is represented only by the valid words; the other words are omitted. By using this scheme, the maximum length of the text description becomes 40. Then, each text description is further represented as the sequence of vectors that are returned for the corresponding valid words from the Google News archive. If we use $\{\mathbf{y}_1, \ldots, \mathbf{y}_{n_t}\}$ to denote the vectors obtained from the pretrained word2vector model for the $n_t$ words of the text description $\mathbf{t}$, the semantic feature vector $\mathcal{F}_t(\mathbf{t})$ of the text can be obtained by average pooling: $\mathcal{F}_t(\mathbf{t}) = \frac{1}{n_t}\sum_{i=1}^{n_t} \mathbf{y}_i$. The similar element-wise addition scheme of the vectors trained with the Skip-gram model has been effectively adopted to represent the meaning of the combined words or phrase in Mikolov et al. [2013]. Note that each word vector is $\ell_2$ normalized before and after the pooling operation.

## 4. PROPOSED ALGORITHM

In this section, we first introduce the formulation of the proposed ECNNs. Then, we illustrate the convolutional layer, dynamic pooling layer, and the folding and multiple-feature maps in detail. Table I gives the summary of notations.

### 4.1. Our Formulation

Though videos and their corresponding text descriptions are different digital signals that are captured and saved with various formats, they convey the same semantic information when we use them to express an action or event. The proposed video embedding aims to learn a function $\mathcal{F}_v$ that can map videos into the same semantic space as the corresponding text descriptions. In the common semantic space, a given video $\mathbf{v}$ and its text description $\mathbf{t}$ have the similar feature vectors. If there are $N$ video and text description pairs, the goal of video embedding can be formulated as

$$\underset{\mathcal{W}}{\operatorname{argmin}} \sum_{i=1}^{N} l_i d_i + (1 - l_i) max(\tau - d_i, 0) + \lambda \Omega(\mathcal{W}), \tag{1}$$

where $d_i = \|\mathcal{F}_v(\mathbf{v}_i) - \mathcal{F}_t(\mathbf{t}_i)\|_2^2$ is the distance between video $\mathbf{v}_i$ and its text description $\mathbf{t}_i$. $\mathcal{F}_t$ is the semantic representation of the text description that is illustrated in detail in Section 3. The $l_i$ denotes the binary label of the $i^{th}$ video–text pair $(\mathbf{v}_i, \mathbf{t}_i)$. $l_i = 1$ means that text $\mathbf{t}_i$ is related to video $\mathbf{v}_i$, while $l_i = 0$ means that they are randomly sampled nonrelevant pairs. $\mathcal{W}$ denotes the parameter of the map function $\mathcal{F}_v$; $\Omega()$ denotes the $\ell_2$ regularization item of the parameters. $\lambda$ is the weight decay that controls the importance of the regularization term, and $\tau$ is a threshold that enforces the distance between the nonrelevant video—text pair, which should be larger than $\tau$.

In contrast to the traditional video embedding method [Habibian et al. 2014], which relies on a linear mapping function, we adopt CNNs to model the nonlinear functions $\mathcal{F}_v$. As shown in Figure 2, the ECNNs consist of three convolutional layers and a fully connected layer.

If we use $\mathbf{x}_i \in \mathcal{R}^{d_v}$ to denote the concept descriptor for the $i^{th}$ frame of the input video $\mathbf{v}$ ($n_v$ frames), the concept representation matrix $\mathbf{X}$ of the video can be obtained by concatenating the $n_v$ column vectors $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n_v}]$. Here, the concept descriptors

Table I. Summary of Notations

| Notation | Description |
|---|---|
| $N$ | Number of video–text pairs |
| $\mathcal{F}_v$ | Function that can map videos into the semantic space |
| $\mathcal{F}_t$ | Function that can compute the features of the text in the semantic space |
| $\mathbf{L}$ | Total number of neural layers |
| $\mathbf{v}$, $\mathbf{t}$ | Video and the related text description |
| $d_i$,$l_i$ | Distance and label of the $i^{th}$ video–text pair |
| $\mathbf{X}$,$\mathbf{Y}$ | Matrices of local semantic features for video and text |
| $\mathbf{C}$,$\mathbf{F}$ | Outputs of the convolution layer and the fully connected layer |
| $\mathbf{W}_i$,$\mathbf{W}_F$ | Filter in the $i^{th}$ convolution layer and the weight matrix in the fully connected layer |

are obtained using the pretrained CNNs, which are described in detail in Section 3. Given a video–text pair ($\mathbf{v}$, $\mathbf{t}$), the video is first mapped into the semantic space through $\mathbf{v} \rightarrow \mathbf{X} \rightarrow \mathbf{C}_1 \rightarrow \mathbf{C}_2 \rightarrow \mathbf{C}_3 \rightarrow \mathbf{F}_v$ layer by layer. If the text description $\mathbf{t}$ is related to the video $\mathbf{v}$, the output of the last fully connected layer for $\mathbf{v}$ should correspond to the same semantic vector of text $\mathbf{t}$; thus, the distance between them should be minimized. Otherwise, the distance between their semantic vectors should be larger than a given threshold $\tau$. For the proposed ECNNs, the parameters $\mathcal{W} = \{\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, \mathbf{W}_F\}$ are the convolutional filters in the three convolutional layers and the weight matrix in the fully connected layer, respectively. These parameters will be optimized in the training step. More details of these layers are illustrated as follows.

### 4.2. Convolution Layer

As introduced in Section 4.1, the video $\mathbf{v}$ is represented as $\mathbf{X}$, and each column of $\mathbf{X}$ is the feature of a frame. We can see that the feature of each frame is extracted independently by using only the visual appearance of the frame image itself, and it does not consider the correlations among other frames. Therefore, the extracted features are all local descriptions.

To consider the correlations among video frames for feature extraction, we adopt a one-dimensional convolution to capture global semantic features based on local descriptions. The first convolutional layer is defined as

$$\mathbf{C}_1 = conv_1(\mathbf{X}, \mathbf{W}) = \begin{bmatrix} \mathbf{X}(\mathbf{1}, :) \otimes \mathbf{W_1}(\mathbf{1}, :) \\ \vdots \\ \mathbf{X}(d_v, :) \otimes \mathbf{W}_1(d_v, :) \end{bmatrix}, \tag{2}$$

where $\otimes$ denotes the one-dimensional convolution operation. The $\mathbf{X}(i, :)$ and $\mathbf{W}_1(i, :)$ denote the $i^{th}$ rows of the matrices $\mathbf{X}$ and $\mathbf{W}_1$, respectively. Here, $d_v$ different filters are used for the $d_v$-dimensional feature vector independently. These filters are optimized during the training step by using gradient descend and backpropagation.

To consider the correlations among frames in a video, we adopt padded convolution. Compared with unpadded convolution, padded convolution ensures that all weights in the filter reach the entire vector, including the elements at the margins. In addition, padded convolution guarantees that a valid nonempty vector can be obtained no matter that the width of the filter and length of the input vector [Kalchbrenner et al. 2014]. Due to the frame sampling step in the implementation of the proposed ECNN (introduced in Section 6), only dozens of frames may be sampled to capture the semantic concept

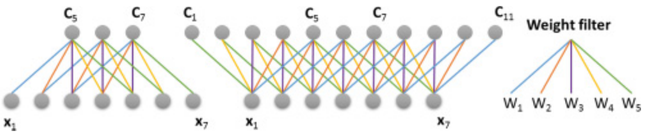

Fig. 4. Unpadded convolution versus padded convolution. The weight filter is shared in these two convolutions.

variation in some videos. As a result, unpadded convolution will cause too few output values in the higher layers and lose most information contained in the original video frames. An example for unpadded and padded convolutions between a 7-dimensional input vector and a filter with size 5 is shown in Figure 4. We can see that unpadded convolution is a subset of the padded convolution operation. It is worth noting that different videos probably have different number of frames. In traditional convolutional neural networks [Krizhevsky et al. 2012], the input size of the convolution layer should be fixed. However, in the proposed model, the column number of the input matrices to the convolution layer can be changed for different videos.

### 4.3. Dynamic k-Max Pooling

Since we denote video and text descriptions as vector sequences, the size of the input in the data layer should be changed for different videos or text descriptions. To make the features of video and text descriptions have the same dimension in the top layers, the size of the data flowing from the data input layer to the top layer should be changed flexibly according to the length of the input vector sequences. To achieve this, the dynamic k-max pooling scheme is adopted after the convolution operation in each layer. As illustrated in Kalchbrenner et al. [2014], the k-max pooling method is a generalized version of the conventional max pooling. Given a vector sequence denoted as matrix $\mathbf{C} \in \mathcal{R}^{d_v \times n_v}$, where each column $\mathbf{c}_i \in \mathcal{R}^{d_v}$ is the feature vector of a video frame, k-max pooling can be written as

$$\text{k-MaxPool}(\mathbf{C}) = \begin{bmatrix} kMax\big(\mathbf{C}(1,:)\big) \\ \vdots \\ kMax\big(\mathbf{C}(d_v,:)\big) \end{bmatrix}, \tag{3}$$

where $kMax\big(\mathbf{C}(i,:)\big)$ denotes the $k$ maximal elements in the $i^{th}$ row of matrix $\mathbf{C}$. The pooled $k$ elements are placed according to their original orders in the row vector $\mathbf{C}(i,:)$.

In each convolution layer, the parameter $k$ of k-max pooling is fixed dynamically according to the length of videos. $k$ is calculated as in Equation (4). Here, $M$ denotes the total number of neural layers, $m$ denotes the index of the current layer, and $s$ denotes the maximum length of the frame sequences for all videos. For the topmost layer, the $k_{top}$ is fixed for all videos such that the sizes of inputs to the loss layer for all videos are consistent.

$$k_m = min(s, (M-(m-1))k_{top}) \tag{4}$$

### 4.4. Folding

In traditional CNNs [Krizhevsky et al. 2012; Karpathy et al. 2014b], 2-dimensional convolution is adopted for input data, such as images and aligned video frames. Here, we mainly focus on the sequential characteristic of the videos and their related text descriptions. In the proposed embedding CNNs, to strengthen the changes among sequential frames and words, we only use 1-dimensional convolution. Thus, the convolution kernel mainly detects feature dependencies across the same rows of the feature maps while the dependence between different rows will be omitted. To alleviate this problem, the folding scheme is adopted. In each convolutional layer, after the convolution operation, every two adjacent rows in a feature map are summed. Thus, the feature maps will have half rows after the folding operations and each row depends on two rows of the feature map in the lower layer.

### 4.5. Multiple Feature Maps

In the convolution layer, a fixed convolution filter focuses on extracting features with a specific pattern. Similar to the 2-dimensional convolution adopted for image classification in Krizhevsky et al. [2012], multiple different convolution filters can be adopted in each 1-dimensional convolution layer to extract multiple feature maps. The convolution and pooling operations for all feature maps can be carried out in parallel.

## 5. OPTIMIZATION

To optimize the objective function (1), we adopt the stochastic gradient descent scheme, which is widely used for training neural networks. In each iteration, the partial derivatives of the weights in each layer are computed with the chain rule. Then, the weights are updated with the adaptive subgradient method [Duchi et al. 2011], in which the gradients computed in previous iterations are also considered.

If we use $\mathcal{L}$ to denote the objective function (1), the partial derivatives with regard to the output feature vector of the last fully connected layer can be computed as follows.

$$\frac{\partial \mathcal{L}}{\partial \mathcal{F}_v(\mathbf{v}_i)} = \begin{cases} \frac{2}{N}\big(\mathcal{F}_v(\mathbf{v}_i) - \mathcal{F}_t(\mathbf{t}_i)\big) & \textbf{if} \quad l_i = 1 \\ \frac{-2}{N}\big(\mathcal{F}_v(\mathbf{v}_i) - \mathcal{F}_t(\mathbf{t}_i)\big) & \textbf{if } l_i = 0 \textbf{ and } d_i < \tau \\ 0 & \textbf{if } l_i = 0 \textbf{ and } d_i \geq \tau \end{cases} \tag{5}$$

Through the chain rule, the partial derivatives with regard to the weight in the last fully connected layer and the input feature can be computed as follows.

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}_F} = \frac{\partial L}{\partial \mathcal{F}_v(\mathbf{v}_i)}\big(1 - \mathcal{F}_v(\mathbf{v}_i)^2\big)\mathbf{C}_3(\mathbf{v}_i) \tag{6}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{C}_3(\mathbf{v}_i)} = \mathbf{W}_F \frac{\partial L}{\partial \mathcal{F}_v(\mathbf{v}_i)}\big(1 - \mathcal{F}_v(\mathbf{v}_i)^2\big) \tag{7}$$

After $\mathbf{W}_F$ is updated with the gradient descent scheme, the derivatives of the lower layers can be computed as in Equation (7) through the backpropagation scheme, as in CNN.

## 6. IMPLEMENTATION DETAILS

Implementations of the proposed video embedding method consist of two main steps. First, local semantic features are extracted for the videos and text descriptions on the VideoStory dataset. Then, these local semantic features are used to train the proposed video embedding model. Finally, the learned model is applied on the training and testing video dataset to extract global semantic features. More details are illustrated, as follows.

### 6.1. Learn the Video Embedding Model

As illustrated in Section 4, based on the extracted local semantic features of both videos and their related texts, we can learn video embedding based on CNNs. The embedding learning aims to optimize all weights in the networks that can map the videos into the global semantic feature space in which the videos and their related text descriptions are constrained to have similar representations. We use 8, 12, and 15 feature maps in the first, second, and third convolution layers, respectively. The kernel size in the three convolution layers is set to 15, 11, and 3, respectively. In the convolution layer and the fully connected layer, we adopt the *tanh* activation function. The dropout scheme is used in the previous layer of the fully connected layer to avoid overfitting. The parameter $k$ of dynamic pooling in the topmost layer is set to 5. The folding to detect the relations among different objects is used after each convolution layer. The learning rate is fixed to 0.001. The gradient descent scheme is adopted to optimize the weights in each layer.

### 6.2. Applying the Learned Embedding Model

In contrast to the embedding learning step, in which the videos have related descriptions, in the training and testing phase, all videos are assumed to have no text descriptions. Once the video embedding model is learned, we can use it to transform all training and testing videos into the global semantic feature space. Specifically, the videos without text descriptions are transmitted through the data layer, convolution layer, and the fully connected layer, which are determined by the learned weights. With these global semantic features, the one-versus-all event classifier is trained for each event. For testing, these binary event classifiers are used to rank the event probabilities in videos.

## 7. EXPERIMENTAL RESULTS

In this section, we first introduce the three video datasets. Then, we illustrate the baseline methods, results, and analyses. For video recognition, we follow the pipeline of the VideoStory method with a few labeled samples (10 samples, as in Habibian et al. [2014]).

### 7.1. Datasets

There are three video datasets used in the experiments: VideoStory, Columbia Consumer Video, and NIST TRECVID HAVIC. The details are as follows.

**VideoStory:** The VideoStory dataset [Habibian et al. 2014] is collected from the video sharing website YouTube. The collection is based on 3000 sentences from the training partition of the NIST TRECVID HAVIC corpus [Strassel et al. 2012]. These 3000 sentences are used to initialize a pool of descriptions. Each text description within the pool is used as a query to search on YouTube; the 25 most relevant videos are collected and their descriptions are added into the description pool. All of these collected videos and descriptions are refined by multiple filters. Finally, the videos with descriptions containing verbs, subjects, and objects simultaneously are accepted. Videos with descriptions that are related to celebrities, TV series, and movie trailers are excluded. In addition, the videos with low visualness descriptions and the ones exceeding 120s are also excluded. The dataset has 45826 videos, with an average length of 58.4s, and contain 743 hours of videos. Each video comes with a short description made of 7.7 individual terms, on average. There are 19159 unique terms in the text descriptions in total.

**Columbia Consumer Video (CCV):** This dataset is released in Jiang et al. [2011], and has $9,317$ consumer videos from YouTube. It consists of over 210h of videos in total (the average length is 80s). All videos are manually annotated with 20 semantic categories at video level. The number of positive examples per category ranges from 224 of “wedding ceremony” to 806 of “music performance”. Among these categories, 15 are event-related: “basketball”, “baseball”, “soccer”, “ice skating”, “skiing”, “swimming”, “biking”, “graduation”, “birthday”, “wedding reception”, “wedding ceremony”, “wedding dance”, “music performance”, “nonmusic performance”, and “parade”. The other 5 categories are objects and scenes: “bird”, “cat”, “dog”, “beach”, and “playground”. In our experiments, we adopt the partitioning scheme as in Habibian et al. [2014], in which the object and scene categories are excluded from the dataset. For simplicity, the 15 categories are denoted as “BK”, “BS”, “SC”, “IS”, “SK”, “SW”, “BK”, “GA”, “BD”, “WR”, “WC”, “WD”, “MP”, “NM”, and “PD”. In the training set, for each event, the 10 training examples are selected based on alphabetical order of the video ID on YouTube. The remaining positive examples in the training set are ignored. The samples in the test set are all used for the performance evaluation as in Jiang et al. [2011].

**NIST TRECVID HAVIC:** This dataset is released in Strassel et al. [2012]. The primary focus of the Heterogeneous Audio Visual Internet Collection (HAVIC) is multidimensional variation inherent in user-generated videos. The dataset contains about 200 videos for Event Kit training, 5K for Background training, 27K for MED test set, 14K for Kindred test set, and 10K videos of a Research collection. All of these videos except the ones from the Research collection have ground-truth annotation at video-level for 20 event categories including “Birthday party”, “Changing vehicle tire”, “Flash mob gathering”, “Getting vehicle unstuck”, “Grooming animal”, “Making sandwich”, “Parade”, “Parkour”, “Repairing appliance”, “Working sewing project”, “Attempting bike trick”, “Cleaning appliance”, “Dog show”, “Giving directions location”, “Marriage proposal”, “Renovating home”, “Rock climbing”, “Town hall meeting”, “Winning race without vehicle”, and “Working metal crafts project”. For simplicity, they are denoted as “BP”, “CV”, “FM”, “GV”, “GA”, “MS”, “PA”, “PK”, “RA”, “WS”, “AB”, “CA”, “DS”, “GD”, “MP”, “RH”, “RC”, “TH”, “WR”, and “WM”, respectively. In the experiment, we follow the 10Ex evaluation procedure outlined in the NIST TRECVID event recognition task [Over et al. 2013]. For each event, the training data is composed of 10 positive videos from the Event Kit training data, and about 5K negative videos are from the Background training data. The event recognition results on both the test set MED and the test set Kindred are reported.

### 7.2. Baseline Methods

We compare the proposed GSF, SFL, and GSFL methods with 5 baseline methods:

**Attributes (A):** In this method, 1346 prespecified attribute classifiers are adopted to represent videos. Every key frame in the video is represented as a Fisher vector coding of densely sampled color SIFT descriptors with spatial pyramids. The individual attribute classifiers are trained by the linear SVM on images from TRECVID and ImageNet. The video representation is obtained by applying the trained classifiers on the video frames sampled by every 2s and then averaging over the entire video.

**Informative Attributes (IA):** This method automatically selects informative attributes for each event and uses the selected subset as video representation. The informative attributes are selected from the same 1346 attribute classifiers as used in the Attributes baseline. A mutual information-based feature selection scheme is adopted on the training data. The reported results are based on the optimal number of selected attributes for each event.

**Low-level (LL):** This method adopts Fisher vector representation using MBH descriptors. The event classifiers are trained directly on the low-level video representations without extracting embedding. In contrast to the previous two baselines, in which only the static local visual features are used, this method also adopts the motion features.

**VideoStory (VS):** This is a linear embedding method, in which videos are represented with dense trajectories with the MBH descriptors and are transformed into a latent feature space. This method can easily translate the videos into texts by the two linear projections. In contrast to the other baselines, this method adopts the VideoStory dataset [Habibian et al. 2014] with text annotation to learn an effective embedding for videos.

Table II. Average Precision (AP) for Event Recognition on the Columbia Consumer Videos Dataset

| Events | A | IA | LL | VS | SF | **GSF** | **SFL** | **GSFL** |
|---|---|---|---|---|---|---|---|---|
| BK | 0.293 | 0.317 | 0.485 | 0.553 | 0.791 | **0.831** | 0.783 | 0.782 |
| BS | 0.401 | 0.463 | 0.298 | 0.299 | 0.591 | **0.721** | 0.597 | 0.707 |
| SC | 0.336 | 0.302 | 0.469 | 0.505 | 0.721 | **0.745** | 0.742 | 0.725 |
| IS | 0.632 | 0.649 | 0.646 | 0.675 | 0.758 | 0.784 | **0.815** | 0.798 |
| SK | 0.641 | 0.651 | 0.610 | 0.671 | 0.825 | 0.813 | 0.813 | **0.835** |
| SW | 0.520 | 0.489 | 0.691 | 0.764 | **0.830** | 0.782 | 0.804 | 0.791 |
| BK | 0.324 | 0.307 | 0.420 | 0.561 | 0.558 | 0.631 | 0.596 | **0.673** |
| GA | 0.083 | 0.058 | 0.135 | 0.121 | 0.580 | 0.625 | 0.595 | **0.665** |
| BD | 0.149 | 0.216 | 0.187 | 0.257 | 0.496 | 0.507 | 0.474 | **0.563** |
| WR | 0.147 | 0.201 | 0.124 | 0.117 | 0.510 | 0.615 | 0.526 | **0.711** |
| WC | 0.216 | 0.248 | 0.387 | 0.324 | 0.533 | 0.588 | 0.559 | **0.664** |
| WD | 0.243 | 0.294 | 0.550 | 0.521 | 0.737 | **0.748** | 0.708 | 0.729 |
| MP | 0.279 | 0.247 | 0.225 | 0.201 | 0.368 | **0.464** | 0.404 | 0.395 |
| NM | 0.195 | 0.190 | 0.334 | 0.282 | 0.614 | 0.705 | 0.603 | **0.761** |
| PD | 0.247 | 0.295 | 0.579 | 0.634 | 0.737 | **0.760** | 0.752 | 0.747 |
| Mean | 0.314 | 0.328 | 0.409 | 0.432 | 0.643 | 0.687 | 0.651 | **0.703** |

*Note*: The results obtained by the proposed methods are highlighted with boldface.

**Semantic Feature (SF):** In this method, the concept descriptors of the sampled frames in the video are extracted using the reference CaffeNet model [Jia et al. 2014], which is trained on 1 million photographs with 1000 object categories for ImageNet large-scale visual recognition challenge [Russakovsky et al. 2015]. The semantic feature vector for each video is obtained by pooling the concept vectors of all frames.

**Global Semantic Feature (GSF):** This method uses the global semantic features extracted by the proposed embedding CNNs trained on the VideoStory dataset [Habibian et al. 2014]. Then, the event classifiers are trained and tested using the global semantic features on the CCV, MED, and Kindred datasets. It is worth noting that the proposed embedding CNNs are trained on videos with text descriptions and applied for event recognition on videos without text descriptions.

**Semantic Feature and Low-Level Feature (SFL):** In this method, the semantic features and the low-level features are combined. Since the semantic features capture the objects and their backgrounds in each frame simultaneously, they can complement the low-level features (MBH), which mainly focus on moving targets. This method can measure the complementarity of these two kinds of features.

**Global Semantic Feature and Low-Level Feature (GSFL):** In this method, the global semantic features and the low-level features are combined. The low-level features (MBH) are encoded with the bag-of-words scheme to consider temporal and spacial information, while the proposed global semantic features consider all the appearance changes of scenes and objects in the videos through convolution.

### 7.3. Results and Analysis

As in Habibian et al. [2014], the performance of event recognition is measured by average precision (AP, area under the uninterpreted PR curve); the mean AP (mAP) of all event categories is also reported.

**Results on the CCV dataset:** The binary event classifier trained for each event can give the probability of whether an event appears in the test video. With different thresholds, we can obtain the prediction results with different precisions and recalls. With the same measurement used in Habibian et al. [2014], we compute the AP for each event. The results on the CCV dataset are shown in Table II. We can see that the global semantic features learned by the proposed ECNN method perform better than the existing methods.

Table III. Average Precision (AP) for Event Recognition on the MED Dataset

| Events | A | IA | LL | VS | SF | **GSF** | **SFL** | **GSFL** |
|---|---|---|---|---|---|---|---|---|
| BP | 0.089 | 0.103 | 0.083 | 0.118 | **0.171** | 0.076 | 0.151 | 0.142 |
| CV | 0.217 | 0.239 | 0.106 | 0.103 | 0.225 | 0.028 | 0.269 | **0.331** |
| FM | 0.432 | 0.434 | 0.544 | 0.535 | 0.318 | 0.446 | 0.306 | **0.554** |
| GV | 0.307 | 0.309 | 0.137 | 0.319 | 0.226 | 0.303 | 0.214 | **0.340** |
| GA | 0.102 | 0.110 | 0.114 | 0.151 | 0.152 | 0.173 | 0.150 | **0.186** |
| MS | 0.055 | 0.054 | 0.073 | 0.074 | 0.111 | 0.103 | **0.117** | 0.116 |
| PD | 0.195 | 0.198 | 0.352 | **0.452** | 0.156 | 0.223 | 0.199 | 0.342 |
| PK | 0.170 | 0.184 | 0.705 | **0.721** | 0.429 | 0.311 | 0.383 | 0.660 |
| RA | 0.143 | 0.163 | 0.174 | 0.184 | 0.153 | 0.314 | 0.188 | **0.381** |
| WS | 0.081 | 0.106 | 0.085 | **0.151** | 0.064 | 0.120 | 0.074 | 0.150 |
| AB | 0.135 | 0.144 | 0.033 | 0.061 | 0.058 | 0.171 | 0.078 | **0.187** |
| CA | 0.007 | 0.033 | 0.072 | **0.078** | 0.027 | 0.032 | 0.017 | 0.039 |
| DS | 0.164 | 0.187 | 0.409 | 0.354 | 0.189 | 0.250 | 0.224 | **0.566** |
| GD | 0.007 | 0.018 | 0.047 | 0.004 | 0.020 | 0.070 | **0.425** | 0.021 |
| MP | 0.002 | **0.018** | 0.007 | 0.004 | 0.003 | 0.004 | 0.001 | 0.004 |
| RH | 0.047 | 0.047 | **0.072** | 0.051 | 0.007 | 0.006 | 0.009 | 0.014 |
| RC | 0.090 | 0.101 | 0.118 | 0.100 | 0.015 | 0.023 | 0.029 | **0.155** |
| TH | 0.157 | **0.176** | 0.149 | 0.118 | 0.025 | 0.006 | 0.014 | 0.024 |
| WR | 0.206 | 0.210 | 0.130 | **0.217** | 0.102 | 0.150 | 0.182 | 0.171 |
| WM | 0.090 | 0.101 | 0.068 | 0.118 | 0.126 | 0.098 | 0.117 | **0.189** |
| Mean | 0.135 | 0.147 | 0.174 | 0.196 | 0.129 | 0.158 | 0.138 | **0.229** |

*Note*: The results obtained by the proposed methods are highlighted with boldface.

**Results on the MED dataset:** With the same training and test splits as used in Habibian et al. [2014], in which 10 positive examples and other negative samples are used to train binary classifiers for each event, we obtain the results as shown in Table III. We can see that, by combining the global semantic features and the low-level features, the performance is improved about 3% compared with the VideoStory method. This improvement demonstrates that the global semantic features can complement well the traditional video features with the local visual and motion information. In contrast to the results on the CCV dataset, in which the semantic features perform much better than the VideoStory method, global features alone cannot achieve better results than low-level features. This is probably due to the data discrepancy between the videos used for training the embedding model and the videos in the MED dataset. Semantic features can also be improved by combining low-level features. However, global features with the low-level feature still perform better.

**Results on the Kindred dataset:** This dataset is used only for testing, and the binary event classifiers trained on the MED dataset are adopted for predicting the appearance probability of each event. The results are shown in Table IV and have a similar conclusion drawn on the MED dataset. The combination of global semantic features and low-level features can improve performance significantly.

**Discussion:** Based on the results on the 3 video datasets, we have the following observations: (1) The proposed global semantic features perform much better than the traditional semantic features, such as the baseline methods A and IA. This demonstrates the effectiveness of the convolution operation for considering the global temporal semantic patterns in the videos. (2) Compared with low-level features, the proposed global semantic features show more than 30% improvement on the CCV dataset. This huge improvement results mainly from two aspects. The first is that global semantic features can capture the global change of a specific object or scene, while low-level features mainly consider local moving targets. The second is that the learned video embedding model practically introduces the extra semantic patterns from the VideoStory dataset. (3) On the MED and Kindred video datasets, global semantic features perform a little worse than low-level features. However, the combination of global features and

Table IV. Average Precision (AP) for Event Recognition on the Kindred Dataset

| Events | A | IA | LL | VS | SF | **GSF** | **SFL** | **GSFL** |
|---|---|---|---|---|---|---|---|---|
| BP | 0.365 | **0.379** | 0.224 | 0.331 | 0.207 | 0.353 | 0.222 | 0.372 |
| CV | 0.087 | 0.109 | 0.167 | 0.180 | 0.172 | 0.203 | 0.162 | **0.221** |
| FM | 0.078 | 0.080 | 0.248 | 0.309 | 0.209 | 0.329 | 0.306 | **0.329** |
| GV | 0.354 | 0.371 | 0.301 | 0.393 | 0.401 | 0.405 | 0.387 | **0.436** |
| GA | 0.328 | 0.336 | 0.381 | 0.501 | 0.189 | 0.325 | 0.501 | **0.536** |
| MS | 0.297 | 0.296 | **0.356** | 0.278 | 0.247 | 0.291 | 0.261 | 0.310 |
| PA | 0.056 | 0.059 | 0.106 | 0.146 | 0.131 | 0.157 | 0.129 | **0.174** |
| PK | 0.023 | 0.037 | 0.619 | 0.792 | 0.143 | 0.603 | 0.785 | **0.814** |
| RA | 0.111 | 0.131 | 0.540 | 0.534 | 0.356 | 0.361 | 0.329 | **0.565** |
| WS | 0.022 | 0.047 | 0.327 | 0.488 | 0.104 | 0.316 | 0.484 | **0.516** |
| AB | 0.042 | 0.041 | 0.099 | 0.198 | 0.202 | 0.204 | 0.188 | **0.247** |
| CA | 0.008 | 0.034 | 0.110 | 0.162 | 0.127 | 0.181 | 0.150 | **0.194** |
| DS | 0.133 | 0.156 | **0.479** | 0.416 | 0.194 | 0.238 | 0.411 | 0.445 |
| GD | 0.003 | 0.014 | 0.004 | 0.003 | 0.004 | 0.018 | 0.009 | **0.033** |
| MP | 0.003 | 0.019 | 0.008 | 0.008 | 0.015 | **0.034** | 0.005 | 0.033 |
| RH | 0.141 | 0.142 | 0.112 | 0.131 | 0.128 | 0.153 | 0.122 | **0.178** |
| RC | 0.244 | 0.255 | 0.557 | 0.618 | 0.239 | 0.437 | 0.403 | **0.653** |
| TH | 0.097 | **0.116** | 0.065 | 0.061 | 0.067 | 0.077 | 0.055 | 0.092 |
| WR | 0.243 | 0.255 | 0.308 | 0.413 | 0.263 | 0.238 | 0.399 | **0.452** |
| WM | 0.083 | 0.104 | 0.241 | 0.278 | 0.153 | 0.205 | 0.270 | **0.307** |
| Mean | 0.136 | 0.149 | 0.263 | 0.312 | 0.177 | 0.257 | 0.278 | **0.345** |

*Note*: The results obtained by the proposed methods are highlighted with boldface.

low-level features achieves the best results. This demonstrates that these two different features can complement and enhance each other. (4) On the three video datasets, the proposed global semantic features perform better than the linear embedding method VideoStory. This is because the multiple convolution layers and fully connected layer of the proposed ECNN method can construct a more effective nonlinear projection function to embed the videos into the shared semantic space.

Here, we present more analysis about the semantic features. In the proposed ECNN method, the semantic features obtained by pretrained CNN models are adopted as the input data to the convolution layers. Practically, the 1000 dimensional features in the last fully connected layer (FC8 layer of the CaffeNet [Jia et al. 2014]) are used as the semantic features. In pretraining, the classification loss is computed based on the difference between these semantic features and the ground-truth labels in the training set. Thus, each dimension of the FC8 layer feature denotes the appearance probability of the related object. We believe that these semantic features are more effective than the lower-level features to model the relation between objects in different locations of the video. To support this point, in Figure 5, we provide a comparison between the middle-level features and -semantic features. We adopt the 4096 dimensional features in the FC7 layer as the middle-level features to learn the embedding model. Then, the learned model is adopted to obtain the embedded features (global middle features). We can see that the proposed ECNN method can learn more effective semantic features for most events.

In Figure 6, we show the convergence analysis of the proposed ECNN. The cost of the validation set descends quickly in the first iterations, and our method converges after about 200 iterations. In the experiment, the competitive results are obtained within 3 epochs.

### 7.4. Parameter Analysis

Several parameters play important roles in the ECNN. In this section, we show their effects on performance.

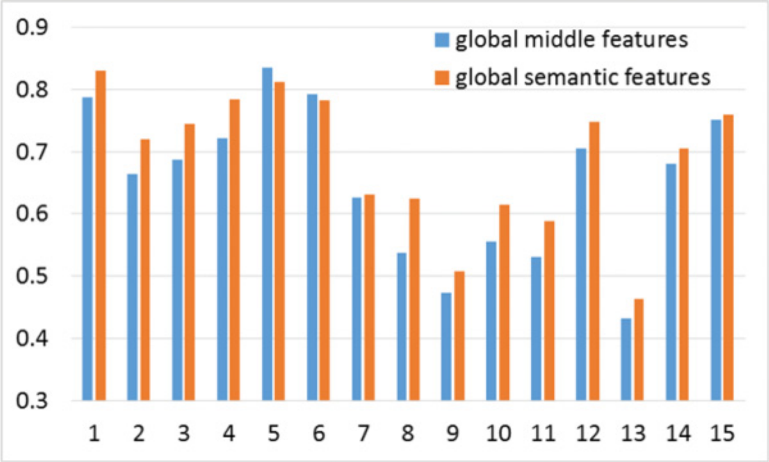

Fig. 5. Precisions of two different features on the CCV dataset.

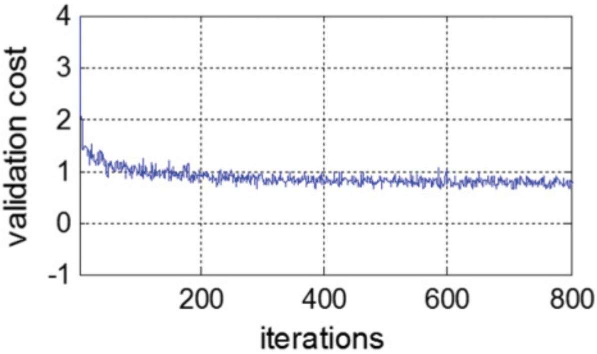

Fig. 6. Convergence analysis of the ECNN.

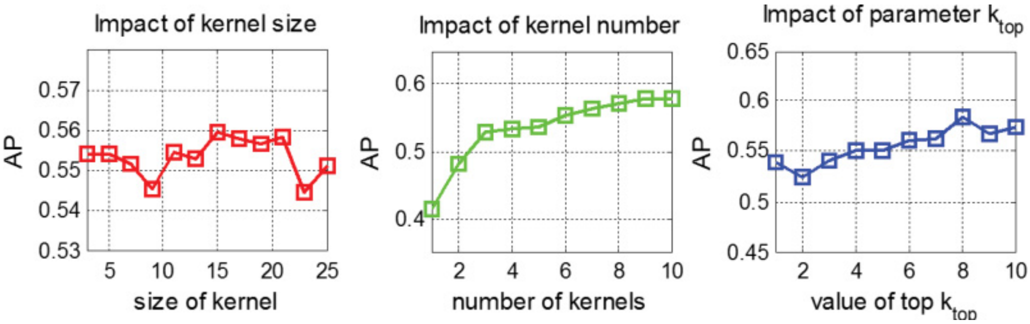

Fig. 7. Effect of three parameters: kernel size, number of kernels, and the pooling size of the top convolution layer (the first 100 dimensions of the semantic features are adopted).

**Effect of kernel size:** Kernel size is one of the most important parameters in the convolution layers. A kernel that is too large will ignore the details while a small kernel will fail to detect the effective features. In the first column of Figure 7, we show the average precision of event recognition using different kernel sizes in the first

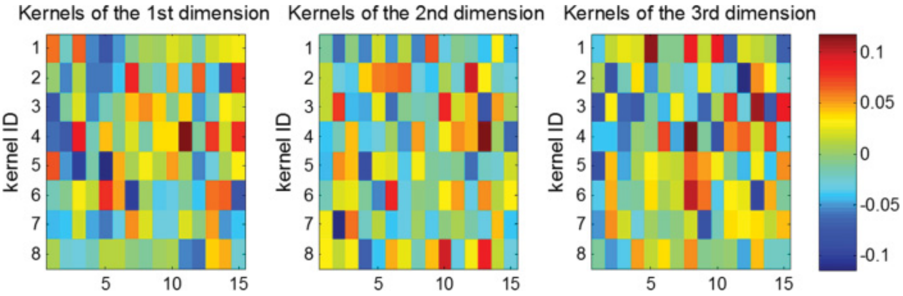

Fig. 8. Kernels learned in the proposed ECNN method for the first three dimensions of the semantic features (8 kernels with size 15 are adopted).

convolution layer. We can see that the best performance is obtained when the kernel size is 15.

**Effect of $k_{top}$:** In most of the traditional convolutional neural networks, to obtain features with the same dimension in the fully connected layer, the pooling size and stride are fixed for all convolution layers. In the proposed ECNN, as illustrated in Section 4.3, through the dynamic k-max pooling, different videos with various lengths can be processed with the learned uniform convolution kernels. Practically, as shown in Equation (4), the size of the pooled features is decided by the dimensions of the inputs. The longer input sequences have features with a higher dimension after pooling. Only the pooling size of the last convolution layer is fixed for ease of the fully connected layer. To explore the effect of the size of the top pooling layer $k_{top}$, in the third column of Figure 7, we show the average precisions of event recognition with $k_{top}$ changing from 1 to 10. We can see that better performance is obtained using $k_{top}$ with larger values. However, performance will decrease once the $k_{top}$ is bigger than 8.

**Effect of the number of maps:** Here, we explore the effect of the multiple feature maps in the proposed ECNN. As illustrated in Section 4.5, multiple feature maps are adopted in each convolution layer to capture the global semantic pattern. Practically, each feature map corresponds to a single convolution kernel. This is similar to the CNNs in which multiple kernels are learned in the first convolution layer to capture the low-level visual pattern. In Figure 8, we show the 8 kernels with size 15 that are learned by the first convolution layer of the proposed ECNN method. As illustrated in Section 4, each dimension of the semantic features has multiple kernels. Here, we show the 8 kernels with only the first three dimensions. We can see that the kernels learned for different dimensions have a large discrepancy. Generally, the more kernels used for each dimension of the semantic features, the more semantic patterns can be captured. As shown in the second column of Figure 7, performance increases constantly when more feature maps are used. When the number of feature maps is greater than 10, performance no longer increases.

## 8. CONCLUSIONS

In this article, we proposed an embedding convolutional neural network (ECNN) for event recognition in videos. The whole network aims to map the videos into the semantic feature space, where the videos and its related text descriptions have similar representations. The networks are comprised of a data-input layer, three one-dimensional convolution layers, and a fully connected layer. In the data-input layer, the videos are represented with sequences of feature vectors; then, these vectors are convolved with three one-dimensional filters and fully connected to the final layer to compute the loss. The extensive results demonstrate the effectiveness of the proposed ECNN model. In the future, we will improve this model by introducing other local cues, such as motion features. In addition, a completely bottom-up convolution network will be adopted for semantic feature extraction in videos.

## REFERENCES

Kobus Barnard, Pinar Duygulu, David A. Forsyth, Nando de Freitas, David M. Blei, and Michael I. Jordan. 2003. Matching words and pictures. *Journal of Machine Learning Research* 3, 1107–1135.

Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. 2006. Greedy layer-wise training of deep networks. In *NIPS*. 153–160.

Piotr Bojanowski, Rémi Lajugie, Edouard Grave, Francis R. Bach, Ivan Laptev, Jean Ponce, and Cordelia Schmid. 2015. Weakly-supervised alignment of video with text. In *2015 IEEE International Conference on Computer Vision (ICCV'15),* Santiago, Chile, December 7–13, 2015, 4462–4470.

Xinlei Chen and C. Lawrence Zitnick. 2015. Mind's eye: A recurrent visual representation for image caption generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015)*, Boston, MA, June 7–12, 2015, 2422–2431.

Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Trevor Darrell, and Kate Saenko. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015)*, Boston, MA, June 7–12, 2015, 2625–2634.

Lixin Duan, Dong Xu, and Shih-Fu Chang. 2012. Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. In *IEEE CVPR*. 1338–1345.

John C. Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12, 2121–2159.

Pinar Duygulu, Kobus Barnard, João F. G. de Freitas, and David A. Forsyth. 2002. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceedings of Computer Vision - 7th European Conference on Computer Vision (ECCV'02), Part IV*. Copenhagen, Denmark, May 28–31, 2002, 97–112.

Ali Farhadi, Seyyed Mohammad Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David A. Forsyth. 2010. Every picture tells a story: Generating sentences from images. In *Proceedings of Computer Vision - 11th European Conference on Computer Vision (ECCV'10), Part IV*. Heraklion, Crete, Greece, September 5–11, 2010, 15–29.

Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. 2013. DeViSE: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013*. Proceedings of a meeting held December 5–8, 2013, Lake Tahoe, NV, 2121–2129.

Yunchao Gong, Qifa Ke, Michael Isard, and Svetlana Lazebnik. 2014. A multi-view embedding space for modeling Internet images, tags, and their semantics. *International Journal of Computer Vision* 106, 2, 210–233.

Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond J. Mooney, Trevor Darrell, and Kate Saenko. 2013. YouTube2Text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *IEEE International Conference on Computer Vision (ICCV'13),* Sydney, Australia, December 1–8, 2013, 2712–2719.

AmirHossein Habibian, Thomas Mensink, and Cees G. M. Snoek. 2014. VideoStory: A new multimedia embedding for few-example recognition and translation of events. In *Proceedings of the ACM MM*. 17–26.

Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural Computation* 18, 7, 1527–1554.

Junlin Hu, Jiwen Lu, and Yap-Peng Tan. 2014. Discriminative deep metric learning for face verification in the wild. In *CVPR*. 1875–1882.

Naveed Imran, Jingen Liu, Jiebo Luo, and Mubarak Shah. 2009. Event recognition from photo collections via PageRank. In *ACM Multimedia*. 621–624.

Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 2013. 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 1, 221–231.

Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*.

Lu Jiang, Alexander G. Hauptmann, and Guang Xiang. 2012. Leveraging high-level and low-level features for multimedia event detection. In *ACM MM*. 449–458.

Yu-Gang Jiang, Guangnan Ye, Shih-Fu Chang, Daniel P. W. Ellis, and Alexander C. Loui. 2011. Consumer video understanding: A benchmark database and an evaluation of human and machine performance. In *ICMR*. 29.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *ACLs*. 655–665.

Andrej Karpathy, Armand Joulin, and Fei-Fei Li. 2014a. Deep fragment embeddings for bidirectional image sentence mapping. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems,* December 8–13, 2014, Montreal, Quebec, Canada, 1889–1897.

Andrej Karpathy and Fei-Fei Li. 2015. Deep visual-semantic alignments for generating image descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15)*, Boston, MA, June 7–12, 2015. 3128–3137.

Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. 2014b. Large-scale video classification with convolutional neural networks. In *CVPR*. 1725–1732.

Ryan Kiros, Ruslan Salakhutdinov, and Richard S. Zemel. 2014. Unifying visual-semantic embeddings with multimodal neural language models. *CoRR* abs/1411.2539.

Niveda Krishnamoorthy, Girish Malkarnenkar, Raymond J. Mooney, Kate Saenko, and Sergio Guadarrama. 2013. Generating natural-language video descriptions using text-mined knowledge. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence*, July 14–18, 2013, Bellevue, WA.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *NIPS*. 1106–1114.

Polina Kuznetsova, Vicente Ordonez, Tamara L. Berg, and Yejin Choi. 2014. TREETALK: Composition and compression of trees for image descriptions. *TACL* 2, 351–362.

Rémi Lebret, Pedro O. Pinheiro, and Ronan Collobert. 2015. Phrase-based image captioning. In *Proceedings of the 32nd International Conference on Machine Learning (ICML'15)*, Lille, France, July 6–11, 2015. 2085–2094.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of IEEE* 86, 11, 2278–2324.

Mengyi Liu, Xin Liu, Yan Li, Xilin Chen, Alexander G. Hauptmann, and Shiguang Shan. 2015. Exploiting feature hierarchies with convolutional neural networks for cultural event recognition. In *2015 IEEE International Conference on Computer Vision Workshop (ICCV Workshops'15)*, Santiago, Chile, December 7–13, 2015. 274–279.

Jiebo Luo, Jie Yu, Dhiraj Joshi, and Wei Hao. 2008. Event recognition: Viewing the world with a third eye. In *Proceedings of the 16th ACM MM*. 1071–1080.

Zhigang Ma, Yi Yang, Zhongwen Xu, Nicu Sebe, and Alexander G. Hauptmann. 2013. We are not equally negative: Fine-grained labeling for multimedia event detection. In *ACM MM*. 293–302.

Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L. Yuille. 2014. Deep captioning with multimodal recurrent neural networks (m-RNN). *CoRR* abs/1412.6632.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*. 3111–3119.

Paul Over, Jon Fiscus, Greg Sanders, David Joy, Martial Michel, George Awad, Alan F. Smeaton, Wessel Kraaij, and Georges Quenot. 2013. TRECVID 2013 – An overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2013*. NIST.

Shengsheng Qian, Tianzhu Zhang, Richang Hong, and Changsheng Xu. 2015. Cross-domain collaborative learning in social multimedia. In *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference (MM'15)*, Brisbane, Australia, October 26–30, 2015, 99–108.

Shengsheng Qian, Tianzhu Zhang, Changsheng Xu, and M. Shamim Hossain. 2014. Social event classification via boosted multimodal supervised latent Dirichlet allocation. *ACM Transactions on Multimedia Computing* 11, 2, 27:1–27:22.

Shengsheng Qian, Tianzhu Zhang, Changsheng Xu, and Jie Shao. 2016. Multi-modal event topic model for social event analysis. *IEEE Transactions on Multimedia* 18, 2, 233–246.

Vignesh Ramanathan, Percy Liang, and Fei-Fei Li. 2013. Video event understanding using natural language descriptions. In *Proceedings of the IEEE ICCV*. 905–912.

Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. 2015. A dataset for movie description. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15),* Boston, MA, June 7–12, 2015. 3202–3212.

Marcus Rohrbach, Wei Qiu, Ivan Titov, Stefan Thater, Manfred Pinkal, and Bernt Schiele. 2013. Translating video content to natural language descriptions. In *IEEE International Conference on Computer Vision (ICCV'13)*, Sydney, Australia, December 1–8, 2013, 433–440.

Rasmus Rothe, Radu Timofte, and Luc J. Van Gool. 2015. DLDR: Deep linear discriminative retrieval for cultural event classification from a single image. In *2015 IEEE International Conference on Computer Vision Workshop (ICCV Workshops'15)*, Santiago, Chile, December 7–13, 2015, 295–302.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision* 115, 3, 211–252.

Ruslan Salakhutdinov and Geoffrey E. Hinton. 2009. Deep Boltzmann machines. In *Proceedings of the International Conference on AISTATS*. 448–455.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR* abs/1409.1556.

Richard Socher, Andrej Karpathy, Quoc V. Le, Christopher D. Manning, and Andrew Y. Ng. 2014. Grounded compositional semantics for finding and describing images with sentences. *TACL* 2, 207–218.

Stephanie Strassel, Amanda Morris, Jonathan G. Fiscus, Christopher Caruso, Haejoong Lee, Paul Over, James Fiumara, Barbara Shaw, Brian Antonishek, and Martial Michel. 2012. Creating HAVIC: Heterogeneous audio visual Internet collection. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*. 2573–2577.

Yi Sun, Xiaogang Wang, and Xiaoou Tang. 2013. Deep convolutional network cascade for facial point detection. In *CVPR*. 3476–3483.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2014. Going deeper with convolutions. *CoRR* abs/1409.4842.

Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. 2015. Book2Movie: Aligning video scenes with book chapters. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15)*, Boston, MA, USA, June 7–12, 2015, 1827–1835.

Jesse Thomason, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Raymond J. Mooney. 2014. Integrating language and vision to generate natural language descriptions of videos in the wild. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING'14), Technical Papers*, August 23–29, 2014, Dublin, Ireland, 1218–1227.

Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond J. Mooney, Trevor Darrell, and Kate Saenko. 2015a. Sequence to sequence - Video to text. In *2015 IEEE International Conference on Computer Vision (ICCV'15)*, Santiago, Chile, December 7–13, 2015, 4534–4542.

Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond J. Mooney, and Kate Saenko. 2015b. Translating videos to natural language using deep recurrent neural networks. In *2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT'15)*, Denver, CO, May 31 - June 5, 2015, 1494–1504.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the ICML*. 1096–1103.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15)*, Boston, MA, June 7–12, 2015, 3156–3164.

Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. 2011. Action recognition by dense trajectories. In *CVPR*. 3169–3176.

Heng Wang and Cordelia Schmid. 2013. Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*.

Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. 2014. Learning fine-grained image similarity with deep ranking. In *CVPR*. 1386–1393.

Limin Wang, Zhe Wang, Wenbin Du, and Yu Qiao. 2015a. Object-scene convolutional neural networks for event recognition in images. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, Boston, MA, June 7–12, 2015, 30–35.

Limin Wang, Zhe Wang, Sheng Guo, and Yu Qiao. 2015b. Better exploiting OS-CNNs for better event recognition in images. In *2015 IEEE International Conference on Computer Vision Workshop (ICCV Workshops'15)*, Santiago, Chile, December 7–13, 2015, 287–294.

Xiaoshan Yang, Tianzhu Zhang, and Changsheng Xu. 2015a. Cross-domain feature learning in multimedia. *IEEE Transactions on Multimedia* 17, 1, 64–78.

Xiaoshan Yang, Tianzhu Zhang, Changsheng Xu, and M. Shamim Hossain. 2015b. Automatic visual concept learning for social event understanding. *IEEE Transactions on Multimedia* 17, 3, 346–358.

Yi Yang, Zhigang Ma, Zhongwen Xu, Shuicheng Yan, and Alexander G. Hauptmann. 2013. How related exemplars help complex event detection in web videos. In *ICCV*. 2104–2111.

Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher J. Pal, Hugo Larochelle, and Aaron C. Courville. 2015. Describing videos by exploiting temporal structure. In *2015 IEEE International Conference on Computer Vision (ICCV'15)*, Santiago, Chile, December 7–13, 2015, 4507–4515.

Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *13th ECCV*. 818–833.

Tianzhu Zhang and Changsheng Xu. 2014. Cross-domain multi-event tracking via CO-PMHT. *ACM Transactions on Multimedia Computing* 10, 4, 31:1–31:19.

Received November 2015; revised May 2016; accepted May 2016