Learning Deep Intrinsic Video Representation by Exploring Temporal Coherence and Graph Structure[*]

Yingwei Pan [†], Yehao Li [‡], Ting Yao [§], Tao Mei [§], Houqiang Li [†], Yong Rui [§]

[†]University of Science and Technology of China, Hefei, China [‡]Sun Yat-Sen University [§]Microsoft Research, Beijing, China

{panyw.ustc, yehaoli.sysu}@gmail.com, {tiyao, tmei, yongrui}@microsoft.com, lihq@ustc.edu.cn

Abstract

Learning video representation is not a trivial task, as video is an information-intensive media where each frame does not exist independently. Locally, a video frame is visually and semantically similar with its adjacent frames. Holistically, a video has its inherent structure—the correlations among video frames. For example, even the frames far from each other may also hold similar semantics. Such context information is therefore important to characterize the intrinsic representation of a video frame. In this paper, we present a novel approach to learn the deep video representation by exploring both local and holistic contexts. Specifically, we propose a triplet sampling mechanism to encode the local temporal relationship of adjacent frames based on their deep representations. In addition, we incorporate the graph structure of the video, as a priori, to holistically preserve the inherent correlations among video frames. Our approach is fully unsupervised and trained in an end-to-end deep convolutional neural network architecture. By extensive experiments, we show that our learned representation can significantly boost several video recognition tasks (retrieval, classification, and highlight detection) over traditional video representations.

1 Introduction

Video has become ubiquitous. This has encouraged the development of advanced techniques to video understanding for a wide variety of applications. One of the fundamental problems is how to learn a "good" representation for a video. A valid question is then what are the generic priors for learning the intrinsic representation of a video that can deal with complex variations without specifying any task.

In general, video is a sequence of frames with large content variance and complexity. There are two kinds of contextual information to be exploited for learning video representations: local temporal coherence and holistic graph structure. First, the adjacent video frames are usually visually and semantically coherent. This can be regarded as an intrinsic property of video to learn a possibly "good" representation with respect to the large variations in a video. Such temporal coherence context has been successfully applied to metric learning, as a regularizer in both the supervised learning [Mobahi *et al.*, 2009] and unsupervised learning [Wiskott and Sejnowski, 2002].

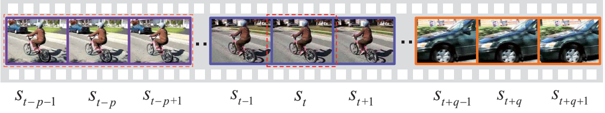

Figure 1: Two generic priors for learning video representations: local temporal coherence and holistic graph structure preservation. Taking the feature learning for frame s_t as an example, the temporal coherence can preserve the visual and semantical similarity among adjacent frames, shown in the same color, i.e., $\{s_{t-1}, s_t, s_{t+1}\}$. Moreover, we can observe some non-adjacent frames also contain similar semantics with s_t, e.g., s_{t-p-1}, s_{t-p}, and s_{t-p+1}. Such inherent semantic correlations is expected to be encoded for learning the representation for the frame s_t.

Second, in addition to the local temporal coherence, a video has its inherent structure over the entire sequence where the frames far from each other may also exhibit similar semantics. Such holistic structure can be viewed as a graph constructed on all the video frames. In this context, learning the graph structure amounts to estimating the similarity matrix on the representations of video frames. It offers the advantages with respect to learning directly from the video and reveals the correlations between video frames. Therefore, another generic prior for learning the "good" video representations is to preserve the graph structure estimated on the learned representations of video frames.

Figure 1 shows an intuitive example of the two priors in video representation learning. By jointly integrating the temporal coherence and graph structure preservation, we present a novel Temporal and Graph-structured Feature Learning (TGFL) approach to learning the representations of video frames. Specifically, a video is represented by a sequence of frames. The temporal coherence is then characterized with a set of frame triplets. Each triplet contains a query frame, a positive frame, and a negative frame, where the positive

[*]This work was performed when Yingwei Pan and Yehao Li were visiting Microsoft Research as research interns. The first two authors contributed equally to this work.

frame is visually similar with and adjacent to the query frame while the negative one is dissimilar with and far from the query. Meanwhile, the graph structure of the video is estimated on a similarity matrix among all the video frames. The spirit of TGFL is to learn the video representation in a deep architecture by simultaneously exploiting the local relative similarity ordering in the triplets and preserving the holistic structure in the entire video. It is worth noticing that our proposed approach to feature learning is generic and applicable to any other sequence data. Different from previous methods for video feature learning which predominantly focus on modeling temporal coherence [Goroshin *et al.*, 2014; Ranzato *et al.*, 2014; Srivastava *et al.*, 2015], we explore both the local (temporal coherence) and holistic (graph structure) contexts to learn the intrinsic video representation.

The contributions of this paper are as follows. First, we study the generic priors which lead to a "good" representation of video frames. Second, we propose a novel approach for learning frame-level video representations in a deep network architecture, which aims to incorporate both temporal coherence and graph structure preservation. Our approach is fully unsupervised and trained in an end-to-end fashion. Specifically, we design a novel deep neural network architecture integrating the proposed temporal and graph-structured loss layer to optimize the whole deep convolutional neural network (DCNN) structure. Third, we demonstrate by extensive experiments that our proposed feature learning outperforms several state-of-the-art representations in three video recognition tasks.

2 Related Work

We group the related work into two categories: feature learning for videos, and graph structure preservation models. The first category reviews the research in feature learning for videos by exploiting spatio-temporal properties, while the second investigates a variety of models considering graph structure preservation.

Feature learning for videos. Learning feature representation for videos is a fundamental yet challenging problem. Le *et al.* use Independent Subspace Analysis (ISA) to learn spatio-temporal feature from unlabeled video data [Le *et al.*, 2011]. Wiskott and others propose that the invariant feature representation can be learnt by maximizing the temporal coherence in video [Wiskott and Sejnowski, 2002; Hurri and Hyvärinen, 2003]. Recently, the work in [Goroshin *et al.*, 2014] utilizes the auto-encoder to learn video features with a temporally and semantically coherence metric. In addition, the Recurrent Neural Networks (RNN) which can model sequence dynamics is also explored for feature learning in video. In [Ranzato *et al.*, 2014], the proposed RNN-based model for feature learning in video explored both spatial and temporal correlations of videos. A Long Short Term Memory (LSTM) Encoder-Decoder model is proposed for feature representation learning and the prediction of video frame [Srivastava *et al.*, 2015].

Graph structure preservation models. Graph structure preservation models aim to preserve the global topological properties of the input graph-structured data, which have shown effective for dimensionality reduction [Tenenbaum *et al.*, 2000; Yan *et al.*, 2007], semi-supervised learning [Melacci and Belkin, 2011; Qi *et al.*, 2012], image search [Pan *et al.*, 2014], video annotation [Moxley *et al.*, 2010] and transfer learning [Long *et al.*, 2014]. In addition, there are also several works considering such graph structure in the deep network architecture. For example, the work in [Bruna *et al.*, 2013] exploits the global structure of graph with the spectrum of Graph Laplacian to generalize convolution operator in the CNN architectures. Furthermore, the spectral network introduced in [Bruna *et al.*, 2013] is extended to deep network architectures with small learning complexity on non-Euclidean domains by incorporating a graph estimation procedure.

Summary. We focus on learning feature representation for video. Different from previous methods for video feature learning which predominantly focus on modeling temporal coherence, we explore both the local (temporal coherence) and holistic (graph structure) contexts to learn the intrinsic video representation.

3 Approach: Temporal and Graph-structured Feature Learning

Our proposed Temporal and Graph-structured Feature Learning (TGFL) approach is to build an embedding space in which the feature representations for frames can be encoded with both temporal coherence and graph structure contexts. The training of TGFL is performed by simultaneously minimizing the triplet ranking loss to characterize temporal coherence among adjacent frames, and preserving the holistic graph structure relationships among all the video frames. Therefore, the objective function of the TGFL consists of of two components, i.e., triplet ranking loss based on the sampled triplets and the graph structure preservation in videos.

Figure 2 shows the overview of our approach. In the following, we will first define the video sequence and the representation of each video frame in the embedding space, followed by constructing the two learning components (temporal coherence and graph structure) for feature learning. It is worth noticing that to precisely measure the temporal coherence, we present a triplet sampling mechanism from the viewpoint of mutual reinforcement between the temporal structure and visual similarity among frames. Then, we formulate the joint objective function and provide the optimization strategy in a deep learning framework. Specifically, we design a novel deep neural network architecture consisting of multiple convolution-pooling layers and a fully connected layer, followed by the proposed temporal and graph-structured loss layer to optimize the whole DCNN structure.

3.1 Notation

As our feature learning approach is unsupervised, a large collection of videos is desired. Suppose we have a collection of videos $\mathcal{V}$ where each video $v \in \mathcal{V}$ can be represented as a temporal sequence of N sampled frames (uniform sampling) $\{s_1, s_2, \ldots, s_N\}$. Let $\mathcal{S}$ denote the frame space. The goal of feature learning for video frames is to construct a mapping $f : \mathcal{S} \rightarrow \mathbb{R}^d$, such that each frame can be mapped into a d-dimensional embedding space. Note that with the mapping

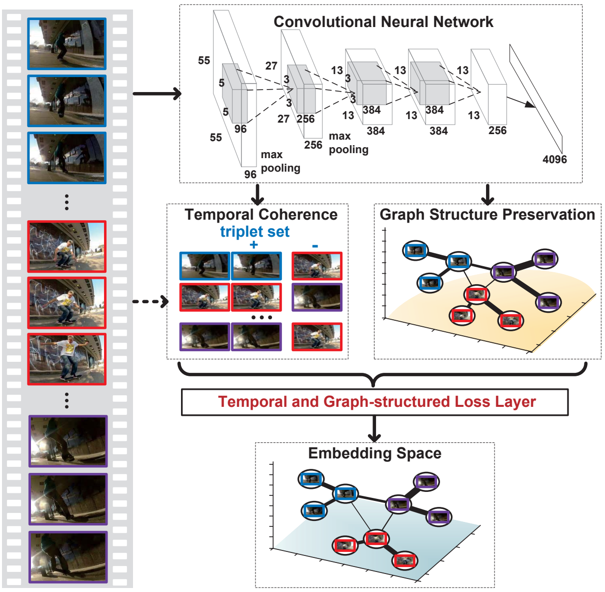

Figure 2: The overview of our TGFL framework for learning intrinsic video representation (better viewed in color). The temporal coherence is to characterize the relative temporal relationships through frame triplet set, while the holistic graph structure is to preserve the inherent correlations among frames. Both of the two priors are simultaneously exploited in our designed temporal and graph-structured loss layer, which is designed on the top of full connected fc6 layer in AlexNet. Our feature learning is fully unsupervised and trained in an end-to-end fashion.

function $f(s)$, the entire sequential frames of video v are projected into this embedding space, which are represented as $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N]^\top \in \mathbb{R}^{N \times d}$.

3.2 Modeling Temporal Coherence

Given a sequence of frames, we aim to exploit the temporal coherence as one generic prior to learn effective representations for video frames. The prior is that the temporally adjacent frames are more likely to be semantically similar than those non-adjacent frames. Therefore, when measuring the representations of video frames in the embedding space, the similarity between a pair of temporally adjacent frames should be higher than that of the pair of non-adjacent frames.

We first define the similarity of two frames $\mathbf{f}_i$ and $\mathbf{f}_j$ according to their Euclidean distances in the embedding space:

$$D\left(\mathbf{f}_i, \mathbf{f}_j\right) = \|\mathbf{f}_i - \mathbf{f}_j\|_2^2 \,. \tag{1}$$

Then, we characterize the temporal coherence of a temporal sequence by the ranking loss measured on a set of triplets, which can be easily fed into the feature learning framework. Denote $\mathcal{T}$ as the set of triplets generated from temporal sequence, and each triplet as $\langle \mathbf{f}_i, \mathbf{f}_j^+, \mathbf{f}_m^- \rangle$ consisting of the query frame $\mathbf{f}_i$, positive frame $\mathbf{f}_j^+$ and negative frame $\mathbf{f}_m^-$. Accordingly, the triplet ranking loss is defined by

$$\ell_{triplet}(\mathbf{f}_i, \mathbf{f}_j^+, \mathbf{f}_m^-) = \max\{0, D(\mathbf{f}_i, \mathbf{f}_j^+) - D(\mathbf{f}_i, \mathbf{f}_m^-) + 1\} \tag{2}$$

The triplet ranking loss exploits the margin ranking loss [Herbrich *et al.*, 2000] which is widely used in feature learning [Pan *et al.*, 2015; Wang *et al.*, 2014]. By minimizing the ranking loss on the set of triplets $\mathcal{T}$, the relative distance relationship on the feature representations of frames in the embedding space is preserved to present the temporal coherence. Specifically, for each triplet $\langle \mathbf{f}_i, \mathbf{f}_j^+, \mathbf{f}_m^- \rangle$, we aim to make the embedding features in close proximity of $\mathbf{f}_i$ and $\mathbf{f}_j^+$, and simultaneously obtain a large distance between $\mathbf{f}_i$ and $\mathbf{f}_m^-$.

The triplet ranking loss is convex and its gradients with respect to $\mathbf{f}_i, \mathbf{f}_j^+, \mathbf{f}_m^-$ are

$$\begin{aligned}
\frac{\partial \ell_{triplet}}{\partial \mathbf{f}_i} &= (2\mathbf{f}_m^- - 2\mathbf{f}_j^+) \times I_{D\left(\mathbf{f}_i, \mathbf{f}_j^+\right) - D\left(\mathbf{f}_i, \mathbf{f}_m^-\right) + 1 > 0} \\
\frac{\partial \ell_{triplet}}{\partial \mathbf{f}_j^+} &= (2\mathbf{f}_j^+ - 2\mathbf{f}_i) \times I_{D\left(\mathbf{f}_i, \mathbf{f}_j^+\right) - D\left(\mathbf{f}_i, \mathbf{f}_m^-\right) + 1 > 0} \,. \\
\frac{\partial \ell_{triplet}}{\partial \mathbf{f}_m^-} &= (2\mathbf{f}_i - 2\mathbf{f}_m^-) \times I_{D\left(\mathbf{f}_i, \mathbf{f}_j^+\right) - D\left(\mathbf{f}_i, \mathbf{f}_m^-\right) + 1 > 0}
\end{aligned} \tag{3}$$

The indicator function $I_C = 1$, if the condition C is true (i.e., $D\left(\mathbf{f}_i, \mathbf{f}_j^+\right) - D\left(\mathbf{f}_i, \mathbf{f}_m^-\right) + 1 > 0$); otherwise $I_C = 0$.

To learn and construct this embedding space, we incorporate the triplet ranking loss as a regularization in learning the mapping function.

Triplet Sampling. When generating triplet set $\mathcal{T}$ from the temporal sequence, one natural way is to randomly select triplets according to the temporal structure based on the assumption that the adjacent frames should be semantically similar while the non-adjacent frames (i.e., with a large time interval) are more likely to be dissimilar in semantics. However, in practice, due to camera shaking or movement, there usually exists the situation that the adjacent frames may have totally different semantics. In addition, it is also possible to find the similar semantics between two frames even with a long time interval. To avoid injecting negative triplets with noise into our feature learning framework, we propose a triplet sampling mechanism from the viewpoint of mutual reinforcement between temporal structure and visual relationships among frames. Given a query frame $\mathbf{f}_i$, we firstly generate a ranking list for all the frames in this temporal sequence based on their Euclidean distances to $\mathbf{f}_i$. Then, only the frames which are both temporally close to the query frame and visually similar at the top of the ranking list will be selected as positive frames $\mathbf{f}_j^+$. Meanwhile, we choose the negative frames $\mathbf{f}_m^-$ which are distant from the query frame and visually dissimilar at the bottom of the ranking list. Note that during each update process of training, the distance ranking list need to be updated based on the features from pervious iteration to generate evolved triplet set.

Therefore, with this rigorous triplet sampling mechanism, the triplet set is collected by considering both temporal structure and visual relationships among the video frames. Our model is benefited from this mechanism to better learn video representations.

3.3 Graph Structure Preservation

Graph structure preservation can be regarded as another regularization indicating that similar points in the original space should be mapped into the positions closely in the embedding space. Technically, we view the holistic structure of video as a graph constructed on the frames in the whole video sequence. The estimation of the underlying graph structure can be measured by the appropriate pairwise similarity between the video frames, which is given by:

$$\ell_{graph} = \sum_{i,j=1}^{N} \mathbf{S}_{ij} \left\|\mathbf{f}_i - \mathbf{f}_j\right\|_2^2, \tag{4}$$

where $\mathbf{S} \in \mathbb{R}^{N\times N}$ denotes the affinity matrix defined on the entire frames among the temporal sequence. Under the graph structure preservation criterion, it is reasonable to minimize Eq. (4), as it will incur a heavy penalty if two visually similar frames are mapped far away in the learnt embedding space.

There are many ways of defining the affinity matrices $\mathbf{S}$. Inspired by [Fang and Zhang, 2013], the elements are computed by Gaussian functions in this work, i.e.,

$$\mathbf{S}_{ij} = \begin{cases} e^{-\frac{\left\|\tilde{\mathbf{f}}_i - \tilde{\mathbf{f}}_j\right\|_2^2}{\sigma^2}} & if\ \tilde{\mathbf{f}}_i \in N_k(\tilde{\mathbf{f}}_j)\ or\ \tilde{\mathbf{f}}_j \in N_k(\tilde{\mathbf{f}}_i) \\ 0 & otherwise \end{cases}, \tag{5}$$

where σ is the bandwidth parameter. It should be noted that $\tilde{\mathbf{f}}_i$ denotes the learnt frame-level feature from pervious iteration in training and $N_k(\tilde{\mathbf{f}}_i)$ represents the set of k nearest neighbors of $\tilde{\mathbf{f}}_i$.

By defining the graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{S}$, where $\mathbf{D}$ is a diagonal matrix with its elements defined as $\mathbf{D}_{ii} = \sum_j \mathbf{S}_{ij}$, Eq. (4) can be rewritten as

$$\ell_{graph} = tr(\mathbf{F}^\top \mathbf{L}\mathbf{F}), \tag{6}$$

and its gradient with respect to $\mathbf{F}$ is

$$\frac{\partial \ell_{graph}}{\partial \mathbf{F}} = 2\mathbf{L}\mathbf{F}. \tag{7}$$

By minimizing this term, the inherent structure between frames can be preserved in the learnt embedding space. We additionally include this regularizer in our framework.

3.4 Formulation

The overall objective function integrates both the triplet ranking loss in Eq. (2) on the triplet set $\mathcal{T}$ and the graph structure preservation in Eq. (6). Hence we get the following overall loss objective

$$\ell = \lambda \sum_{t\in\mathcal{T}} \ell_{triplet}^{(t)} + (1-\lambda)\ell_{graph}, \tag{8}$$

where $\lambda \in [0, 1]$ is the tradeoff parameter.

With this overall loss objective, the crucial goal for its optimization is to learn the mapping function f. Inspired by the success of DCNN on feature learning for image [Krizhevsky *et al.*, 2012; Wang *et al.*, 2014; Feng *et al.*, 2016] and video [Ramanathan *et al.*, 2015; Zha *et al.*, 2015; Gan *et al.*, 2015] tasks, we employ a deep neural network architecture to learn the feature representation for video frames. Specifically, the embedding feature representation is leant on top of the fully connected fc6 layer of AlexNet [Krizhevsky *et al.*, 2012], which is pre-trained on ImageNet ILSVRC12 dataset [Russakovsky *et al.*, 2014]. In the training stage, to solve the optimization according to overall loss objective in Eq. (8), we design a temporal and graph-structured loss (TGL) layer on the top of fully connected fc6 layer in AlexNet. It is also worth noticing that we use L_2 normalization layer to normalize the output of fc6 layer and then feed the normalized results into our TGL layer. The TGL layer does not have any parameter. During learning, it evaluates the model's violation of two generic priors of temporal coherence and graph structure preservation, and back-propagates the gradients to the lower layers so that the lower layers can adjust their parameters to minimize the overall loss. The training process of TGL layer is given in Algorithm 1.

Algorithm 1 The Training of TGL Layer

```
Given a tradeoff parameter $\lambda$.
**Forward Pass:**
  Fetch input batch $\mathbf{F}$ with $N$ sample frames in one video.
  Generate selected triplet set $\mathcal{T}$.
  Compute all the triplet ranking losses on $\mathcal{T}$ via Eq. (2).
  Update affinity matrices $\mathbf{S}$.
  Compute graph structure preservation loss via Eq. (6).
  Compute overall loss output with tradeoff parameter $\lambda$.
**Backward Pass:**
  Compute gradient w.r.t input for triplet ranking loss via Eq. (3).
  Compute gradient w.r.t input for graph structure preservation
    via Eq. (7).
  Backward the overall gradient w.r.t input with tradeoff parameter $\lambda$ to lower layers.
```

4 Experiments

We evaluate our video representation by conducting three video recognition tasks (retrieval, classification, and highlight detection) on two popular video datasets, i.e., Columbia Consumer Videos (CCV) [Jiang *et al.*, 2011] which is a benchmark of consumer video retrieval and classification tasks, and YouTube Highlight [Sun *et al.*, 2014] which is an unconstrained first person video dataset for highlight detection.

4.1 Dataset and Settings

CCV. CCV dataset contains $9,317$ videos collected from YouTube. It consists of 20 semantic classes including popular events, e.g., "birthday party," "cats," "playground," and "graduation ceremony." For the video retrieval task, given a test query frame, the task is to estimate the similarity between each frame and query frame measured on their learned representations. Furthermore, for each query frame, we order all the frames based on the similarity scores. In the experiments, we use a subset of $5,803$ videos whose durations are at least 25 sec with $2,903$ videos for training and 2,900 videos for testing. The ground truth data are carefully generated on the testing videos. Specifically, following [Goroshin *et al.*, 2014], videos in the test set are automatically segmented into scenes by detecting large L_2 changes among adjacent frames. We randomly select 2,000 scenes from testing videos and use the middle frame from each scene as the test query frame. The entire pool for retrieval consists of 1.13 million sampled frames from both the training and testing videos. For each test query frame, only the temporal neighbors from the same scene are defined as semantically similar samples. The other frames in the pool are all used as dissimilar ones w.r.t the

Method	VR	VC	VHD
fc6	67.52%	62.82%	42.89%
fc7	64.89%	62.18%	42.41%
TE [Ramanathan *et al.*, 2015]	68.42%	63.15%	44.98%
Temporal Coherence (TC)	70.24%	64.27%	47.06%
Graph Structure (GS)	70.26%	63.36%	46.02%
TGFL	**72.13%**	**65.26%**	**48.08%**

Table 1: MAP of different methods for Video Retrieval (VR), Video Classification (VC), and Video Highlight Detection (VHD) tasks.

test query frame. For the video classification task, we use the same train/test splits as retrieval task (i.e., $2,903$ videos for training and $2,900$ for testing).

YouTube Highlight. This dataset is collected from YouTube for six domains including "skating," "gymnastics," "dog," "parkour," "surfing," and "skiing." Each domain contains about 100 videos with various lengths. The total duration is $1,430$ minutes. After removing unvalid and no-highlight videos, the dataset is partitioned into two parts: a training set with 352 videos and a testing set with 110 videos. For each testing video, the ground-truth highlighted moments are human labeled on Amazon Mechanical Turk.

Settings. In the experiments, we uniformly pick up three frames every second and compose $N = 256$ frames of each video. For a long video, only the first selected N frames are represented as a temporal sequence. All the positive frames are selected from the adjacent frames within one second around the query frame. The k nearest neighbors preserved in Eq. (5) and tradeoff parameter λ in Eq. (8) are both determined by using a validation set for each task. Finally, for both retrieval and classification tasks, $k = 10$ and $\lambda = 0.4$. For highlight detection task, we set $k = 8$ and $\lambda = 0.5$. In video retrieval task, we retrieve the frames in the whole dataset according to their cosine similarities w.r.t the query frame measured on our learned representation. In video classification task, following [Ramanathan *et al.*, 2015], we uniformly sample four frames per video and perform "mean pooling" process over all sampled frames to generate the video representation. The linear Support Vector Machine (SVM) is adopted to classify the videos. In video highlight detection task, following the pairwise ranking model proposed in [Sun *et al.*, 2014], we utilize the same linear ranking SVM to rank and detect video highlights.

4.2 Compared Methods

In video retrieval task, we use mean average precision (MAP) to evaluate the retrieval quality for test query frames. For video classification task, following [Jiang *et al.*, 2011], the average precision (AP) is used to measure performance for each class and MAP is adopted to report the overall performance. For highlight detection task, within each video, the best method should first detect the ground truth highlighted moments rather than other moments. Hence, we also calculate AP of highlight detection for each testing video and use MAP to evaluate the learnt feature to highlight detection.

To evaluate our model, we compare the following methods on retrieval, classification, and highlight detection tasks:

(1) fc6 and fc7: feature extracted from the top of the fully connected fc6 or fc7 layer in AlexNet pre-trained on ImageNet ILSVRC12 dataset [Russakovsky *et al.*, 2014].

(2) Temporal Embedding (TE) [Ramanathan *et al.*, 2015]: feature learning in a margin ranking loss based embedding framework to make the contextual representations in close proximity to the target frame and simultaneously dissimilar to other negative frames in a pairwise manner.

(3) Temporal and Graph-structured Feature Learning (TGFL) based on our proposal presented in Algorithm 1. Two slightly different runs are named as Temporal Coherence (TC) and Graph Structure (GS), which consider individual temporal coherence or graph structure preservation in the overall objective (Eq. (8)), respectively.

4.3 Performance Comparison

Evalution of video retrieval. Table 1 shows the MAP performances of six runs on three tasks. Overall, for video retrieval task, our TGFL consistently outperforms the other runs. In particular, the MAP of TGFL can achieve 72.13%, which makes the improvement over fc6 by 6.8%. Furthermore, TGFL can be further improved with large quantities of unlabeled videos, which are largely available and freely accessible on Web. There is a clear performance gap between the two runs TC and TE. Though both runs involve utilization of temporal context, they are fundamentally different in the way that the learnt representations of TE are as a result of embedding the target frame by its contextual representations in a pairwise manner, and TC is by characterizing relative temporal relationships through a set of frame triplets. The results basically indicate the advantage of learning video representations by exploiting temporal coherence. Moreover, TGFL by further preserving graph structure is superior to TC, which indicates that the two principles of temporal coherence and graph structure reinforce each other in feature learning. Figure 3 further illustrates the top eight retrieved video frames in response to query frame based on the learnt representations by different methods. We can clearly see that the proposed TGFL gets more satisfying ranking results and retrieves five relevant video frames in the returned top eight frames.

Evaluation of video classification. The MAP performances for video classification task are reported in the third column of Table 1. Our TGFL still consistently outperforms other baselines, which makes the improvement over fc6 by 3.9%. The performance gain can be attributed to the video feature learning by exploiting both temporal coherence and graph structure simultaneously.

Evaluation of video highlight detection. The fourth column in Table 1 shows the MAP values of different approaches for video highlight detection task. Overall, TGFL consistently exhibits better performance than other approaches. Compared to fc6, TGFL raises the MAP from 42.89% to 48.08%, making the improvement by 12.1%. Similar to the observations in video retrieval and classification tasks, TC exhibits better performance than TE, but shows worse performance than TGFL. Figure 4 shows eight segments uniformly sampled from a video of "surfing," "parkour," "skating," and "gymnastics." Each segment is represented by one sampled frame. As illustrated in the figure, the eight segments

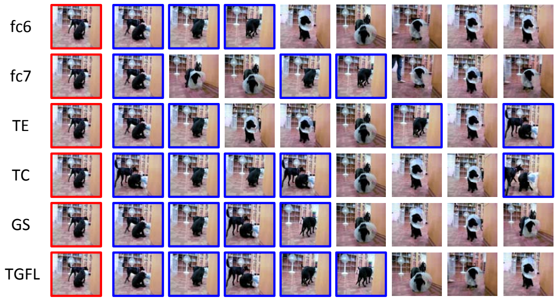

Figure 3: Examples showing the top eight video frames retrieval results based on the learnt representations by different methods in response to query frame. In each row, the first frame with a red bounding box is the query frame and the similar video frames in the retrieved list are enclosed in blue bounding boxes.

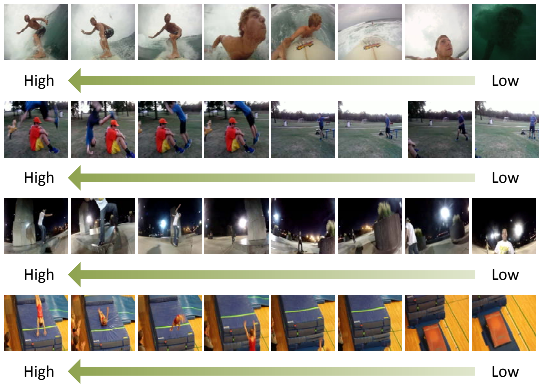

Figure 4: Examples of segments ranking from low (right) to high (left) according to our predicted highlight scores for "surfing," "parkour," "skating," and "gymnastics" categories.

are ranked according to their predicted highlight scores on the learnt representations by our TGFL and we can easily see that the ranking order reflects the relative degree of interest within a video.

4.4 Effect of the number of nearest neighbors k

In order to show the relationship between the performance and the number of nearest neighbors, we conducted experiments to evaluate the performance of our TGFL framework with the number of nearest neighbors in range of $\{5, 6, 7, 8, 9, 10, 11, 12\}$. The MAP with different number of nearest neighbors are shown in Figure 5. As illustrated in the figure, TGFL achieves the best results when we choose Top-10 nearest neighbors on video retrieval and classification tasks while the optimal $k = 8$ in video highlight detection task. Furthermore, the performance difference by using different number of nearest neighbors is within 0.01 on all three tasks, which basically verifies that our TGFL has a good property of being affected very slightly when choosing different number of nearest neighbors.

4.5 Effect of the tradeoff parameter λ

A common problem with multiple regularization terms in a joint optimization objective is the need to set the parameters to tradeoff each component. In the previous experiments, the tradeoff λ is optimally set in order to examine the performance of λ on video retrieval, classification, and highlight detection irrespective of the parameter influence. We further conducted experiments to test the sensitivity of λ towards the three video recognition tasks.

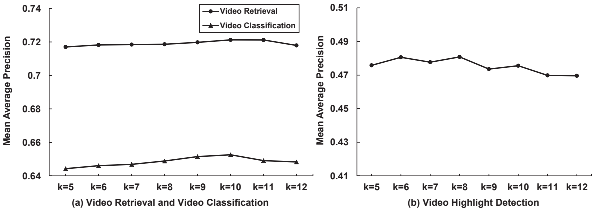

Figure 5: The MAP performance curves with different numbers of nearest neighbors on (a) video retrieval and video classification and (b) video highlight detection, respectively.

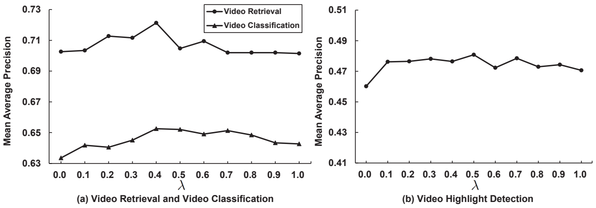

Figure 6: The MAP performance curves with different tradeoff parameter λ on (a) video retrieval and video classification and (b) video highlight detection, respectively.

Figure 6 shows the MAP performance with respect to different values of λ. We can see that all the performance curves are smooth when λ varies in a range from $\{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. The performances fluctuate within the range of 0.02 on three tasks. Thus, it is not sensitive to the change of the tradeoff parameter. More importantly, the fusion of temporal coherence and graph structure by any tradeoff weights consistently leads to a performance boost against individual component ($\lambda = 0$ or $\lambda = 1.0$). The result again confirms the advantage of exploiting both principles of temporal coherence and graph structure which are complementary for feature learning.

5 Conclusions

We have showed that learning a good video representation should take both the local temporal coherence from adjacent video frames and holistic intrinsic structure among all the frames into consideration. We present a temporal and graph-structured feature learning approach to learn the intrinsic representation of video frames by exploiting such context information. We also validate the effectiveness of the learned representation through extensive experiments on three video recognition tasks. Our future works include: 1) using RNN to better model the temporal coherence, and 2) investigating different pooling schemes for transferring our frame representation to video-level representation.

Acknowledgments. This work was supported in part by the 973 Programme under contract No. 2015CB351803, NSFC under contract No. 61325009 and No. 61390514.

References

[Bruna *et al.*, 2013] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In *ICLR*, 2013.

[Fang and Zhang, 2013] Zheng Fang and Zhongfei Mark Zhang. Discriminative feature selection for multi-view cross-domain learning. In *CIKM*, 2013.

[Feng *et al.*, 2016] Jie Feng, Yan Wang, and Shih-Fu Chang. 3d shape retrieval using a single depth image from low-cost sensors. In *WACV*, 2016.

[Gan *et al.*, 2015] Chuang Gan, Naiyan Wang, Yi Yang, Dit-Yan Yeung, and Alex G Hauptmann. DevNet: A deep event network for multimedia event detection and evidence recounting. In *CVPR*, 2015.

[Goroshin *et al.*, 2014] Ross Goroshin, Joan Bruna, Jonathan Tompson, David Eigen, and Yann LeCun. Unsupervised learning of spatiotemporally coherent metrics. *arXiv preprint arXiv:1412.6056*, 2014.

[Herbrich *et al.*, 2000] Ralf Herbrich, Thore Graepel, and Klaus Obermayer. *Large Margin Rank Boundaries for Ordinal Regression*. MIT Press, 2000.

[Hurri and Hyvärinen, 2003] Jarmo Hurri and Aapo Hyvärinen. Simple-cell-like receptive fields maximize temporal coherence in natural video. *Neural Computation*, 2003.

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

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[Le *et al.*, 2011] Quoc V Le, Will Y Zou, Serena Y Yeung, and Andrew Y Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, 2011.

[Long *et al.*, 2014] Mingsheng Long, Jianmin Wang, Guiguang Ding, Dou Shen, and Qiang Yang. Transfer learning with graph co-regularization. *Knowledge and Data Engineering, IEEE Transactions on*, 2014.

[Melacci and Belkin, 2011] S. Melacci and M. Belkin. Laplacian support vector machines trained in the primal. *Journal of Machine Learning Research*, 12:1149–1184, 2011.

[Mobahi *et al.*, 2009] H. Mobahi, R. Collobert, and J. Weston. Deep learning from temporal coherence in video. In *ICML*, 2009.

[Moxley *et al.*, 2010] Emily Moxley, Tao Mei, and Bangalore S Manjunath. Video annotation through search and graph reinforcement mining. *Multimedia, IEEE Transactions on*, 12(3):184–193, 2010.

[Pan *et al.*, 2014] Yingwei Pan, Ting Yao, Tao Mei, Houqiang Li, Chong-Wah Ngo, and Yong Rui. Click-through-based cross-view learning for image search. In *SIGIR*, 2014.

[Pan *et al.*, 2015] Yingwei Pan, Ting Yao, Houqiang Li, Chong-Wah Ngo, and Tao Mei. Semi-supervised hashing with semantic confidence for large scale visual search. In *SIGIR*, 2015.

[Qi *et al.*, 2012] Zhiquan Qi, Yingjie Tian, and Yong Shi. Laplacian twin support vector machine for semi-supervised classification. *Neural Networks*, 2012.

[Ramanathan *et al.*, 2015] Vignesh Ramanathan, Kevin Tang, Greg Mori, and Li Fei-Fei. Learning temporal embeddings for complex video analysis. In *ICCV*, 2015.

[Ranzato *et al.*, 2014] MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014.

[Russakovsky *et al.*, 2014] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2014.

[Srivastava *et al.*, 2015] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using LSTMs. In *ICML*, 2015.

[Sun *et al.*, 2014] Min Sun, Ali Farhadi, and Steve Seitz. Ranking domain-specific highlights by analyzing edited videos. In *ECCV*, 2014.

[Tenenbaum *et al.*, 2000] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 2000.

[Wang *et al.*, 2014] Jiang Wang, Yang Song, Tommy Leung, Catherine Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, 2014.

[Wiskott and Sejnowski, 2002] Laurenz Wiskott and Terrence J. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, 2002.

[Yan *et al.*, 2007] Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and Stephen Lin. Graph embedding and extensions: a general framework for dimensionality reduction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2007.

[Zha *et al.*, 2015] Shengxin Zha, Florian Luisier, Walter Andrews, Nitish Srivastava, and Ruslan Salakhutdinov. Exploiting image-trained CNN architectures for unconstrained video classification. In *BMVC*, 2015.