Learning Temporal Embeddings for Complex Video Analysis

Vignesh Ramanathan[1] Kevin Tang[2] Greg Mori[3] Li Fei-Fei[2]
[1]Department of Electrical Engineering, Stanford University
[2]Computer Science Department, Stanford University
[3]School of Computing Science, Simon Fraser University
vigneshr@cs.stanford.edu, kdtang@cs.stanford.edu, mori@cs.sfu.ca, feifeili@cs.stanford.edu

Abstract

In this paper, we propose to learn temporal embeddings of video frames for complex video analysis. Large quantities of unlabeled video data can be easily obtained from the Internet. These videos possess the implicit weak label that they are sequences of temporally and semantically coherent images. We leverage this information to learn temporal embeddings for video frames by associating frames with the temporal context that they appear in. To do this, we propose a scheme for incorporating temporal context based on past and future frames in videos, and compare this to other contextual representations. In addition, we show how data augmentation using multi-resolution samples and hard negatives helps to significantly improve the quality of the learned embeddings. We evaluate various design decisions for learning temporal embeddings, and show that our embeddings can improve performance for multiple video tasks such as retrieval, classification, and temporal order recovery in unconstrained Internet video.

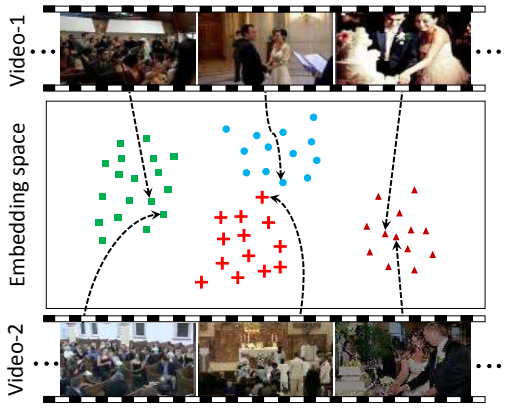

Figure 1. The temporal context of a video frame is crucial in determining its true semantic meaning. For instance, consider the above example where the embeddings of different semantic classes are shown in different colors. The middle frame from the two wedding videos correspond to visually dissimilar classes of "church ceremony" and "court ceremony". However, by observing the similarity in their temporal contexts we expect them to be semantically closer. Our work leverages such powerful temporal context to learn semantically rich embeddings.

1. Introduction

Video data is plentiful and a ready source of information – what can we glean from watching massive quantities of videos? At a fine granularity, consecutive video frames are visually similar due to temporal coherence. At a coarser level, consecutive video frames are visually distinct but semantically coherent.

Learning from this semantic coherence present in video at the coarser-level is the main focus of this paper. Purely from unlabeled video data, we aim to learn embeddings for video frames that capture semantic similarity by using the temporal structure in videos. The prospect of learning a generic embedding for video frames holds promise for a variety of applications ranging from generic retrieval and similarity measurement, video recommendation, to automatic content creation such as summarization or collaging. In this paper, we demonstrate the utility of our video frame embeddings for several tasks such as video retrieval, classification and temporal order recovery.

The idea of leveraging sequential data to learn embeddings in an unsupervised fashion is well explored in the Natural Language Processing (NLP) community. In particular, distributed word vector representations such as word2vec [23] have the unique ability to encode *regularities and patterns* surrounding words, using large amounts of unlabeled data. In the embedding space, this brings together words that may be very different, but which share similar contexts in different sentences. This is a desirable property we would like to extend to video frames as well as shown in Fig. 1. We would like to have a representation for frames which captures the semantic context around the frame beyond the visual similarity obtained from temporal coherence.

However, the task of embedding frames poses multiple

challenges specific to the video domain: 1. Unlike words, the set of frames across all videos is not discrete and quantizing the frames leads to a loss in information; 2. Temporally proximal frames within the same video are often visually similar and might not provide useful contextual information; 3. The correct representation of context surrounding a frame is not obvious in videos. The main contribution of our work is to propose a new ranking loss based embedding framework, along with a contextual representation specific to videos. We also develop a well engineered data augmentation strategy to promote visual diversity among the context frames used for embedding.

We evaluate our learned embeddings on the standard tasks of video event retrieval and classification on the TRECVID MED 2011 [28] dataset, and compare to several recently published spatial and temporal video representations [7, 33]. Aside from semantic similarity, the learned embeddings capture valuable information in terms of the temporal context shared between frames. Hence, we also evaluate our embeddings on two related tasks: 1. temporal frame retrieval, and 2. temporal order recovery in videos. Our embeddings improve performance on all tasks, and serves as a powerful representation for video frames.

2. Related Work

Video features. Standard tasks in video such as classification and retrieval require a well engineered feature representation, with many proposed in the literature [1, 8, 13, 21, 25, 26, 27, 29, 32, 39, 40]. Deep network features learned from spatial data [10, 15, 33] and temporal flow [33] have also shown comparable results. However, recent works in complex event recognition [41, 44] have shown that spatial Convolutional Neural Network (CNN) features learned from ImageNet [2] without fine-tuning on video, accompanied by suitable pooling and encoding strategies achieves state-of-the-art performance. In contrast to these methods which either propose handcrafted features or learn feature representations with a fully supervised objective from images or videos, we try to learn an embedding in an unsupervised fashion. Moreover, our learned features can be extended to other tasks beyond classification and retrieval.

There are several works which improve complex event recognition by combining multiple feature modalities [12, 24, 36]. Another related line of work is the use of subevents defined manually [7], or clustered from data [20] to improve recognition. Similarly, Yang et al. used low dimensional features from deep belief nets and sparse coding [42]. While these methods are targeted towards building features specifically for classification in limited settings, we propose a generic video frame representation which can capture semantic and temporal structure in videos.

Unsupervised learning in videos. Learning features with unsupervised objectives has been a challenging task in the image and video domain [9, 22, 37]. Notably, [22] develops an Independent Subspace Analysis (ISA) model for feature learning using unlabeled video. Recent work from [5] also hints at a similar approach to exploit the slowness prior in videos. Also, recent attempts extend such autoencoder techniques for next frame prediction in videos [31, 35]. These methods try to capitalize on the temporal continuity in videos to learn an LSTM [43] representation for frame prediction. In contrast to these methods which aim to provide a unified representation for a complete temporal sequence, our work provides a simple yet powerful representation for independent video frames and images.

Embedding models. The idea of learning and representing temporal continuity has been discussed in pioneering works like [3]. More recent works such as word2vec [23] learn embeddings such that words with similar contexts are closer to each other. Another interesting model based on a Markovian approach was also proposed in [6]. A related idea in computer vision is the embedding of text in the semantic visual space [4, 18] based on large image datasets labeled with captions or class names. While these methods focus on different scenarios for embedding text, the aim of our work is to generate an embedding for video frames.

3. Our Method

Given a large collection of unlabeled videos, our goal is to leverage their temporal structure to learn an effective embedding for video frames. We wish to learn an embedding such that the *context* frames surrounding each *target* frame can determine the representation of the *target* frame, similar to the intuition from word2vec [23]. For example, in Fig. 1, *context* such as "crowd" and "cutting the cake" provides valuable information about the *target* "ceremony" frames that occur in between. This idea is fundamental to our embedding objective and helps in capturing semantic and temporal interactions in video.

While the idea of representing frames by embeddings is lucrative, the extension from language to visual data is not straightforward. Unlike language we do not have a natural, discrete vocabulary of words. This prevents us from using a softmax objective as in the case of word2vec [23]. Further, consecutive frames in videos often share visual similarity due to temporal coherence. Hence, a naive extension of [23] does not lead to good vector representations of frames.

To overcome the problem of lack of discrete words, we use a ranking loss [14] which explicitly compares multiple pairs of frames across all videos in the dataset. This ensures that the *context* in a video scores the *target* frame higher than others in the dataset. We also handle the problem of visually similar frames in temporally smooth videos through a carefully designed sampling mechanism. We obtain context frames by sampling the video at multiple temporal scales, and choosing hard negatives from the same video.

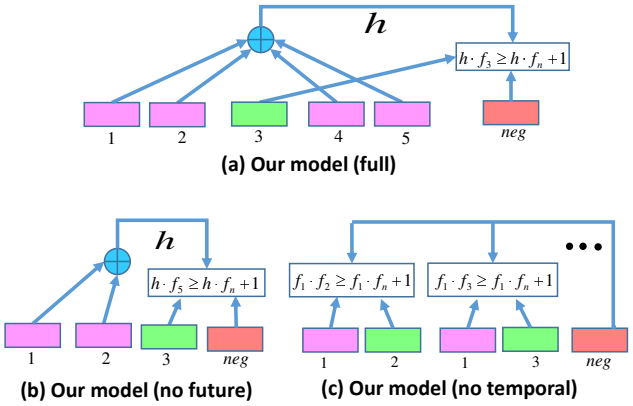

Figure 2. Visualizations of the temporal context of frames used in: (a) our model (full), (b) our model (no future), and (c) our model (no temporal). Green boxes denote target frames, magenta boxes denote contextual frames, and red boxes denote negative frames.

3.1. Embedding objective

We are given a collection of videos $\mathcal{V}$, where each video $v \in \mathcal{V}$ is a sequence of frames $\{s_{v1}, \ldots, s_{vn}\}$. We wish to obtain an embedding f_{vj} for each frame s_{vj}. Let $f_{vj} = f(s_{vj}; W_e)$ be the temporal embedding function which maps the frame s_{vj} to a vector. The model embedding parameters are given by W_e, and will be learned by our method. We embed the frames such that the *context* frames around the *target* frame predict the *target* frame better than other frames. The model is learned by minimizing the sum of objectives across all videos. Our embedding loss objective is shown below:

$$J(W_e) = \sum_{v \in \mathcal{V}} \sum_{\substack{s_{vj} \in v \\ s_- \neq s_{vj}}} \max\left(0, 1 - (f_{vj} - f_-) \cdot h_{vj}\right), \quad (1)$$

where f_- is the embedding of a negative frame s_-, and the context surrounding the frame s_{vj} is represented by the vector h_{vj}. Note that unlike the word vector embedding models in word2vec [23], we do not use an additional linear layer for softmax prediction on top of the context vector.

Another alternative could be a regression loss. However, as noted in [31], this can lead to low training error by simply blurring the representation of all frames in a video. We also experimented with multiple loss functions, and empirically found the ranking loss to perform the best.

3.2. Context representation

As we verify later in the experiments, the choice of context is crucial to learning good embeddings. A video frame at any time instant is semantically correlated with both past and future frames in the video. Hence, a natural choice for context representation would involve a window of frames centered around the *target* frame, similar to the skip-gram idea used in word2vec [23]. Along these lines, we propose a context representation given by the average of the frame embeddings around the *target* frame. Our context vector h_{vj} for a frame s_{vj} is:

$$h_{vj} = \frac{1}{2T} \sum_{t=1}^{T} f_{vj+t} + f_{vj-t}, \quad (2)$$

where T is the window size, and f_{vj} is the embedding of the frame s_{vj}. This embedding model is shown in Fig. 2(a). For negatives, we use frames from other videos as well as frames from the same video which are outside the temporal window, as explained in Sec. 3.4.

Two important characteristics of this context representation is that it 1. makes use of the temporal order in which frames occur and 2. considers contextual evidence from both past and future. In order to examine their effect on the quality of the learned embedding, we also consider two weaker variants of the context representation below.

Our model (no future). This one-sided contextual representation tries to predict the embedding of a frame in a video only based on the embeddings of frames from the past as shown in Fig. 2(b). For a frame s_{vj}, and window size T the context $h_{vj}^{nofuture}$ is given by:

$$h_{vj}^{nofuture} = \frac{1}{T} \sum_{t=1}^{T} f_{vj-t}. \quad (3)$$

Our model (no temporal). An even weaker variant of context representation is simple co-occurrence without temporal information. We also explore a contextual representation which completely neglects the temporal ordering of frames and treats a video as a bag of frames. The context h_{vj}^{notemp} for a target frame s_{vj} is sampled from the embeddings corresponding to all other frames in the same video:

$$h_{vj}^{notemp} \in \{f_{vk} \mid k \neq j\}. \quad (4)$$

This contextual representation is visualized in Fig. 2(c).

3.3. Embedding function

In the previous sections, we introduced a model for representing context, and now move on to discuss the embedding function $f(s_{ij}; W_e)$. In practice, the embedding function can be a CNN built from the frame pixels, or any underlying image or video representation. However, following the recent success of ImageNet trained CNN features for complex event videos [41, 44], we choose to learn an embedding on top of the fully connected fc6 layer feature representation obtained by passing the frame through a standard CNN [19] architecture. In this case, the underlying

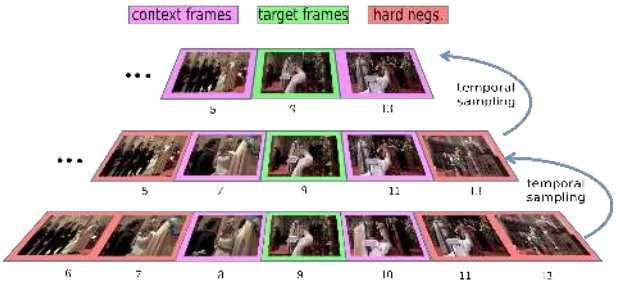

Figure 3. Multi-resolution sampling and hard negatives used in our full context model (T = 1). For a target frame (green), we sample context frames (magenta) at varying resolutions, as shown by the rows in this figure. We take hard negatives as examples in the same video that fall outside the context window (red).

representation is pre-trained from ImageNet domain which is vastly different from the TRECVID domain. Note that our method is agnostic to the choice of this underlying feature. Our learning procedure is still unsupervised, since we do not use any labels to learn our embeddings from these representations. We use a simple model with a fully connected layer followed by a rectified linear unit (ReLU) and local response normalization (LRN) layer, with dropout regularization. In this architecture, the learned model parameters W_e correspond to the weights and bias of our affine layer.

3.4. Data augmentation

We found that a careful strategy for sampling context frames and negatives is important to learning high quality embeddings in our models. This helps both in handling the problem of temporal smoothness and prevents the model from overfitting to less interesting video-specific properties.

Multi-resolution sampling. Complex events progress at different paces within different videos. Densely sampling frames in slowly changing videos can lead to context windows comprised of frames that are visually very similar to the target frame. On the other hand, a sparse sampling of fast videos could lead to context windows only composed of disjoint frames from unrelated parts of the video. We overcome these problems through multi-resolution sampling as shown in Fig. 3. For every target frame, we sample context frames from multiple temporal resolutions. This ensures a good trade-off between visual variety and semantic relatedness in the context windows.

Hard negatives. The context frames, as well as the target to be scored are chosen from the same video. This causes the model to cluster frames from the same video based on less interesting video-specific properties such as lighting, camera characteristics and background, without learning anything semantically meaningful. We avoid such problems by choosing hard negatives from within the same video as well. Empirically, this improves performance for all tasks. The negatives are chosen from outside the range of the context window within a video as depicted in Fig. 3.

3.5. Implementation details

The context window size was set to $T = 2$, and the embedding dimension to 4096. The learning rate was set to 0.01 and gradually annealed in steps of 5000. The training is typically completed within a day on 1 GPU with Caffe [11] for a dataset of approximately 40000 videos. All videos were first down-sampled to 0.2 fps before training.

4. Experimental Setup

Our embeddings are aimed at capturing semantic and temporal interactions within complex events in a video, and thus we require a generic set of videos with a good variety of actions and sub-events within each video. Most standard datasets such as UCF-101 [34] and Sport-1M [15] are comprised of short video clips capturing a single sports action, making them unsuitable for our purpose. Fortunately, the TRECVID MED 2011 [28] dataset provides a large set of diverse videos collected directly from YouTube. More importantly, these videos are not simple single clip videos; rather they are complex events with rich interactions between various sub-events within the same video [7]. Specifically, we learn our embeddings on the complete MED11 DEV and TEST sets comprised of 40021 videos. A subset of 256 videos from the DEV and TEST set was used for validation. The DEV and TEST sets are typical random assortments of YouTube videos with minimal constraints.

We compare our embeddings against different video representations for three video tasks: video retrieval, complex event classification, and temporal order recovery. All experiments are performed on the MED11 event kit videos, which are completely disjoint from the training and validation videos used for learning our embeddings. The event kit is composed of 15 event classes with approximately $100 - 150$ videos per event, with a total of 2071 videos.

We stress that the embeddings are learned in an unsupervised setting since we only use the temporal and semantic structure of the video data, without video labels. We do not tune them specifically to any event class.

5. Video Retrieval

In retrieval tasks, we are given a query, and the goal is to retrieve a set of related examples from a database. We start by evaluating our embeddings on two types of retrieval tasks: event retrieval and temporal retrieval. The retrieval tasks help to evaluate the ability of our embeddings to group together videos belonging to the same semantic event class and frames that are temporally coherent.

Method	mAP (%)
Two-stream pre-trained [33]	20.09
fc6	20.08
fc7	21.24
Our model (no temporal)	21.92
Our model (no future)	21.30
Our model (no hard neg.)	24.22
Our model	**25.07**

Table 1. Event retrieval results on the MED11 event kits.

5.1. Event retrieval

In event retrieval, we are given a query video from the MED11 event kit and our goal is to retrieve videos that contain the same event from the remaining videos in the event kit. For each video in the event kit, we sort all other videos in the dataset based on their similarity to the query video using the cosine similarity metric, which we found to work best for all representations. We use Average Precision (AP) to measure the retrieval performance of each video and provide the mean Average Precision (mAP) over all videos in Tab. 1. For all methods, we uniformly sample 4 frames per video and represent the video as an average of the features extracted from them. The chance mAP is 6.53%. The different baselines used for comparison are explained below:

- *Two-stream pre-trained*: We use the two-stream CNN from [33] pre-trained on the UCF-101 dataset. The models were used to extract spatial and temporal features from the video with a temporal stack size of 5.
- *fc6* and *fc7*: Features extracted from the ReLU layers following the corresponding fully connected layers of a standard CNN model [19] pre-trained on ImageNet.
- *Our model (no temporal)*: Our model trained with no temporal context (Fig. 2(c)).
- *Our model (no future)*: Our model trained with no future context (Fig. 2(b)) but with multi-resolution sampling and hard negatives.
- *Our model (no hard neg.)*: Our model trained without hard negatives from the same video.
- *Our model*: Our full model trained with multi-resolution sampling and hard negatives.

We observe that our full model outperforms other representations for event retrieval. We note that in contrast to most other representations trained on ImageNet, our model is capable of being trained with large quantities of unlabeled video which is easy to obtain. This confirms our hypothesis that learning from unlabeled video data can improve feature representations. While the two-stream model also has the advantage of being trained specifically on a video dataset, we observe that the learned representations do not transfer

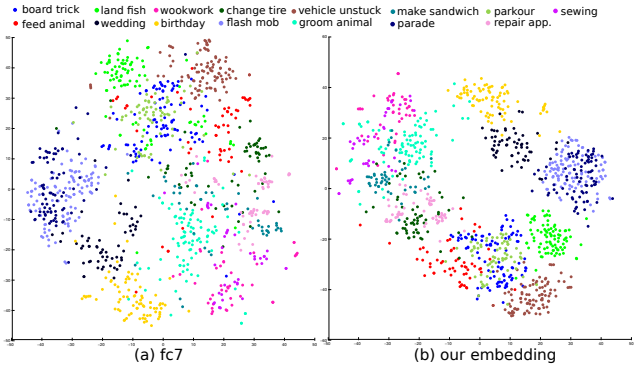

Figure 4. t-SNE plot of the semantic space for (a) fc7 and (b) our embedding. The different colors correspond to different events.

favorably to the MED11 dataset in contrast to fc7 and fc6 features trained on ImageNet. A similar observation was made in [41, 44], where simple CNN features trained from ImageNet consistently provided the best results.

Our embeddings capture the temporal regularities and patterns in videos without the need for expensive labels, which allows us to more effectively represent the semantic space of events. The performance gain of our full context model over the representation without temporal order shows the need for utilizing the temporal information while learning the embeddings. For the same temporal window size, the model without future uses smaller context. This potentially leads to lower visual variety in the context window, leading to a performance drop.

Visualizing the embedding space. To gain a better qualitative understanding of our learned embedding space, we use t-SNE [38] to visualize the embeddings in a 2D space. In Fig. 4, we visualize the fc7 features and our embedded features by sampling a random set of videos from the event kits. The different colors in the graph correspond to each of the 15 different event classes, as listed in the figure. Visually, we can see that certain event classes such as "Grooming an animal", "Changing a vehicle tire", and "Making a sandwich" enjoy better clustering in our embedded framework as opposed to the fc7 representation.

Another way to visualize this space is in terms of the actual words. Each video in the MED11 event kits is associated with a short synopsis describing the video. We represent each word from this synopsis collection by averaging the embeddings of videos associated with that word. The features are then used to produce a t-SNE plot as shown in Fig. 5. We avoid noisy clustering due to simple co-occurrence of words by only plotting words which do not frequently co-occur in the same synopsis. We observe many interesting patterns. For instance, objects such as "river", "pond" and "ocean" which provide the same context for a "fishing" event are clustered together. Similarly crowded settings such as "bollywood", "military", and "carnival" are clustered together.

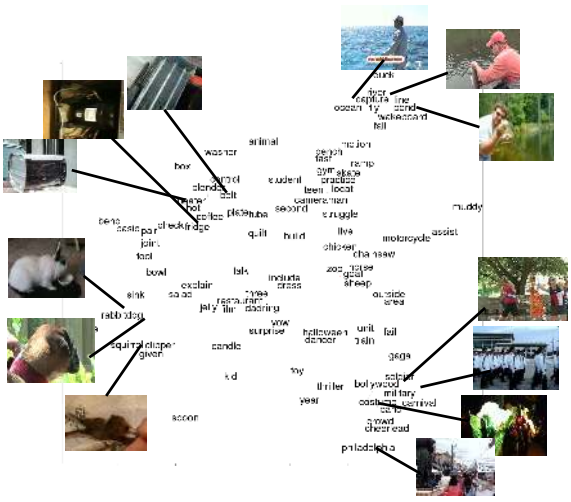

Figure 5. t-SNE visualization of words from synopses describing MED11 event kit videos. Each word is represented by the average of our embeddings corresponding to the videos associated with the word. We show sample video frames for a subset of the words.

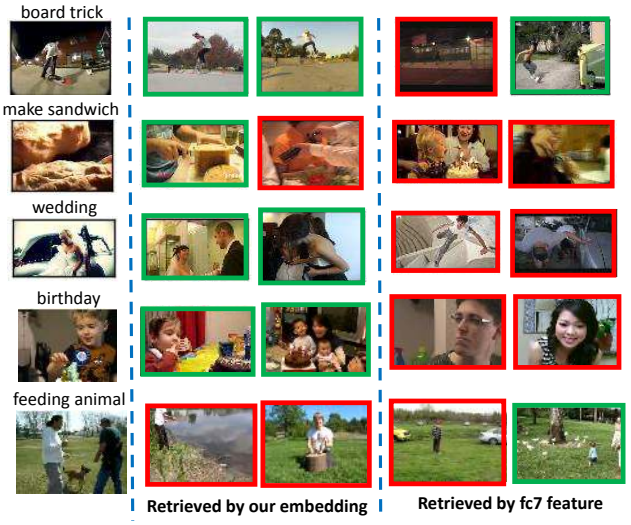

Figure 6. The retrieval results for fc7 (last two columns) and our embedding (middle two columns). The first column shows the query frame and event, while the top 2 frames retrieved from the remaining videos are shown in the middle two column for our embedding, and the last two columns for fc7. The incorrect frames are highlighted in red, and correct frames in green.

Event retrieval examples. We visualize the top frames retrieved for a few query frames from the event kit videos in Fig. 6. We observe a few interesting examples where the query appears visually distinct from our retrieved results. The retrieved actions might co-occur in the same context as the query, which is captured by the temporal context in our model. For instance, the frame of a "bride near a car" retrieves frames of "couple kissing". Similarly, the frame of "kneading dough" retrieves frames of "spreading butter".

Method	mAP (%)
Two-stream pre-trained [33]	20.11
fc6	19.27
fc7	22.99
Our model (no temporal)	22.50
Our model (no future)	21.71
Our model (no hard neg.)	24.12
Our model	**26.74**

Table 2. Temporal retrieval results on the MED11 event kits.

5.2. Temporal retrieval

In the temporal retrieval task, we test the ability of our embedding to capture the temporal structure in videos. We sample four frames from different time instants in a video and try to retrieve the frames in between the middle two frames. This is an interesting task which has potential for commercial applications such as ad placements in video search engines. For instance, the context at any time instant in a video can be used to retrieve the most suited video ad from a pool of video ads, to blend into the original video.

For this experiment, we use a subset of 1396 videos from the MED11 event kits which are at least 90 seconds long. From each video, we uniformly sample 4 context frames, 3 positive frames from in between the middle two context frames, and 12 negative distractors from the remaining segments of the video. In addition to the 12 negative distractors from the same video, all frames from other videos are also treated as negative distractors. For each video, given the 4 context frames we evaluate our ability to retrieve the 3 positive frames from this large pool of distractors.

We retrieve frames based on their cosine similarity to the average of the features extracted from the context frames. We use mean Average Precision (mAP) and the same baselines as event retrieval. The results are shown in Tab. 2. Our embedding representation is seen to outperform the other representations. This shows their ability to capture long-term interactions between events at different time-instants.

Temporal retrieval examples. We visualize the top examples retrieved for a few temporal queries in Fig. 7. We can see just how difficult this task is, as often frames that seem to be viable options for temporal retrieval are not part of the ground truth. For instance, in the "sandwich" example, our embedding wrongly retrieves frames of human hands to keep up with the temporal flow of the video.

6. Complex Event Classification

The complex event classification task on the MED11 event kits is one of the more challenging classification tasks. We follow the protocol of [7, 30] and use the same train/test splits. Since the goal of our work is to evaluate the effectiveness of video frame representations, we use a simple linear Support Vector Machine classifier for all methods.

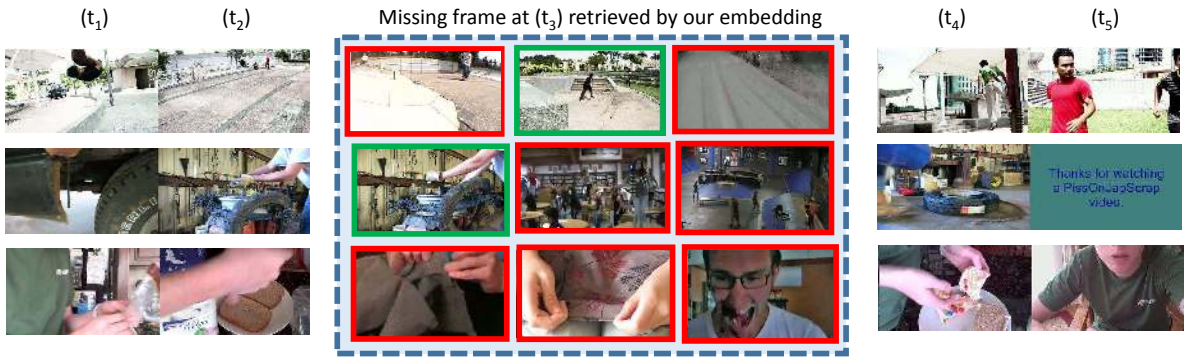

Figure 7. The retrieval results for our embedding model on the temporal retrieval task. The first and last 2 columns show the 4 context frames sampled from each video, and the middle 3 columns show the top 3 frames retrieved by our embedding. The correctly retrieved frames are highlighted in green, and incorrect frames highlighted in red.

Method	mAP (%)
Two-stream fine-tuned [33]	62.99
ISA [22]	55.87
Izadinia et al. [7] linear	62.63
Izadinia et al. [7] full	66.10
Raman. et al. [30]	66.39
fc6	68.56
fc7	69.17
Our model (no temporal)	69.57
Our model (no future)	69.22
Our model (no hard neg.)	69.81
Our model	**71.17**

Table 3. Event classification results on the MED11 event kits.

Unlike retrieval settings, we are provided labeled training instances in the event classification task. Thus, we fine-tune the last two layers of the two-stream model (pre-trained on UCF-101) on the training split of the event kits, and found this to perform better than the pre-trained model.

In addition to baselines from previous tasks, we also compare with [7], [22] and [30], with results shown in Tab. 3. Note that [7, 30] use a combination of multiple image and video features including SIFT, MFCC, ISA, and HOG3D. Further, they also use additional labels such as low-level events within each video. In Tab. 3, Izadinia et al. linear refers to the results without low-level event labels.

We observe that our method outperforms ISA [22], an unsupervised neural network feature. Additionally, the Imagenet pre-trained CNN features seem to perform better than most previous representations, which is also consistent with previous work [41, 44]. Our performance gain could be attributed to the use large amounts of unlabeled data to learn a better representations.

7. Temporal Order Recovery

An effective representation for video frames should be able to not only capture visual similarities, but also preserve the structure between temporally coherent frames. This facilitates holistic video understanding tasks beyond classification and retrieval. With this in mind, we explore the video temporal order recovery task, which seeks to show how the temporal interaction between different parts of a complex event are inherently captured by our embedding.

In this task, we are given as input a jumbled sequence of frames belonging to a video, and our goal is to order the frames into the correct sequence. This has been previously explored in the context of photostreams [17], and has potential for use in applications such as album generation.

Solving the order recovery problem. Since our goal is to

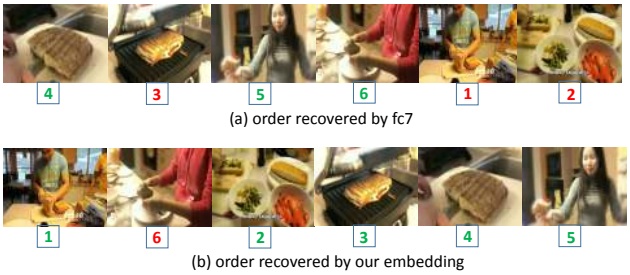

Figure 9. An example of the temporal ordering retrieved by fc7 and our method for a "Making a sandwich" video. The frame indexes already in the correct order are shown in green, and others in red.

Method	1.4k Videos	1k Videos
Random chance	50.00	50.00
Two-stream [33]	42.05	44.18
fc6	42.43	43.33
fc7	41.67	43.15
Our model (pairwise)	42.03	43.72
Our model (no future)	40.91	42.98
Our model (no hard neg.)	41.02	41.95
Our model	**40.41**	**41.13**

Table 4. Video temporal order recovery results evaluated using the Kendell tau distance (normalized to 0-100). Smaller distance indicates better performance. The 1.4k Videos refers to the set of videos used in the temporal retrieval task, and the 1k Videos refers to a further subset with the most visually dissimilar frames.

Figure 8. After querying the Internet for images of the "wedding" event, we cluster them into sub-events and temporally organize the clusters using our model. On the left, we show sample images crawled for the "wedding" event, and on the right the temporal order recovered by our model is visualized along with manual captions for the clusters.

evaluate the effectiveness of various feature representations for this task, we use a simple greedy technique to recover the temporal order. We assume that we are provided the first two frames in the video and proceed to retrieve the next frame (third frame) from all other frames in the video. This is done by averaging the first two frames and retrieving the closest frame in cosine similarity. We go on to greedily retrieve the fourth frame using the average of the second and third frames, and continue until all frames are retrieved. In order to enable easy comparison across all videos, we sample the same number of frames (12) from each video before scrambling them for the order recovery problem. An example comparing our embeddings to fc7 is show in Fig. 9.

Evaluation. We evaluate the performance for solving the order recovery problem using the Kendall tau [16] distance between the groundtruth sequence of frames and the sequence returned by the greedy method. The Kendall tau distance is a metric that counts the number of pairwise disagreements between two ranked lists; the larger the distance the more dissimilar the lists. The performance of different features for this task is shown in Tab. 4, where the Kendall tau distance is normalized to be in the range $0 - 100$.

Similar to the temporal retrieval setting, we use the subset of 1396 videos which are at least 90 seconds long. These results are reported in the first column of the table. We observed that our performance was quite comparable to that of fc7 features for videos with visually similar frames like those from the "parade" event, as they lack interesting temporal structure. Hence, we also report results on the subset of 1000 videos which had the most visually distinct frames. These results are shown in the second column of the table. We also evaluated the human performance of this task on a random subset of 100 videos, and found the Kendell tau to be around 42. This is on par with the performance of the automatic temporal order produced by our methods, and illustrates the difficulty of this task for humans as well.

We observe that our full context model trained with a temporal objective achieves the best Kendall tau distance. This improvement is more marked in the case of the 1k Videos with more visually distinct frames. This shows the ability of our model to bring together sequences of frames that should be temporally and semantically coherent.

Ordering actions on the Internet. Image search on the Internet has improved to the point where we can find relevant images with textual queries. Here, we wanted to investigate whether we could also temporally order images returned for complex event textual queries. As a toy example, we used query expansion on the "wedding" query, and crawled Google for a large set of images. We clustered the resulting images semantically, and for each cluster, averaged our embeddings to obtain a representation. We then used our method to recover the temporal ordering of these clusters of images. In Fig. 8, we show the recovered temporal ordering, and some example images from each cluster. Interestingly, the recovered order seems consistent with typical weddings.

8. Conclusion

In this paper, we presented a model to embed video frames. We treated videos as sequences of frames and embedded them in a way which captures the temporal context surrounding them. Our embeddings were learned from a large collection of more than 40000 unlabeled videos, and have shown to be more effective for multiple video tasks. The learned embeddings performed better than other video frame representations for all tasks. The main thrust of our work is to push a framework for learning frame-level representations from large sets of unlabeled video, which can then be used for a wide range of generic video tasks.

Acknowledgements

We thank A. Karpathy and S. Yeung for helpful comments. This research is partially supported by grants from ONR MURI and Intel ISTC-PC.

References

[1] N. Dalal et al. Human detection using oriented histograms of flow and appearance. In *ECCV*, 2006.

[2] J. Deng et al. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[3] J. L. Elman. Finding structure in time. *Cognitive science*, 1990.

[4] A. Frome et al. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013.

[5] R. Goroshin et al. Unsupervised feature learning from temporal data. *arXiv:1504.02518*, 2015.

[6] E. Grave, G. Obozinski, and F. Bach. A markovian approach to distributional semantics with application to semantic compositionality. In *Coling*, 2014.

[7] H. Izadinia and M. Shah. Recognizing complex events using large margin joint low-level event model. In *ECCV*, 2012.

[8] M. Jain, H. Jégou, and P. Bouthemy. Better exploiting motion for better action recognition. In *CVPR*, 2013.

[9] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *ICCV*, 2007.

[10] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *T-PAMI*, 2013.

[11] Y. Jia et al. Caffe: Convolutional architecture for fast feature embedding. *arXiv:1408.5093*, 2014.

[12] L. Jiang et al. Leveraging high-level and low-level features for multimedia event detection. In *ACM ICM*, 2012.

[13] Y.-G. Jiang et al. Trajectory-based modeling of human actions with motion reference points. In *ECCV*, 2012.

[14] A. Karpathy et al. Deep fragment embeddings for bidirectional image sentence mapping. In *NIPS*, 2014.

[15] A. Karpathy et al. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[16] M. G. Kendall. A new measure of rank correlation. *Biometrika*, pages 81–93, 1938.

[17] G. Kim and E. P. Xing. Jointly aligning and segmenting multiple web photo streams for the inference of collective photo storylines. In *CVPR*, 2013.

[18] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv:1411.2539*, 2014.

[19] A. Krizhevsky et al. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[20] K.-T. Lai et al. Recognizing complex events in videos by learning key static-dynamic evidences. In *ECCV*, 2014.

[21] I. Laptev et al. Learning realistic human actions from movies. In *CVPR*, 2008.

[22] Q. Le et al. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, 2011.

[23] T. Mikolov et al. Efficient estimation of word representations in vector space. *arXiv:1301.3781*, 2013.

[24] P. Natarajan, S. Wu, S. Vitaladevuni, X. Zhuang, S. Tsakalidis, U. Park, and R. Prasad. Multimodal feature fusion for robust event detection in web videos. In *CVPR*, 2012.

[25] J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010.

[26] S. Oh et al. Multimedia event detection with multimodal feature fusion and temporal concept localization. *Machine Vision and Applications*, 25, 2014.

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

[28] P. Over et al. An overview of the goals, tasks, data, evaluation mechanisms and metrics. In *TRECVID*, 2014.

[29] X. Peng, C. Zou, Y. Qiao, and Q. Peng. Action recognition with stacked fisher vectors. In *ECCV*, 2014.

[30] V. Ramanathan et al. Video event understanding using natural language descriptions. In *ICCV*, 2013.

[31] M. Ranzato et al. Video (language) modeling: a baseline for generative models of natural videos. *arXiv:1412.6604*, 2014.

[32] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*, 2012.

[33] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

[34] K. Soomro et al. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv:1212.0402*, 2012.

[35] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. *arXiv:1502.04681*, 2015.

[36] A. Tamrakar et al. Evaluation of low-level features and their combinations for complex event detection in open source videos. In *CVPR*, 2012.

[37] G. Taylor et al. Convolutional learning of spatiotemporal features. In *ECCV*, 2010.

[38] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *JMLR*, 9(2579-2605):85, 2008.

[39] H. Wang et al. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009.

[40] H. Wang et al. Action Recognition by Dense Trajectories. In *CVPR*, 2011.

[41] Z. Xu, Y. Yang, and A. G. Hauptmann. A discriminative cnn video representation for event detection. *arXiv:1411.4006v1*, 2015.

[42] Y. Yang and M. Shah. Complex events detection using data-driven concepts. In *ECCV*, 2012.

[43] W. Zaremba, I. Sutskever, and O. Vinyals. Recurrent neural network regularization. *arXiv:1409.2329*, 2014.

[44] S. Zha, F. Luisier, W. Andrews, N. Srivastava, and R. Salakhutdinov. Exploiting image-trained cnn architectures for unconstrained video classification. *arXiv:1503.04144v2*, 2015.