A Discriminative CNN Video Representation for Event Detection

Zhongwen Xu[†] Yi Yang[†] Alexander G. Hauptmann[§]
[†]QCIS, University of Technology, Sydney [§]SCS, Carnegie Mellon University
zhongwen.xu@student.uts.edu.au yee.i.yang@gmail.com alex@cs.cmu.edu

Abstract

In this paper, we propose a discriminative video representation for event detection over a large scale video dataset when only limited hardware resources are available. The focus of this paper is to effectively leverage deep Convolutional Neural Networks (CNNs) to advance event detection, where only frame level static descriptors can be extracted by the existing CNN toolkits. This paper makes two contributions to the inference of CNN video representation. First, while average pooling and max pooling have long been the standard approaches to aggregating frame level static features, we show that performance can be significantly improved by taking advantage of an appropriate encoding method. Second, we propose using a set of latent concept descriptors as the frame descriptor, which enriches visual information while keeping it computationally affordable. The integration of the two contributions results in a new state-of-the-art performance in event detection over the largest video datasets. Compared to improved Dense Trajectories, which has been recognized as the best video representation for event detection, our new representation improves the Mean Average Precision (mAP) from 27.6% to 36.8% for the TRECVID MEDTest 14 dataset and from 34.0% to 44.6% for the TRECVID MEDTest 13 dataset.

1. Introduction and Related Work

Complex event detection [1, 2], which targets the detection of such events as "renovating a home" in a large video collection crawled from Youtube, has recently attracted a lot of research attention in computer vision. Compared to concept analysis in videos, *e.g.*, action recognition, event detection is more difficult primarily because an event is more complex and thus has greater intra-class variations. For example, a "marriage proposal" event may take place indoors or outdoors, and may consist of multiple concepts such as ring (object), kneeling down (action) and kissing (action).

Recent research efforts have shown that combining multiple features, including static appearance features [9, 25, 41], motion features [23, 7, 43, 44, 33] and acoustic features [28], yields good performance in event detection, as evidenced by the reports of the top ranked teams in the TRECVID Multimedia Event Detection (MED) competition [3, 22, 29, 30] and research papers [26, 31, 40, 45] that have tackled this problem. By utilizing additional data to assist complex event detection, researchers propose the use of "video attributes" derived from other sources to facilitate event detection [27], or to utilize related exemplars when the training exemplars are very few [46]. As we focus on improving video representation in this paper, this new method can be readily fed into those frameworks to further improve their performance.

Dense Trajectories and its enhanced version *improved Dense Trajectories* (IDT) [44] have dominated complex event detection in recent years due to their superior performance over other features such as the motion feature STIP [23] and the static appearance feature Dense SIFT [3]. Despite good performance, heavy computation costs greatly restrict the usage of the improved Dense Trajectories on a large scale. In the TRECVID MED competition 2014 [2], the National Institute of Standards and Technology (NIST) introduced a very large video collection, containing 200,000 videos of 8,000 hours in duration. Paralleling 1,000 cores, it takes about one week to extract the improved Dense Trajectories for the 200,000 videos in the TRECVID MEDEval 14 collection. Even after the spatial re-sizing and temporal down-sampling processing, it still takes 500 cores one week to extract the features [3]. As a result of the unaffordable computation cost, it would be extremely difficult for a relatively smaller research group with limited computational resources to process large scale MED datasets. It becomes important to propose an efficient representation for complex event detection with only affordable computational resources, *e.g.*, *a single machine*, while at the same time attempting to achieve better performance.

One instinctive idea would be to utilize the deep learning approach, especially Convolutional Neural Networks (CNNs), given their overwhelming accuracy in image analysis and fast processing speed, which is achieved by leveraging the massive parallel processing power of GPUs [21]. However, it has been reported that the event detection

	MEDTest 13	MEDTest 14
IDT [44, 3]	**34.0**	**27.6**
CNN in Lan *et al*. [22]	29.0	N.A.
CNN_{avg}	32.7	24.8

Table 1. Performance comparison (mean Average Precision in percentage). Lan *et al*. [22] is the only attempt to apply CNN features in TRECVID MED 2013. CNN_{avg} are our results from the average pooling representation of frame level CNN descriptors.

performance of CNN based video representation is worse than the improved Dense Trajectories in TRECVID MED 2013 [22, 3], as shown in Table 1. A few technical problems remain unsolved.

Firstly, CNN requires a large amount of labeled video data to train good models from scratch. The large scale TRECVID MED datasets (*i.e*., MEDTest 13 [1] and MEDTest 14 [2]) only have 100 positive examples per event, with many null videos which are irrelevant. The number of labeled videos is smaller than that of the video collection for sports videos [20]. In addition, as indicated in [46], event videos are quite different from action videos, so it makes little sense to use the action dataset to train models for event detection.

Secondly, when dealing with a domain specific task with a small number of training data, fine-tuning [12] is an effective technique for adapting the ImageNet pre-trained models for new tasks. However, the video level event labels are rather coarse at the frame level, *i.e*., not all frames necessarily contain the semantic information of the event. If we use the coarse video level label for each frame, performance is barely improved by frame level fine-tuning; this was verified by our preliminary experiment[1].

Lastly, given the frame level CNN descriptors, we need to generate a discriminative video level representation. Average pooling is the standard approach [32, 3] for static local features, as well as for the CNN descriptors [22]. Table 1 shows the performance comparisons of the improved Dense Trajectories and CNN average pooling representation. We provide the performance of Lan *et al*. [22] for reference as well. We can see that the performance of CNN average pooling representation cannot get better than the hand-crafted feature improved Dense Trajectories, which is fairly different from the observations in other vision tasks [12, 6, 13].

The contributions of this paper are threefold. First, this is the first work to leverage the encoding techniques to generate video representation based on CNN descriptors. Second, we propose to use a set of latent concept descriptors as frame descriptors, which further diversifies the output with aggregation on multiple spatial locations at deeper stage of the network. The approach forwards video frames for only one round along the deep CNNs for descriptor extraction. With these two contributions, the proposed video CNN representation achieves more than 30% relative improvement over the state-of-the-art video representation on the large scale MED dataset, and this can be conducted on a single machine in two days with 4 GPU cards installed. In addition, we propose to use Product Quantization [15] based on CNN video representation to speed up the execution (event search) time. According to our extensive experiments, we show that the approach significantly reduces the I/O cost, thereby making event prediction much faster while retaining almost the same level of precision.

[1]However, with certain modification of the CNN structure, *e.g*. cross-frame max-pooling [11], fine-tuning could be helpful.

2. Preliminaries

Unless otherwise specified, this work is based on the network architecture released by [37], *i.e*., the configuration with 16 weight layers in the VGG ILSVRC 2014 classification task winning solutions. The first 13 weight layers are convolutional layers, five of which are followed by a max-pooling layer. The last three weight layers are fully-connected layers. In the rest of this paper, we follow the notations in [6, 12]: pool_5 refers to the activation of the last pooling layer, fc_6 and fc_7 refer to the activation of the first and second fully-connected layers, respectively. Though the structure in [37] is much deeper than the classic CNN structure in [21, 6, 12], the subscripts of pool_5, fc_6 and fc_7 notations still correspond if we regard the convolution layers between the max-pooling layers as a "compositional convolutional layer" [37]. We utilize the activations before Rectified Linear Units (*i.e*., fc_6 and fc_7) and after them (*i.e*., fc_6_relu and fc_7_relu), since we observe significant differences in performance between these two variants.

3. Video CNN Representation

We begin by extracting the frame level CNN descriptors using the Caffe toolkit [18] with the model shared by [37]. We then need to generate video level vector representations on top of the frame level CNN descriptors.

3.1. Average Pooling on CNN Descriptors

As described in state-of-the-art complex event detection systems [3, 32], the standard way to achieve image-based video representation in which local descriptor extraction relies on individual frames alone, is as follows: (1) Obtain the descriptors for individual frames; (2) Apply normalization on frame descriptors; (3) *Average pooling* on frame descriptors to obtain the video representation, *i.e*., $\boldsymbol{x}_{\text{video}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i$, $\boldsymbol{x}_i$ is the frame-level descriptor and N is the total number of frames extracted from the video; (4) Re-normalization on video representation.

Max pooling on frames to generate video representation

is an alternative method but it is not typical in event detection. We observe similar performance with average pooling, so we omit this method.

3.2. Video Pooling on CNN descriptors

Video pooling computes video representation over the whole video by pooling all the descriptors from all the frames in a video. The Fisher vector [35, 36] and Vector of Locally Aggregated Descriptors (VLAD) [16, 17] have been shown to have great advantages over Bag-of-Words (BoWs) [38] in local descriptor encoding methods. The Fisher vector and VLAD have been proposed for image classification and image retrieval to encode image local descriptors such as dense SIFT and Histogram of Oriented Gradients (HOG). Attempts have also been made to apply Fisher vector and VLAD on local motion descriptors such as Histogram of Optical Flow (HOF) and Motion Boundary Histogram (MBH) to capture the motion information in videos. To our knowledge, *this is the first work on the video pooling of CNN descriptors and we broaden the encoding methods from local descriptors to CNN descriptors in video analysis*.

3.2.1 Fisher Vector Encoding

In Fisher vector encoding [35, 36], a Gaussian Mixture Model (GMM) with K components can be denoted as $\Theta = \{(\mu_k, \Sigma_k, \pi_k), k = 1, 2, \ldots, K\}$, where μ_k, Σ_k, π_k are the mean, variance and prior parameters of k-th component learned from the training CNN descriptors in the frame level, respectively. Given $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ of CNN descriptors extracted from a video, we have mean and covariance deviation vectors for the k-th component as:

$$\boldsymbol{u}_k = \frac{1}{N\sqrt{\pi_k}} \sum_{i=1}^{N} q_{ki} \left(\frac{\boldsymbol{x}_i - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k} \right)$$
$$\boldsymbol{v}_k = \frac{1}{N\sqrt{2\pi_k}} \sum_{i=1}^{N} q_{ki} \left[\left(\frac{\boldsymbol{x}_i - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k} \right)^2 - \mathbf{1} \right], \quad (1)$$

where q_{ki} is the posterior probability. By concatenation of the $\boldsymbol{u}_k$ and $\boldsymbol{v}_k$ of all the K components, we form the Fisher vector for the video with size $2D'K$, where D' is the dimension of CNN descriptor $\boldsymbol{x}_i$ after PCA pre-processing. PCA pre-processing is necessary for a better fit on the diagonal covariance matrix assumption [36]. Power normalization, often Signed Square Root (SSR) with $z = \text{sign}(z)\sqrt{|z|}$, and ℓ_2 normalization are then applied to the Fisher vectors [35, 36].

3.2.2 VLAD Encoding

VLAD encoding [16, 17] can be regarded as a simplified version of Fisher vector encoding. With K coarse centers

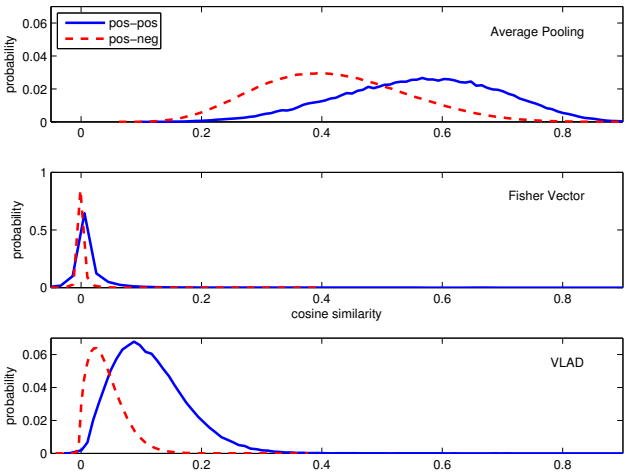

Figure 1. Probability distribution of the cosine similarity between positive-positive (blue and plain) and positive-negative (red and dashed) videos using fc_7 features, for average pooling (top), encoding with the Fisher vector using 256-component GMM (middle), and encoding with VLAD using 256 centers (bottom). As the range of probability of Fisher vectors is very different from average pooling and VLAD, we only use consistent axes for average pooling and VLAD. This figure is best viewed in color.

$\{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_K\}$ generated by K-means, we can obtain the difference vector regarding center c_k by:

$$\boldsymbol{u}_k = \sum_{i:\text{NN}(\boldsymbol{x}_i)=\boldsymbol{c}_k} (\boldsymbol{x_i} - \boldsymbol{c}_k), \quad (2)$$

where $\text{NN}(\boldsymbol{x_i})$ indicates $\boldsymbol{x}_i$'s nearest neighbors among K coarse centers.

The VLAD encoding vector with size $D'K$ is obtained by concatenating $\boldsymbol{u}_k$ over all the K centers. Another variant of VLAD called VLAD-k, which extends the nearest centers with the k-nearest centers, has shown good performance in action recognition [19, 34]. Without specification, we utilize VLAD-k with $k = 5$ by default. Except for the power and ℓ_2 normalization, we apply intra-normalization [4] to VLAD.

3.2.3 Quantitative Analysis

Given the above three approaches, we need to find out which one is the most appropriate for the CNN descriptors. To this end, we conduct an analytic experiment on the MEDTest 14 training set [2] to study the discriminative ability of three types of video representations, *i.e.*, average pooling, video pooling with Fisher vector, and video pooling with VLAD on the CNN descriptors. Specifically, we calculate the cosine similarity within the positive exemplars among all the events (denoted as pos-pos), and the cosine similarity between positive exemplars and negative exemplars (denoted as pos-neg). The results are shown in Fig-

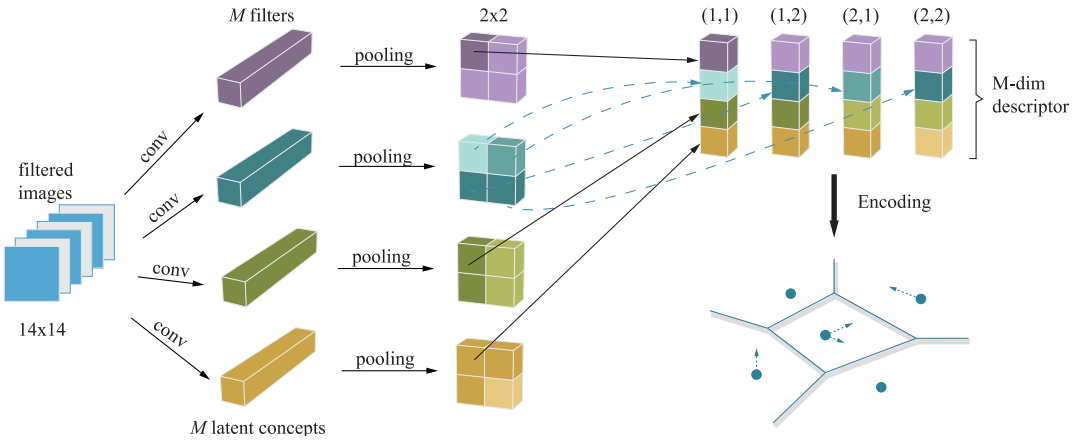

Figure 2. Illustration of the latent concept descriptors encoding procedure. We adopt M filters in the last convolutional layer as M latent concept classifiers. Before the last convolutional layer, M filters (*e.g.*, a cuboid of size $3 \times 3 \times 512$) produce the prediction outputs at every convolution location, followed by the max-pooling operations. Then, we get the responses of windows of different sizes and strides (in this example the output size is 2×2) for each latent concept. Color strength corresponds to the strength of response of each filter. Finally, we accumulate the responses for the M filters at the same location into the latent concept descriptors. Each dimension corresponds to one latent concept. After obtaining all latent concept descriptors of all frames, we then apply encoding methods to get the final video representation. This figure is best viewed in color.

ure 1. With a good representation, the data points of positive and negative exemplars should be far away from each other, *i.e.*, the cosine similarity of "pos-neg" should be close to zero. In addition, there should be a clear difference between the distributions of "pos-pos" and "pos-neg".

Average pooling: In Figure 1, we observe that the "pos-neg" cosine similarity distribution is far from zero, which is highly indicative that a large portion of the positive and negative exemplar pairs are similar to each other. In addition, the intersection of areas under the two lines span over a large range of $[0.2, 0.8]$. Both observations imply that average pooling may not be the best choice.

Fisher vector: Although the "pos-neg" similarity distribution is fairly close to zero, a large proportion of the "pos-pos" pairs also fall into the same range. No obvious difference between the distributions of "pos-pos" and "pos-neg" can be observed.

VLAD: The distribution of the "pos-neg" pairs is much closer to zero than average pooling while a relatively small proportion of the "pos-pos" similarity is close to the peak of the "pos-neg" similarity.

From the above analytic study, we can see that VLAD is the most fit for the CNN descriptors because the VLAD representation has the best discriminative ability, which is also consistent with the experimental results in Section 5.1.

3.3. CNN Latent Concept Descriptors

Compared to the fully-connected layers, pool_5 contains spatial information. However, if we follow the standard way and flatten pool_5 into a vector, the feature dimension will be very high, which will induce heavy computational cost. Specifically, the features dimension of pool_5 is $a \times a \times M$, where a is the size of filtered images of the last pooling layer and M is the number of convolutional filters in the last convolutional layer (in our case, $a = 7$ and $M = 512$). In the VGG network [37], pool_5 features are vectors of 25,088-D while the fc_6 and fc_7 features have only 4096-D. As a result, researchers tend to ignore the general features extracted from pool_5 [6, 13]. The problem is even more severe in the video pooling scheme because the frame descriptors with high dimensions would lead to instability problems [10].

Note that the convolutional filters can be regarded as generalized linear classifiers on the underlying data patches, and each convolutional filter corresponds to a latent concept [24]. We propose to formulate the general features from pool_5 as the vectors of *latent concept descriptors*, in which each dimension of the latent concept descriptors represents the response of the specific latent concept. Each filter in the last convolutional layer is independent from other filters. The response of the filter is the prediction of the linear classifier on the convolutional location for the corresponding latent concept. In that way, pool_5 layer of size $a \times a \times M$ can be converted into a^2 latent concept descriptors with M dimensions. Each latent concept descriptor represents the responses from the M filters for a specific pooling location. Once we obtain the latent concept descriptors for all the frames in a video, we then apply an encoding method to generate the video representation. In this case, each frame contains a^2 descriptors instead of one descriptor for the frame, as illustrated in Figure 2.

In [14], He *et al.* claim that the aggregation at a deeper

layer is more compatible with the hierarchical information processing in our brains than cropping or wrapping the original inputs, and they propose to use a Spatial Pyramid Pooling (SPP) layer for object classification and detection, which not only achieves better performance but also relaxes the constraint that the input must be fixed-size. Different from [14], we do not train the network with the SPP layer from scratch, because it takes much longer time, especially for a very deep neural network. Instead, at the last pooling layer, we adopt multiple windows with different sizes and strides without retraining the CNNs. In that way, visual information is enriched while only marginal computation cost is added, as we forward frames through the networks only once to extract the latent concept descriptors.

After extracting the CNN latent concept descriptors for all spatial locations of each frame in a video, we then apply video pooling to all the latent concept descriptors of that video. As in [14], we apply four different CNN max-pooling operations and obtain (6×6), (3×3), (2×2) and (1×1) outputs for each independent convolutional filter, a total of 50 spatial locations for a single frame. The dimension of latent concept descriptors (512-D) is shorter than the descriptors from the fully-connected layers (4,096-D), while the visual information is enriched via multiple spatial locations on the filtered images.

3.4. Representation Compression

For the engineering aspect of a fast event search [2] on a large video collection, we can utilize techniques such as Product Quantization (PQ) [15] to compress the Fisher vector or VLAD representation. With PQ compression, the storage space in disk and memory can be reduced by more than an order of magnitude, while the performance remains almost the same. The basic idea of PQ is to decompose the representation into sub-vectors with equal length B, and then within each sub-vector, K-means is applied to generate 2^m centers as representative points. All the sub-vectors are approximated by the nearest center and encoded into the index of the nearest center. In this way, B float numbers in the original representation become m bit code; thus, the compression ratio is $\frac{B \times 32}{m}$. For example, if we take $m = 8$ and $B = 4$, we can achieve 16 times reduction in storage space.

Targeting at prediction on compressed data instead of on the original features, we can decompose the learned linear classifier w with an equal length B. With look-up tables to store the dot-product between sub-vectors of 2^m centers and the corresponding sub-vector of w, the prediction speed on large-amount of videos can be accelerated by $\frac{D}{B}$ times look-up operations and $\frac{D}{B} - 1$ times addition operations for each video assuming D is the feature dimension [36] .

4. Experiment Settings

4.1. Datasets

In our experiments, we utilize the largest event detection datasets with labels[2], namely TRECVID MEDTest 13 [1] and TRECVID MEDTest 14 [2]. They have been introduced by NIST for all participants in the TRECVID competition and research community to conduct experiments on. For both datasets, there are 20 complex events respectively, but with 10 events overlapping. MEDTest 13 contains events E006-E015 and E021-E030, while MEDTest 14 has events E021-E040. Event names include "Birthday party", "Bike trick", *etc*. Refer to [1, 2] for the complete list of event names. In the training section, there are approximately 100 positive exemplars per event, and all events share negative exemplars with about 5,000 videos. The testing section has approximately 23,000 search videos. The total duration of videos in each collection is about 1,240 hours.

4.2. Features for Comparisons

As reported in [3] and compared with the features from other top performers [30, 29, 22] in the TRECVID MED 2013 competition, we can see that the improved Dense Trajectories has superb advantages over the original Dense Trajectories (used by all other teams except [3]), and is even better than approaches that combine many low-level visual features [30, 29, 22]. Improved Dense Trajectories extracts local descriptors such as trajectory, HOG, HOF, and MBH, and Fisher vector is then applied to encode the local descriptors into video representation. Following [44, 3], we first reduce the dimension of each descriptor by a factor of 2 and then utilize 256 components to generate the Fisher vectors. We evaluate four types of descriptor in improved Dense Trajectories, and report the results of the best combination of descriptors and the two individual descriptors that have the best performance (HOG and MBH).

In addition, we report the results of some popular features used in the TRECVID competition for reference, such as STIP [23], MoSIFT [7] and CSIFT [41], though their performance is far weaker than improved Dense Trajectories.

4.3. Evaluation Details

In all the experiments, we apply linear Support Vector Machine (SVM) with LIBSVM toolkit [5]. We conduct extensive experiments on two standard training conditions: in 100Ex, 100 positive exemplars are given in each event and in 10Ex, 10 positive exemplars are given. In the 100Ex condition, we utilize 5-fold cross-validation to choose the parameter of regularization coefficient C in linear SVM. In the 10Ex condition, we follow [22] and set C in linear SVM to 1.

[2]Labels for MEDEval 13 and MEDEval 14 are not publicly available.

We sample every five frames in the videos and follow the pre-processing of [21, 6] on CNN descriptor extraction. We extract the features from the center crop only. CNN descriptors are extracted using Caffe [18] with the best publicly available model [37], and we utilize vlfeat [42] to generate Fisher vector and VLAD representation.

Mean Average Precision (mAP) for binary classification is applied to evaluate the performance of event detection according to the NIST standard [1, 2].

5. Experiment Results

5.1. Results for Video Pooling of CNN descriptors

In this section, we show the experiments on video pooling of fc_6, fc_6_relu, fc_7 and fc_7_relu. Before aggregation, we first apply PCA with whitening on the ℓ_2 normalized CNN descriptors. Unlike local descriptors such as HOG, MBH, which have dimensions less than 200-D, the CNN descriptors have much higher dimensions (4,096-D). We conduct experiments with different reduced dimensions, *i.e.*, 128, 256, 512 and 1,024, and utilize the reduced dimensions that best balance performance and storage cost in corresponding features, *i.e.*, 512-D for fc_6 and fc_6_relu and 256-D for fc_7 and fc_7_relu. We utilize 256 components for Fisher vectors and 256 centers for VLAD as common choices in [36, 16]. We will study the impact of parameters in Section 5.3. PCA projections, components in GMM for Fisher vectors, and centers in K-means for VLAD are learned from approximately 256,000 sampled frames in the training set.

Since we observe similar patterns in MEDTest 13 and MEDTest 14 under both 100Ex and 10Ex, we take MEDTest 14 100Ex as an example to compare with different representations, namely average pooling, video pooling with Fisher vectors and video pooling with VLAD. From Table 2, we can see that both video pooling with Fisher vectors and VLAD demonstrate great advantages over the average pooling representation. On the video pooling of CNN descriptors, Fisher vector encoding does not exhibit better performance than VLAD. Similar observations have been expressed in [10]. We suspect that the distribution of CNN descriptors is quite different from the local descriptors, *e.g.*, HOG, HOF. We will study the theoretical reasons for the poorer performance of Fisher vector than VLAD on CNN video pooling in future research.

	fc_6	fc_6_relu	fc_7	fc_7_relu
Average pooling	19.8	24.8	18.8	23.8
Fisher vector	28.3	28.4	27.4	29.1
VLAD	**33.1**	32.6	**33.2**	31.5

Table 2. Performance comparison (mAP in percentage) on **MEDTest 14 100Ex**

We compare the performance of VLAD encoded CNN descriptors with state-of-the-art feature improved Dense Trajectories (IDT) and average pooling on CNN descriptors in Figure 3. We also illustrate the performance of the two strongest descriptors inside IDT (HOG and MBH). We can see very clearly that VLAD encoded CNN features significantly outperform IDT and average pooling on CNN descriptors over all settings. For more references, we provide the performance of a number of widely used features [29, 30, 22] on MEDTest 14 for comparison. MoSIFT [7] with Fisher vector achieves mAP 18.1% on 100Ex and 5.3% on 10Ex; STIP [23] with Fisher vector achieves mAP 15.0% on 100Ex and 7.1% on 10Ex; CSIFT [41] with Fisher vector achieves mAP 14.7% on 100Ex and 5.3% on 10Ex. Note that with VLAD encoded CNN descriptors, we can achieve better performance with 10Ex than the relatively poorer features such as MoSIFT, STIP, and CSIFT with 100Ex!

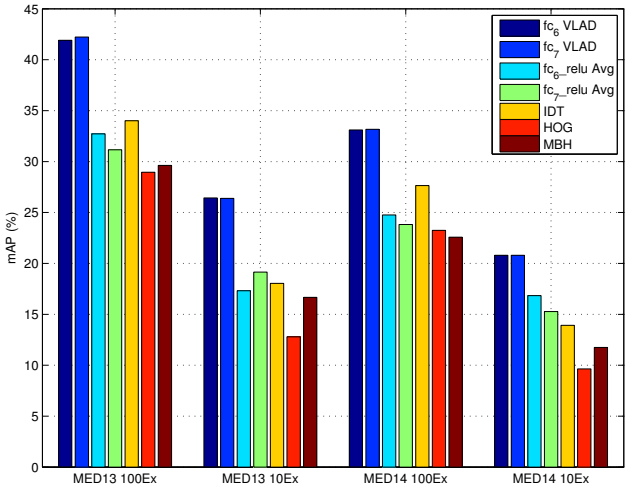

Figure 3. Performance comparisons on MEDTest 13 and MEDTest 14, both 100Ex and 10Ex. This figure is best viewed in color.

5.2. Results for CNN Latent Concept Descriptors with Spatial Pyramid Pooling

We evaluate the performance of *latent concept descriptors* (LCD) of both the original CNN structure and the structure with the Spatial Pyramid Pooling (SPP) layer plugged in to validate the effectiveness of SPP. Before encoding the latent concept descriptors, we first apply PCA with whitening. Dimension reduction is conducted from 512-D to a range of dimensions such as 32-D, 64-D, 128-D, and 256-D, and we find that 256-D is the best choice. We observe a similar pattern with video pooling of fc layers indicating that Fisher vector is inferior to VLAD on video pooling. We omit the results for Fisher vector due to limited space.

We show the performance of our proposed latent concept descriptors (LCD) in Table 3 and Table 4. In both 100Ex and 10Ex over two datasets, we can see clear gaps

	100Ex	10Ex
Average pooling	31.2	18.8
LCD_{VLAD}	38.2	25.0
LCD_{VLAD} + SPP	**40.3**	**25.6**

Table 3. Performance comparisons for pool_5 on **MEDTest 13**. LCD_{VLAD} is VLAD encoded LCD from the original CNN structure, while LCD_{VLAD} + SPP indicates VLAD encoded LCD with SPP layer plugged in.

	100Ex	10Ex
Average pooling	24.6	15.3
LCD_{VLAD}	33.9	22.8
LCD_{VLAD} + SPP	**35.7**	**23.2**

Table 4. Performance comparisons for pool_5 on **MEDTest 14**. Notations are the same as Table 3.

over the pool_5 features with average pooling, which demonstrates the advantages of our proposed novel utilization of pool_5. With SPP layer, VLAD encoded LCD (LCD_{VLAD} + SPP) continues to increase the performance further from the original structure (LCD_{VLAD}). The aggregation at a deeper stage to generate multiple levels of spatial information via multiple CNN max-pooling demonstrates advantages over the original CNN structure while having only minimal computation costs. The SPP layer enables a single pass of the forwarding in the network compared to the multiple passes of applying spatial pyramid on the original input images.

5.3. Analysis of the Impact of Parameters

We take VLAD encoded fc_7 features under MEDTest 14 100Ex as an example to see the impact of parameters in the video pooling process.

Dimensions of PCA: The original dimension of fc_7 is quite high compared to local descriptors. It is essential to investigate the impact of dimensions in PCA in the preprocessing stage, since it is critical to achieve a better trade-off of performance and storage costs. Table 5 shows that in dimensions of more than 256-D, performance remains similar, whereas encoding in 128-D damages the performance significantly.

Dimension	128-D	256-D	512-D	1024-D
mAP	30.6	**33.2**	33.1	33.2

Table 5. Impact of dimensions of CNN descriptors after PCA, with fixed $K = 256$ in VLAD.

Number of Centers in Encoding: We explore various numbers of centers K in VLAD, and the results are shown in Table 6. With the increase of K, we can see that the discriminative ability of the generated features improves. However when $K = 512$, the generated vector may be too sparse, which is somewhat detrimental to performance.

VLAD-k: We experiment with the traditional VLAD as well, with nearest center only instead of k-nearest centers.

K	32	64	128	256	512
mAP	28.7	29.7	30.4	**33.2**	32.1

Table 6. Impact on numbers of centers (K) in VLAD, with fixed PCA dimension of 256-D.

mAP drops from 33.2% to 32.0%.

Power Normalization: We remove the SSR post-processing and test the features on the VLAD encoded fc_7. mAP drops from 33.2% to 27.0%, from which we can see the significant effect of SSR post-processing.

Intra-normalization: We turn off the intra-normalization. mAP drops from 33.2% to 30.6%.

5.4. Results for Product Quantization Compression

	original	$B = 4$	$B = 8$
mAP	33.2	33.5 ($\uparrow$ 0.3)	33.0 ($\downarrow$ 0.2)
space reduction	-	16$\times$	32$\times$

Table 7. Performance change analysis for VLAD encoded fc_7 with PQ compression. B is the length of the sub-vectors in PQ and $m = 8$.

We conduct experiments on VLAD encoded fc_7 to see the performance changes with Product Quantization (PQ) compression. From the results in Table 7, we can see that PQ with $B = 4$ maintains the performance and even improves slightly. When $B = 8$, performance drops slightly. If we compress with $B = 4$, we can store VLAD encoded fc_7 features in 3.1 GB for the MEDEval 14, which contains 200,000 videos of 8,000 hours' duration. With further compression with a lossless technique such as Blosc[3] [8], we can store the features of the whole collection in less than 1 GB, which can be read by a normal SSD disk in a few seconds. Without PQ compression, the storage size of the features would be 48.8 GB, which severely compromises the execution time due to the I/O cost. Utilization of compression techniques largely saves the I/O cost in the prediction procedure, while preserving the performance.

In our speed test on the MEDEval 14 collection using the compressed data but not the original features, we can finish the prediction on 200,000 videos in 4.1 seconds per event using 20 threads on an Intel Xeon E5-2690v2 @ 3.00 GHz.

5.5. Results for Fusing Multiple Layers Extracted from the Same Model

We investigate average late fusion [39] to fuse the prediction results from different layers with PQ compression, *i.e.*, VLAD encoded LCD with SPP, fc_6 and fc_7. From Table 8 we can see that the simple fusion pushes the performance further beyond the single layers on MEDTest 13 and MEDTest 14, and achieves significant advantages over improved Dense Trajectories (IDT). Our proposed

[3] Blosc can reduce the storage space by a factor of 4

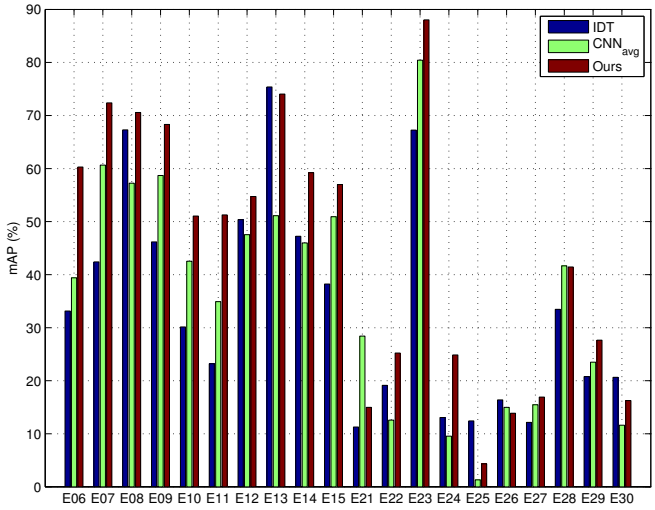

Figure 4. MEDTest 13 100Ex per event performance comparison (in mAP percentage). This figure is best viewed in color.

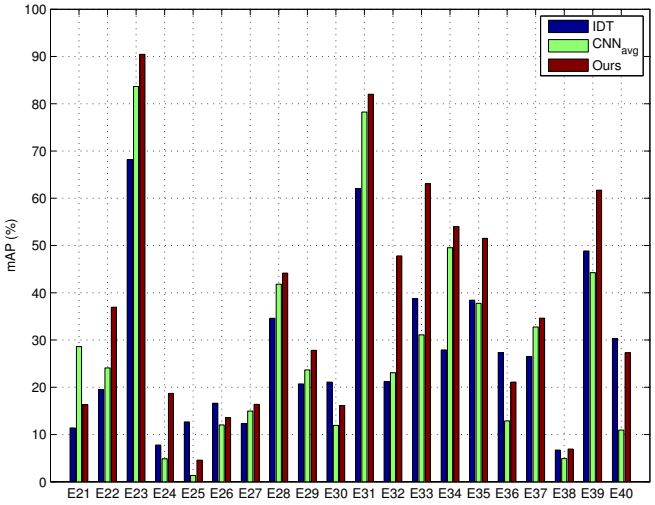

Figure 5. MEDTest 14 100Ex per event performance comparison (in mAP percentage). This figure is best viewed in color.

method pushes the state-of-the-art performance much further, achieves *more than 30% relative improvement* on 100Ex, and *more than 65% relative improvement* on 10Ex over both challenging datasets.

	Ours	IDT	Relative Improv
MED13 100Ex	**44.6**	34.0	31.2%
MED13 10Ex	**29.8**	18.0	65.6%
MED14 100Ex	**36.8**	27.6	33.3%
MED14 10Ex	**24.5**	13.9	76.3%

Table 8. Performance comparison of all settings; the last column shows the relative improvement of our proposed representation over IDT.

Figure 4 and Figure 5 show the per-event mAP comparison of the 100Ex setting on MEDTest 13 and MEDTest 14. We provide results for average pooling on CNN descriptors with late fusion of three layers as well, denoted as CNN$_{avg}$. Our proposed representation beats two other strong baselines in *15 out of 20 events* in MEDTest 13 and *14 out of 20 events* in MEDTest 14, respectively.

5.6. Comparison to the state-of-the-art Systems

We compare the MEDTest 13[4] results with the top performers in the TRECVID MED 2013 competition [3, 30, 22]. The AXES team does not show their performance on MEDTest 13 [3]. Natarajan *et al*. [30] report mAP 38.5% on 100Ex, 17.9% on 10Ex from their *whole visual system* of combining all their low-level visual features. Lan *et al*. [22] report 39.3% mAP on 100Ex of their *whole system including non-visual features* while they conducted 10Ex on their internal dataset. Our results achieve 44.6% mAP on 100Ex and 29.8% mAP on 10Ex, which significantly outperforms the top performers in the competition who combine more than 10 kinds of features with sophisticated schemes. To show that our representation is complementary to features from other modalities, we perform average late fusion of our proposed representation with IDT and MFCC, and generate a lightweight system with static, motion and acoustic features, which achieves **48.6% mAP on 100Ex, and 32.2% mAP on 10Ex**.

[4]In [3, 30, 22], teams report performance on MEDEval 13 as well, while MEDEval 13 is a different collection used in the competition, where only NIST can evaluate the performance.

6. Conclusion

TRECVID Multimedia Event Detection (MED) has suffered from huge computation costs in feature extraction and classification processes. Using Convolutional Neural Network (CNN) representation seems to be a good solution, but generating video representation from CNN descriptors has different characteristics from image representation. We are the first to leverage encoding techniques to generate video representation from CNN descriptors. And we propose latent concept descriptors to generate CNN descriptors more properly. For fast event search, we utilize Product Quantization to compress the video representation and predict on the compressed data. Extensive experiments on the two largest event detection collections under different training conditions demonstrate the advantages of our proposed representation. We have achieved promising performance which is superior to the state-of-the-art systems which combine 10 more features. The proposed representation is extendible and the performance can be further improved by better CNN models and/or appropriate fine-tuning techniques.

7. Acknowledgement

This paper is in part supported by the 973 program 2012CB316400, in part by the ARC DECRA project, and in part by Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20068. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPUs used for this research.

References

[1] TRECVID MED 13. http://www.nist.gov/itl/iad/mig/med13.cfm. 1, 2, 5, 6

[2] TRECVID MED 14. http://www.nist.gov/itl/iad/mig/med14.cfm. 1, 2, 3, 5, 6

[3] R. Aly, R. Arandjelovic, K. Chatfield, M. Douze, B. Fernando, Z. Harchaoui, K. McGuinness, N. E. O'Connor, D. Oneata, O. M. Parkhi, et al. The AXES submissions at TrecVid 2013. 2013. 1, 2, 5, 8

[4] R. Arandjelović and A. Zisserman. All about VLAD. In *CVPR*, 2013. 3

[5] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011. 5

[6] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014. 2, 4, 6

[7] M.-Y. Chen and A. Hauptmann. Mosift: Recognizing human actions in surveillance videos. *CMU TR*, 2009. 1, 5, 6

[8] R. G. Cinbis, J. Verbeek, and C. Schmid. Segmentation driven object detection with Fisher vectors. In *ICCV*, 2013. 7

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 1

[10] M. Douze, J. Revaud, C. Schmid, and H. Jégou. Stable hyper-pooling and query expansion for event detection. In *ICCV*, 2013. 4, 6

[11] C. Gan, N. Wang, Y. Yang, D.-Y. Yeung, and A. Hauptmann. Devnet: A deep event network for multimedia event detection and evience recounting. In *CVPR*, 2015. 2

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2

[13] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *ECCV*, 2014. 2, 4

[14] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*. 2014. 4, 5

[15] H. Jegou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *TPAMI*, 33(1):117–128, 2011. 2, 5

[16] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, 2010. 3, 6

[17] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *TPAMI*, 34(9):1704–1716, 2012. 3

[18] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding. *http://caffe. berkeleyvision. org*, 2013. 2, 6

[19] V. Kantorov and I. Laptev. Efficient feature extraction, encoding and classification for action recognition. In *CVPR*, 2014. 3

[20] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 2

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 2, 6

[22] Z.-Z. Lan, L. Jiang, S.-I. Yu, et al. CMU-Informedia at TRECVID 2013 Multimedia Event Detection. In *TRECVID 2013 Workshop*, 2013. 1, 2, 5, 6, 8

[23] I. Laptev. On space-time interest points. *IJCV*, 64(2-3):107–123, 2005. 1, 5, 6

[24] M. Lin, Q. Chen, and S. Yan. Network in network. *CoRR*, abs/1312.4400, 2013. 4

[25] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1

[26] Z. Ma, Y. Yang, N. Sebe, and A. Hauptmann. Knowledge adaptation with partially shared features for event detection using few exemplars. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(9):1789–1802, 2014. 1

[27] Z. Ma, Y. Yang, Z. Xu, S. Yan, N. Sebe, and A. G. Hauptmann. Complex event detection via multi-source video attributes. In *CVPR*, 2013. 1

[28] F. Metze, S. Rawat, and Y. Wang. Improved audio features for large-scale multimedia event detection. In *ICME*, 2014. 1

[29] G. K. Myers, R. Nallapati, J. van Hout, et al. The 2013 SESAME Multimedia Event Detection and Recounting system. In *TRECVID 2013 Workshop*, 2013. 1, 5, 6

[30] P. Natarajan, S. Wu, F. Luisier, et al. BBN VISER TRECVID 2013 Multimedia Event Detection and Multimedia Event Recounting Systems. In *TRECVID 2013 Workshop*, 2013. 1, 5, 6, 8

[31] 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. 1

[32] D. Oneata, M. Douze, J. Revaud, S. Jochen, D. Potapov, H. Wang, Z. Harchaoui, J. Verbeek, C. Schmid, R. Aly, et al. AXES at TRECVid 2012: KIS, INS, and MED. In *TRECVID workshop*, 2012. 2

[33] D. Oneata, J. Verbeek, and C. Schmid. Action and event recognition with Fisher vectors on a compact feature set. In *ICCV*, 2013. 1

[34] X. Peng, L. Wang, X. Wang, and Y. Qiao. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *arXiv preprint arXiv:1405.4506*, 2014. 3

[35] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*. 2010. 3

[36] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *IJCV*, 105(3):222–245, 2013. 3, 5, 6

[37] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2, 4, 6

[38] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *CVPR*, 2003. 3

[39] C. G. Snoek, M. Worring, and A. W. Smeulders. Early versus late fusion in semantic video analysis. In *MM*. ACM, 2005. 7

[40] A. Tamrakar, S. Ali, Q. Yu, J. Liu, O. Javed, A. Divakaran, H. Cheng, and H. Sawhney. Evaluation of low-level features and their combinations for complex event detection in open source videos. In *CVPR*, 2012. 1

[41] K. E. Van De Sande, T. Gevers, and C. G. Snoek. Evaluating color descriptors for object and scene recognition. *TPAMI*, 32(9):1582–1596, 2010. 1, 5, 6

[42] A. Vedaldi and B. Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. In *MM*. ACM, 2010. 6

[43] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, 2011. 1

[44] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013. 1, 2, 5

[45] Z. Xu, Y. Yang, I. Tsang, N. Sebe, and A. G. Hauptmann. Feature weighting via optimal thresholding for video analysis. In *ICCV*, 2013. 1

[46] Y. Yang, Z. Ma, Z. Xu, S. Yan, and A. G. Hauptmann. How related exemplars help complex event detection in web videos? In *ICCV*, 2013. 1, 2