VLAD^3: Encoding Dynamics of Deep Features for Action Recognition

Yingwei Li[†] Weixin Li[†] Vijay Mahadevan[§] Nuno Vasconcelos[†]
[†]University of California, San Diego
{yil325, wel107, nuno}@ucsd.edu
[§]Yahoo Research
vmahadev@yahoo-inc.com

Abstract

Previous approaches to action recognition with deep features tend to process video frames only within a small temporal region, and do not model long-range dynamic information explicitly. However, such information is important for the accurate recognition of actions, especially for the discrimination of complex activities that share sub-actions, and when dealing with untrimmed videos. Here, we propose a representation, ***VLAD for Deep Dynamics (VLAD^3)****, that accounts for different levels of video dynamics. It captures short-term dynamics with deep convolutional neural network features, relying on linear dynamic systems (LDS) to model medium-range dynamics. To account for long-range inhomogeneous dynamics, a VLAD descriptor is derived for the LDS and pooled over the whole video, to arrive at the final VLAD^3 representation. An extensive evaluation was performed on Olympic Sports, UCF101 and THUMOS15, where the use of the VLAD^3 representation leads to state-of-the-art results.*

1. Introduction

Object and action recognition are two important problems in computer vision. Over the last decade, the dominant representation for both problems was the bag-of-words model, combining histograms of descriptors such as HoG or HoF and pooling over space-time volumes [24, 32]. More recently, substantial gains in object recognition have been reported with the use of deep convolution neural networks (CNNs) [13, 27, 30]. This has motivated several attempts to apply CNNs to the action recognition problem [11, 26, 31]. However, for action, the gains over state-of-the-art bag-of-words approaches [32] have not been as impressive. This can be at least partially explained by the structure of the video signal.

While most images tend to exhibit localized spatial structure, the temporal structure of video is constrained by the laws of Newtonian mechanics, which determine the motion of objects in natural scenes. Since this motion tends to be smooth, the video dynamics are homogeneous over a substantial number of frames. We refer to this as the *medium-range structure* of video, which tends to be *homogeneous*. On the other hand, over the *long-range*, natural scene dynamics can be very *inhomogeneous* since videos frequently depict a number of actions. An example is given in Figure 1, where the action of interest ("javelin throw") is embedded in video that also contains actions that precede (athlete warm-ups) and follow (crowd shot, shot of the score board, *etc.*) it. This video is only marginally informative about the action of interest (depicts actions shared by all track events). Overall, the statistics of the video can be represented at various levels of granularity, giving rise to the hierarchy of Figure 1.

In the *short-term*, a video can be characterized as a sequence of frames with characteristic motion patterns. For example, the optical flow of the "pole-fly" scene is very distinct from that of the "score board" scene. We refer to this as *short-term video dynamics*. At the next level, the video can be grouped into shots of dynamics that are discriminative for the target action. Since these dynamics tend to span a substantial number of frames (at least seconds, sometimes minutes) they are denoted as *medium-range dynamics*. Note that, as shown in Figure 1, a single action, such as the "javelin throw," can span several shots, including the sub-actions "run," "throw," "pole flight," and "pole landing." Hence, the dynamics of a single action can frequently be decomposed into a sequence of states, which typically have themselves homogeneous mid-range dynamics. Finally, the target action is embedded into marginally or totally unrelated video. Hence, the *long-range dynamics* of video tend to be highly inhomogeneous. While chunks of these dynamics correspond to the action of interest, their location can vary from one video to the next.

Since the temporal structure of video statistics follows the hierarchy of Figure 1, it is sensible to consider a similar hierarchy for video models. The lowest level in this hierarchy includes models whose temporal support accounts for a few video frames. This includes approaches based on hand-crafted descriptors and a number of recent CNN models, which use a few frames as inputs and a purely feedforward structure [11, 26, 31]. This is denoted as the **short-term** level of the model hierarchy. The next class of models includes recurrent neural networks (RNNs) and their vari-

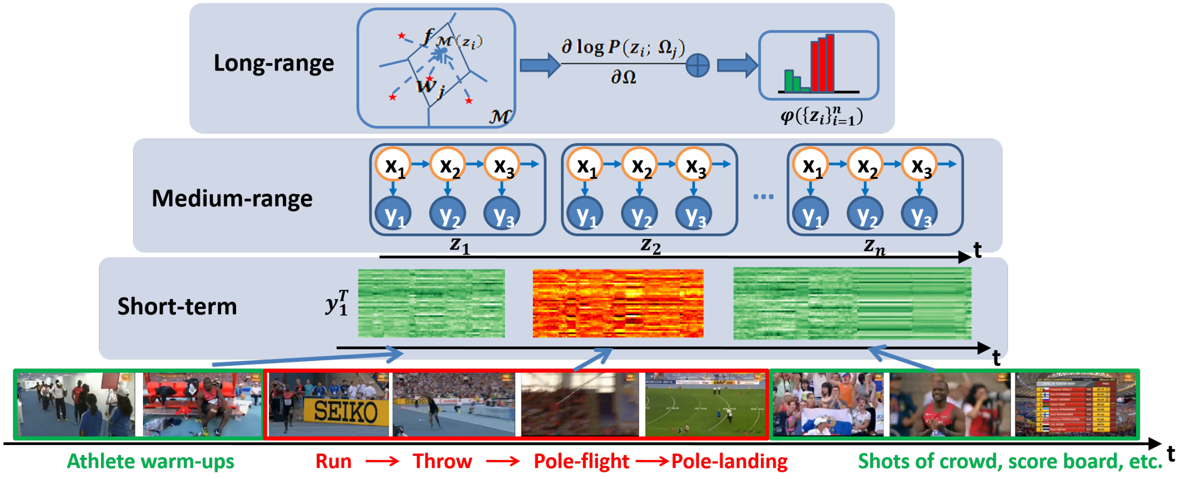

Long-range
$f_{\mathcal{M}}(z_i)$
w_j
$\mathcal{M}$
$\frac{\partial \log P(z_i;\,\Omega_j)}{\partial \Omega}$
$\varphi(\{z_i\}_{i=1}^n)$
Medium-range
x_1 x_2 x_3
y_1 y_2 y_3
z_1
z_2
...
z_n
t
Short-term
y_1^T
t
SEIKO
t
Athlete warm-ups
Run → Throw → Pole-flight → Pole-landing
Shots of crowd, score board, etc.

Figure 1: The VLAD^3 is inspired by the hierarchical structure of video dynamics. A **short-term** stage captures short-term appearance and motion patterns with deep features. A **medium-range** stage models the dynamics of segments of deep features, using an LDS. Finally, a **long-range** stage computes and pools a VLAD descriptor, derived from the LDS.

ants [6, 9], or a combination of a short-term representation, such as CNN activations, and a pooling operator along video trajectories [33]. These representations have some ability to capture the medium-range dynamics of Figure 1, but not completely. While they can model longer segments of homogeneous dynamics, it is less clear that they can account for actions composed of multiple distinct segments, such as the "javelin throw" of Figure 1. They can thus be considered as representations that fall between short and medium range. Beyond them, the **medium-range** level of the hierarchy includes models that explicitly account for a hidden state, such as the hidden Markov model (HMM) [29] or the linear dynamic system (LDS) [2], which allows them to model actions composed of multiple states corresponding to shots of sub-actions. Finally, the third-level of the hierarchy includes models that can account for short-term motion, medium range dynamics with multiple states and highly inhomogeneous long-range dynamics. We refer to this as the **long-range** level of the hierarchy.

In this work, we propose a representation for video that encompasses the three-level hierarchy of Figure 1. At the **short-term** level, this representation extracts features from a small temporal window over video frames, jointly capturing their appearance and motions. It consists of a deep CNN, whose layers implement spatiotemporal filters of reduced temporal support (16 frames). Semantically, this level of the representation accounts for action segments, e.g. "arm movement" or "running". At the **medium-range** level, the CNN features extracted by the **short-term** level are processed by a linear dynamic system (LDS). This accounts for medium-range dynamics by modeling the CNN feature as a sequence of observations from a stochastic process with a hidden state. By transitioning through states, it can account for non-stationary dynamics, e.g. that throwing a javelin consists of a temporal sequence of states such as "running," "throwing," "pole flight," and "pole landing". As the state process has a Gauss-Markovian structure, this model has much better scalability than recurrent networks and can be easily learned from much longer time sequences. In this work, LDSs can be learned from features spanning the whole video sequence. Finally, the last stage of the representation is motivated by observations that the VLAD descriptor [1] performs well for data with non-homogeneous statistics, e.g. image classification [1] or even prior work on action recognition [32, 21, 34]. We derive a VLAD descriptor for the LDS likelihood of CNN responses and use it as the final, **long-range** level representation of the video hierarchy. This representation is denoted as the *VLAD for Deep Dynamics (VLAD^3)*.

Overall, this work includes three main contributions. First, at the level of statistical modeling, we derive the VLAD descriptor for the LDS model. Second, at the level of video representation, we study the hypothesis that effective action representations must be discriminative in short-term, scalable enough to capture medium range dynamics, and robust to the heterogeneity of long-range dynamics. This leads to the proposed combination of short-term CNN features, medium-range LDS models, and global VLAD descriptor. Finally, we test the hypothesis through a large scale video recognition experiment which shows that the proposed representation achieves state-of-the-art performance on three challenging datasets - Olympic sports, UCF101, and THUMOS15 - which have been the subject of substantial prior research.

2. Related work

Long-range video dynamics are difficult to capture with CNN models with small temporal support, e.g. space-time models implemented by application of a spatial CNN to a small set of stacked video frames [11], combinations of CNNs operating on image and optical flow information [26], or even models that implement layers of 3D convolutional filters [31]. As these representations scale poorly in their temporal support, the resulting classifiers typically account for only 10 to 20 video frames. This is not sufficient to characterize the dynamics of the underlying scene, which can unfold over tens of seconds or even minutes. While these techniques are frequently mapped into a *holistic* video-level representation by application of a global pooling operation, this representation is a summary of the short-term statistics of the video sequence, not a model of its long-range dynamics.

The modeling of such dynamics can, in principle, be achieved with more sophisticated deep learning models. In fact, a branch of the deep learning literature has evolved to address this problem. This includes methods based on recurrent neural networks (RNNs) [6], or variants such as the long short-term memory (LSTM) of [9]. Such models have recently started to appear in the action recognition literature, e.g. by stacking an LSTM upon a CNN [4] so as to learn the high-level temporal structure of low-level visual features, or by using an LSTM to model the dynamics of CNN activations [19]. While models such as the RNN or LSTM can, in principle, model sequences of infinite length, they are usually trained with relatively small temporal support. While this observation holds even for datasets that are sizable, such as Sports1M [19], it is conceivable that the introduction of larger ones will eliminate the problem. This is unclear at this point. Similarly to the CNN approaches, these models have so far only been learned from sequences of 16 [4] to 30 [19] frames.

Alternatively, there has been interest in combining deep learning features with statistical representations of better temporal scalability. The hypothesis is that, while deep features are highly discriminative for *short-term dynamics*, action recognition will benefit from their combination with models of long-range dynamics. We refer to this as the *long-range dynamics* hypothesis. For example, [33] proposed to pool deep features along video trajectories, a procedure commonly used in the bag-of-words literature [32]. This can, in principle, exploit the temporal scalability of hand-crafted trajectories to substantially expand the temporal support of the video representation. However, because trajectories are obtained by tracking, they can be quite sensitive to the drift problem. In practice, this limits the maximum trajectory length, which is 15 frames in [33] and still below the temporal extent of the dynamics of most scenes.

While the long-range dynamics hypothesis has motivated the deployment of methods such as LSTM [4] or pooling of deep features along motion trajectories [33], the *modeling* of the long-range dynamics of deep learning features has so far received limited attention in the vision literature. In the broader context of action recognition, dynamics are frequently captured by rather straightforward operations, e.g. spatiotemporal extensions of spatial pyramid pooling [15], or latent support vector machines (SVM) that rely on anchor points to delimit motion segments [20]. Such approaches can only capture coarse dynamics.

Finer grained models usually rely on a more elaborate treatment of sequences of features, usually through generative models. Most generative models of video dynamics have been proposed for attribute-based representations [17]. These characterize the video in terms of elementary semantic units, such as "leg motion," "raised arm," etc. The emphasis on attribute dynamics can be partly explained by the discrete nature of these variables, which enables simple learning and inference with widely used statistical models. For example, [29] models attribute sequences with the combination of a Hidden Markov Model (HMM) and the associated Fisher vector. However, discrete dynamic models tend to underperform their continuous counterparts [2]. For example, by connecting state and observations through a PCA-like transformation, the LDS extracts a much more compact representation of the scene dynamics. This usually guarantees better generalization than what is possible with an HMM. The difficulty is that, due to the non-Euclidean nature of attribute spaces, the LDS is not a suitable model for attribute data.

This has motivated non-Euclidean extensions of the LDS. For example, [16] introduced the binary dynamic system (BDS), which is basically an LDS for discrete observations. However, such non-Euclidean variants introduce a non-trivial difficulty. While the LDS has exact inference, through the efficient Kalman smoothing computations [25], this is only possible because the state and observation distributions are both Gaussian, forming a conjugate pair. This conjugacy is broken for non-Euclidean observations, rendering exact inference impossible. This, in turn, makes it impossible to compute a VLAD vector for models such as BDS without resorting to complex Markov Chain Monte Carlo procedures or some form of approximate inference.

The proposed VLAD^3 representation is related to all these previous efforts. Similarly to [29], we propose a generative model of dynamics and the associated Fisher vector (albeit we use the simpler VLAD [21]). However, rather than the discrete HMM we rely on a continuous model of dynamics. While conceptually this makes the proposed approach more similar to the BDS, we eliminate the difficulties of this approach by eliminating its reliance on human defined attributes. Instead, we propose to apply the dynamic model (LDS) directly to CNN features, which are powerful in discriminating short-term dynamics. Overall, we rely on the CNN features for discrimination and on the LDS to capture the medium-range dynamics of these features. Besides eliminating the need for (additional) attribute annotations and classifiers, this model supports exact inference via Kalman smoothing. We exploit this property to design an efficient algorithm for the computation of the VLAD descriptor, which is not available for the BDS.

3. LDS for deep feature dynamics

In this section, we briefly review the linear dynamic system (LDS) model and its extensions that are of interest for this work.

3.1. Linear Dynamic System

The LDS is composed by a hidden Gauss-Markov state process and Gaussian observation process, according to

$$\left\{ \begin{array}{rcl} \boldsymbol{x}_{t+1} & = & A\boldsymbol{x}_t + \boldsymbol{v}_t, \qquad (1a) \\ \boldsymbol{y}_t & = & C\boldsymbol{x}_t + \boldsymbol{u} + \boldsymbol{w}_t, \qquad (1b) \end{array} \right.$$

where $\boldsymbol{y} \in \mathbb{R}^m$ is the observation and $\boldsymbol{x} \in \mathbb{R}^n$ the hidden state. This model is parametrized by the state transition matrix $A \in \mathbb{R}^{n\times n}$, the observation matrix $C \in \mathbb{R}^{m\times n}$, and the bias vector $\boldsymbol{u} \in \mathbb{R}^m$, and includes two noise components, the state noise $\boldsymbol{v}_t \sim \mathcal{N}(\mathbf{0}, Q)$ and the observation noise $\boldsymbol{w}_t \sim \mathcal{N}(\mathbf{0}, R)$. These are Gaussian processes of zero mean and covariance matrices Q and R, respectively. Finally, the initial state is $\boldsymbol{x}_1 = \boldsymbol{\mu} + \boldsymbol{v}_0 \sim \mathcal{N}(\boldsymbol{\mu}, S)$.

3.2. LDS learning

The LDS parameters $\Omega = \{C, A, \boldsymbol{u}, \boldsymbol{\mu}, R, Q, S\}$ can be learned by maximum likelihood (ML) using the expectation-maximization (EM) algorithm. An adaptation of the LDS, known as the dynamic texture model [5], is popular for video representation in computer vision. Dynamic texture (DT) learning [5] uses a much simpler approximation to estimate the LDS parameters, via a two-step algorithm. This starts by performing a principal component analysis (PCA) of the observation sequence $\{\boldsymbol{y}_t\}$. The resulting principal components are interpreted as the columns of the observation matrix C and the associated coefficients as an estimate of the hidden state variables $\{\boldsymbol{x}_t\}$. The second step then learns the transition matrix A and the noise parameters by solving a least squares problem.

3.3. LDS codebook learning

Besides learning a single LDS, several methods have been proposed to learn mixtures or codebooks of LDSs. Again, an ML solution can be obtained with recourse to the EM algorithm, leading to the mixture of dynamic textures model [3]. However, a simpler alternative is provided by the *bag-of-models clustering (BMC)* procedure of [16]. This procedure clusters observation sequences in model space, rather than in the observation space. Given a sample set of sequences $\mathcal{D} = \{\boldsymbol{z}_i\}_{i=1}^N$, where $\boldsymbol{z} = \{\boldsymbol{y}_t\}_{t=1}^\tau$ is an individual sequence, BMC iterates between a cluster assignment and a cluster refinement step.

The assignment step operates on the representation of each sequence $\boldsymbol{z}_i$ as a LDS model. The sequence is first subject to the mapping

$$f_\mathcal{M} : \mathcal{Z} \supseteq \{\boldsymbol{z}\} \mapsto M \in \mathcal{M} \qquad (2)$$

from the space of observation sequences $\mathcal{Z}$ to the model space $\mathcal{M}$. This mapping is implemented with an LDS learning algorithm, as discussed in Section 3.2. The resulting LDS, $f_\mathcal{M}(\boldsymbol{z}_i)$, is then assigned to one of the models in the LDS codebook, according to

$$q_i = \arg\min_j D_\mathcal{M}(f_\mathcal{M}(\boldsymbol{z}_i), \boldsymbol{w}_j), \qquad (3)$$

where $\boldsymbol{w}_j \in \mathcal{M}$, the j^{th} model in the codebook, is an LDS $P(\boldsymbol{z}; \Omega_j)$ of parameters Ω_j and

$$D_\mathcal{M} : \mathcal{M} \times \mathcal{M} \mapsto \mathbb{R} \qquad (4)$$

is a distance metric on model space $\mathcal{M}$.

The refinement step updates the codebook models according to

$$\boldsymbol{w}_j = f_\mathcal{M}(\{\boldsymbol{z}_i : q_i = j\}). \qquad (5)$$

This consists of gathering all sequences $\boldsymbol{z}_i$ assigned to each model and relearning the model parameters with the LDS learning algorithm of Section 3.2.

In this work, we rely on the BMC procedure to learn LDS codebooks, using the dynamic texture procedure to implement the mapping of (2) and the popular *Martin distance* of [18] as the distance of (4).

4. VLAD encoding for deep dynamics

In this section, we review the VLAD descriptor and introduce the VLAD encoding for Deep Dynamics (VLAD^3).

4.1. VLAD review

The vector of linearly aggregated descriptors (VLAD) is a simplified version of the Fisher vector descriptor of [10]. A codebook $\mathcal{V} = \{\boldsymbol{w}_j\}_{j=1}^V$ of V generative models $\boldsymbol{w}_j = P(\boldsymbol{z}; \Omega_j)$ of parameters Ω_j is first learned with recourse to a clustering algorithm. In this work, this is the BMC procedure of Section 3.3. The VLAD is an efficient encoding of the first-order statistics of a sample $\mathcal{D} = \{\boldsymbol{z}_i\}_{i=1}^N$ with respect to this codebook. Each sample point $\boldsymbol{z}_i$ is first assigned to a subset of the codewords according to

$$q_{ij} = \begin{cases} 1, & \text{if } \boldsymbol{w}_j \text{ is in the k-nearest neighborhood set of } \boldsymbol{z}_i, \\ 0, & \text{otherwise} \end{cases}$$

where the *k-nearest neighborhood* of $\boldsymbol{w}_j$ is defined by the distance of (3), which is the Martin distance in this work.

The VLAD encoding $\phi(\mathcal{D})$ is the vector

$$\phi(\mathcal{D}) = [\phi_1(\mathcal{D}), \ldots, \phi_V(\mathcal{D})] \tag{6}$$

such that

$$\phi_j(\mathcal{D}) = \sum_{i=1}^{n} q_{ij} \left. \frac{\partial \log P(\boldsymbol{z}_i; \Omega)}{\partial \Omega} \right|_{\Omega=\Omega_j}, \tag{7}$$

where $\frac{\partial \log P(\boldsymbol{z}_i;\Omega)}{\partial \Omega}$ is the gradient of the log-likelihood $\log P(\boldsymbol{z}_i; \Omega)$ with respect to the parameters in Ω. In summary, $\phi(\mathcal{D})$ pools the log-likelihood gradients of the sample $\mathcal{D} = \{\boldsymbol{z}_i\}_{i=1}^N$ with association q_{ij}.

4.2. LDS log-likelihood gradients

In the context of this work, the observation $\boldsymbol{z}$ is a sequence $\boldsymbol{y}_1^\tau = \{\boldsymbol{y}_t\}_{t=1}^\tau$ of localized spatiotemporal features extracted from a video sequence of length τ with a deep CNN, and $P(\boldsymbol{z}; \Omega) = P(\boldsymbol{y}_1^\tau; \Omega)$ is the LDS of (1). Using the chain rule of probability and the Markovian property of the hidden LDS states, this has likelihood

$$\begin{aligned} P(\boldsymbol{y}_1^\tau) &= \int P(\boldsymbol{y}_1^\tau | \boldsymbol{x}_1^\tau) P(\boldsymbol{x}_1^\tau)\, d\boldsymbol{x}_1^\tau \qquad (8) \\ &= \int \prod_{t=1}^{\tau} P(\boldsymbol{y}_t|\boldsymbol{x}_t) \prod_{t=2}^{\tau} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) P(\boldsymbol{x}_1)\, d\boldsymbol{x}_1^\tau, \end{aligned}$$

where

$$\begin{aligned} P(\boldsymbol{y}_t|\boldsymbol{x}_t) &= \mathcal{G}(\boldsymbol{y}_t; C\boldsymbol{x}_t + \boldsymbol{u}, R), && (9) \\ P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) &= \mathcal{G}(\boldsymbol{x}_t; A\boldsymbol{x}_{t-1}, Q), && (10) \\ P(\boldsymbol{x}_1) &= \mathcal{G}(\boldsymbol{x}_1; \boldsymbol{\mu}, S), && (11) \end{aligned}$$

and $\mathcal{G}(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma)$ is a Gaussian distribution of mean $\boldsymbol{\mu}$ and covariance Σ. One of the important properties of the LDS is that all the terms of (8) are Gaussian. As is usual for the VLAD, we only consider the gradients with respect to parameters that affect the mean of these distributions, namely the vectors $\boldsymbol{\mu}, \boldsymbol{u}$, and matrices A, C. Consider the gradient with respect to C. Using

$$\frac{\partial \log P(\boldsymbol{y}_1^\tau)}{\partial C} = \frac{1}{P(\boldsymbol{y}_1^\tau)} \frac{\partial P(\boldsymbol{y}_1^\tau)}{\partial C} \tag{12}$$

and

$$\begin{aligned} \frac{\partial P(\boldsymbol{y}_1^\tau)}{\partial C} &= \int \left[\frac{\partial}{\partial C} \prod_{t=1}^{\tau} P(\boldsymbol{y}_t|\boldsymbol{x}_t) \right] P(\boldsymbol{x}_1^\tau)\, d\boldsymbol{x}_1^\tau \\ &= \int \left[\sum_{t=1}^{\tau} \frac{\partial P(\boldsymbol{y}_t|\boldsymbol{x}_t)}{\partial C} \prod_{j \neq t} P(\boldsymbol{y}_j|\boldsymbol{x}_j) \right] P(\boldsymbol{x}_1^\tau)\, d\boldsymbol{x}_1^\tau \\ &= \int \left[\sum_{t=1}^{\tau} \frac{1}{P(\boldsymbol{y}_t|\boldsymbol{x}_t)} \frac{\partial P(\boldsymbol{y}_t|\boldsymbol{x}_t)}{\partial C} \right] P(\boldsymbol{y}_1^\tau|\boldsymbol{x}_1^\tau) P(\boldsymbol{x}_1^\tau)\, d\boldsymbol{x}_1^\tau \end{aligned}$$

it follows that

$$\begin{aligned} \frac{\partial \log P(\boldsymbol{y}_1^\tau)}{\partial C} &= \sum_{t=1}^{\tau} \int \frac{\partial \log P(\boldsymbol{y}_t|\boldsymbol{x}_t)}{\partial C} P(\boldsymbol{x}_1^\tau|\boldsymbol{y}_1^\tau) d\boldsymbol{x}_1^\tau \\ &= \sum_{t=1}^{\tau} E_{\boldsymbol{x}_t|\boldsymbol{y}_1^\tau} \left[\frac{\partial \log P(\boldsymbol{y}_t|\boldsymbol{x}_t)}{\partial C} \right]. \qquad (13) \end{aligned}$$

Similarly, it can be shown that

$$\begin{aligned} \frac{\partial \log P(\boldsymbol{y}_1^\tau)}{\partial A} &= \sum_{t=2}^{\tau} E_{\boldsymbol{x}_{t-1}^t|\boldsymbol{y}_1^\tau} \left[\frac{\partial \log P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}{\partial A} \right], && (14) \\ \frac{\partial \log P(\boldsymbol{y}_1^\tau)}{\partial \boldsymbol{u}} &= \sum_{t=1}^{\tau} E_{\boldsymbol{x}_t|\boldsymbol{y}_1^\tau} \left[\frac{\partial \log P(\boldsymbol{y}_t|\boldsymbol{x}_t)}{\partial \boldsymbol{u}} \right], && (15) \\ \frac{\partial \log P(\boldsymbol{y}_1^\tau)}{\partial \boldsymbol{\mu}} &= E_{\boldsymbol{x}_1|\boldsymbol{y}_1^\tau} \left[\frac{\partial \log P(\boldsymbol{x}_1)}{\partial \boldsymbol{\mu}} \right]. && (16) \end{aligned}$$

Using (9)-(11) and the Gaussian log-likelihood derivatives

$$\begin{aligned} \frac{\partial \log \mathcal{G}(\boldsymbol{x}; P\boldsymbol{s} + \boldsymbol{b}, \Sigma)}{\partial \boldsymbol{b}} &= \Sigma^{-1}(\boldsymbol{x} - P\boldsymbol{s} - \boldsymbol{b}), && (17) \\ \frac{\partial \log \mathcal{G}(\boldsymbol{x}; P\boldsymbol{s} + \boldsymbol{b}, \Sigma)}{\partial P} &= \Sigma^{-1}(\boldsymbol{x} - P\boldsymbol{s} - \boldsymbol{b})\boldsymbol{s}^\intercal, && (18) \end{aligned}$$

it follows that

$$\begin{aligned} \frac{\partial \log P(\boldsymbol{y}_1^\tau)}{\partial C} &= R^{-1} \sum_{t=1}^{\tau} \left[(\boldsymbol{y}_t - \boldsymbol{u})\boldsymbol{\alpha}_t^\intercal - C\boldsymbol{\beta}_{t,t} \right], && (19) \\ \frac{\partial \log P(\boldsymbol{y}_1^\tau)}{\partial A} &= Q^{-1} \sum_{t=2}^{\tau} \left[\boldsymbol{\beta}_{t,t-1} - A\boldsymbol{\beta}_{t-1,t-1} \right], && (20) \\ \frac{\partial \log P(\boldsymbol{y}_1^\tau)}{\partial \boldsymbol{u}} &= R^{-1} \sum_{t=1}^{\tau} \left[(\boldsymbol{y}_t - \boldsymbol{u}) - C\boldsymbol{\alpha}_t \right], && (21) \\ \frac{\partial \log P(\boldsymbol{y}_1^\tau)}{\partial \boldsymbol{\mu}} &= S^{-1}(\boldsymbol{\alpha}_1 - \boldsymbol{\mu}) && (22) \end{aligned}$$

with

$$\boldsymbol{\alpha}_t = E[\boldsymbol{x}_t|\boldsymbol{y}_1^\tau], \qquad \boldsymbol{\beta}_{t_1,t_2} = E[\boldsymbol{x}_{t_1}\boldsymbol{x}_{t_2}^\intercal|\boldsymbol{y}_1^\tau]. \tag{23}$$

These expectations are part of the standard LDS inference, and can be computed efficiently with recourse to Kalman smoothing [25], which only requires a single forward and a single backward pass through the sequence $\boldsymbol{y}_1^\tau$.

4.3. VLAD^3 encoding

To derive the VLAD^3 encoding of a long sequence $\boldsymbol{y}_1^T$, the sequence is first decomposed into a set of overlapping segments $\{\boldsymbol{z}_i\}_{i=1}^N$, where $\boldsymbol{z}_i$ is a subsequence of $\boldsymbol{y}_1^T$ of length τ, and an LDS is learned per subsequence. Given a codebook $\mathcal{V} = \{\boldsymbol{w}_j\}_{j=1}^V$ of V LDSs, where $\boldsymbol{w}_j = P(\boldsymbol{z}; \Omega_j)$, the sample association matrix q_{ij} of Section 4.1 is computed, using (3) and the Martin distance to define the k-nearest neighborhood of each $\boldsymbol{z}_i$. The log-likelihood gradients with respect to model $\boldsymbol{w}_j$ are then computed by using $\boldsymbol{y}_1^\tau = \boldsymbol{z}_i$ and $C = C_j$, $A = A_j$, $\boldsymbol{u} = \boldsymbol{u}_j$, and $\boldsymbol{\mu} = \boldsymbol{\mu}_j$, in (19)-(23). The expectations of (23) are obtained with the Kalman smoothing filter. Finally, the gradient information is pooled across the video-subsequences extracted from $\boldsymbol{y}_1^T$, according to (6) and (7). Following [34], the resulting vector is also post-processed with intra-normalization, power-normalization, and l_2 normalization.

5. Experiments

In this section, we describe the datasets used and discuss several experiments conducted to evaluate the performance of VLAD^3.

5.1. Datasets

Three datasets were used in our experiments: UCF101, Olympic Sports, and THUMOS15.

UCF101 [28] is widely used for video classification. It consists of 13,320 videos of 101 human action classes, covering a broad set of activities such as sports, musical instruments, and human-object interaction. However, most of the activities (*e.g*. "Knitting", "Drumming") are short, simple, and repetitive. Some of these activities can be discriminated without any modeling of dynamics, *e.g*. because they occur against a very specific background scene, or due to the presence of certain objects. In terms of the hierarchy of Figure 1, this dataset is somewhere between the **short-term** and **medium-range** levels. We adopt the three train-test splits suggested in [28].

Olympic Sports [20] contains YouTube videos of 16 sport activities, for a total of 783 videos. Each video is trimmed to contain only the activity of interest, but this can be a complex activity, composed by a sequence of simpler sub-actions. These sub-actions can be shared by different activities. For example, "long jump" and "triple jump" have very similar action segments ("running," "jumping"), only differing in their temporal sequencing and length (a single long jump vs. three shorter hops). Hence, temporal dynamics are critical for discrimination on this dataset, which can be confidently considered a dataset at the **medium-range** level of the hierarchy of Figure 1. On Olympic, we adopt the train-test split of [20].

The task of the THUMOS challenge [8] is to recognize human action classes in open source videos. The validation set of THUMOS 2015 contains 2,104 untrimmed videos of 101 activity classes, which are the same as those of UCF101. However, in THUMOS15 each video includes one/multiple instances of one/multiple actions, in varying temporal locations. The irrelevant video segments can be considered as semantic noise and the video statistics can be highly non-homogeneous. This datasets is an example of the **long-range** level of Figure 1. Since groundtruth is not available for the test set of THUMOS15, we followed the 5-fold cross-validation train-test strategy of [35] on the validation set.

5.2. Experimental set-up

In all experiments, the short-term representation of the VLAD^3 was based on the C3D features of [31], extracted from a 16-frame video window. The temporal stride was set to 16 for THUMOS15 and 4 on the other datasets. C3D is a 3D-convolutional deep network learned from a large video dataset. We used the 4096-d *fc6* feature vector, which was l_2 normalized and dimensionality reduced into a 256-d vector by PCA. Given the sequence of CNN vectors extracted from the video sequence, we defined a temporal sliding window of length τ and stride three. The τ feature vectors within the window centered at time t composed the sequence $\boldsymbol{z}_t$. A set of such sequences, collected from a random subset of the training video sequences was used to learn an LDS codebook of size $V = 128$, as discussed in Section 3.3. Each LDS codeword had a hidden state of dimension $n \in \{6, 8, 10, 20, 24\}$. The k-nn parameter of VLAD encoding was set to 5. The maximum τ used in the experiments was 40, corresponding to a temporal support of 656 frames for THUMOS15 and 176 frames for the other datasets. We cross-validated n and τ for each action class.

5.3. The importance of modeling dynamics

A number of experiments were performed to evaluate the importance of dynamic modeling in action recognition. In these experiments we considered a family of representations that cover different levels of the hierarchy of Figure 1. They are all based on the C3D deep features, differing only in the modeling of video dynamics. The first, denoted T3, pools feature responses with the temporal pyramid (of scale 3) of [15]. Since this is a very crude representation of video dynamics, this model is representative of the **short-term** level of Figure 1. The second, denoted CTR [2], consists of the spectral signature of an LDS learned from the entire video sequence. It can capture long-range dynamics, but assumes that these are homogeneous. It can thus be seen as a representative of the **medium-range** level of Figure 1. Finally, we considered, two representatives of the **long-range** level, the HMM Fisher vector of [29] and the proposed VLAD^3. These are very similar, differing only in the use of discrete or continuous state variables.

In all cases, the classifier was a 1-vs-rest linear SVM with cross-validated parameter C. Performance was evaluated by the mean average precision (mAP) metric. For the UCF101 dataset, where performance is usually reported in terms of the accuracy (Acc), we also computed this metric. A common strategy to boost action recognition performance is the fusion of different models. Following [2, 26, 21], we also tried to late-fuse the representations above with a *holistic* representation obtained by average pooling the CNN fea-

	UCF101*		Olympic†	THUMOS15$^\diamondsuit$
method	mAP(%)	Acc(%)	mAP(%)	mAP(%)
T3 *	84.35	82.96	80.14	56.50
CTR †	84.28	81.59	80.89	61.52
HMM-FV $^\diamondsuit$	85.14	80.41	88.15	72.30
VLAD^3 $^\diamondsuit$	**89.31**	**84.08**	**90.78**	**76.84**
	+ holistic			
T3 *	89.11	84.04	86.53	72.30
CTR †	88.44	83.00	86.19	72.03
HMM-FV $^\diamondsuit$	89.56	84.29	88.41	75.49
VLAD^3 $^\diamondsuit$	**90.47**	**84.65**	**90.81**	**78.15**

Table 1: Activity recognition performance on UCF101, Olympic Sports, and THUMOS15.

tures across the video sequence. The results of all experiments are summarized in Table 1. The top half of the table shows the performance of the individual methods and the bottom half the results of their fusion with the holistic approach. The symbols on the table reflect the location of each method and dataset on the hierarchy of Figure 1. A $\star$ is used to identify **short-term** representations/video, a † for **medium-range** and a $\diamondsuit$ for **long-range**.

These results confirm the hypothesis that the modeling of continuous dynamics is beneficial for action recognition. As expected, the gains of this modeling depend on the class of dynamics present in each dataset. While, for example, there is no significant difference between T3 and CTR on UCF101, where most video only has short-term dynamics, the difference becomes much more significant on THUMOS15. On Olympic and THUMOS15, where dynamics are more important, the gains of the **long-range** models (HMM-FV and VLAD^3) are substantial. Among these models, the VLAD^3 has clearly better performance. This confirms the gains previously observed in [2] for continuous dynamic models. Overall, the proposed VLAD^3 representation achieves significant gains in all datasets. Compared to T3, these increase from 5% on the **short-term** level UCF101, to 10% in the **medium-range** level Olympic Sports, to 20% on the **long-range** THUMOS15.

The increase from 5% in UCF101 to 20% on THUMOS15 is particularly relevant, since the two datasets have exactly the same activity classes, differing only on the inhomogeneity of their dynamics. This supports the hypothesis that 1) there are benefits to modeling dynamics, and 2) the dynamic model must account for *both* the long-term nature of these dynamics and their inhomogeneity. More detailed evidence in support of this hypothesis is given in Figure 2, which shows a plot of the per-class average precision for Olympic Sports. Classes, such as *bowling* or *diving platform 10m*, that differ from the rest in terms of short-term motion patterns are perfectly discriminated by the holistic C3D representation. On the other hand, for classes with similar short-term motion patterns, e.g. *hammer-throw*, *high-jump*, *triple jump* and the remaining track activities, the dynamic modeling of the VLAD^3 enables very significant gains in classification performance.

Finally, while late-fusion decreases the gap between the methods, it does not change the conclusions above. On the most challenging datasets, the gains of VLAD^3 are significant even after late-fusion. In fact, in all datasets, the top performance achieved by late-fusion of any of the **short-term** and **medium-range** methods is at most equal (and usually inferior) to that of the vanilla VLAD^3.

5.4. Role of long-range dynamics and heterogeneity

In addition to establishing a video representation that covers all levels of the hierarchy of Figure 1, the VLAD^3 can be used as a tool to investigate the importance of each "step" on this hierarchy, i.e. from short-term to long-range dynamics and from homogeneous to noisy dynamics. In fact, this can be done by measuring the performance of the VLAD^3 as a function of the temporal support τ of the subsequences used to compute it. In the limit of $\tau = 1$, the VLAD^3 includes no modeling of medium or long-range dynamics, reducing to a standard Gaussian-VLAD. As τ increases, and the LDS accounts for mid-range dynamics, performance is expected to improve. Finally, for large τ, as these dynamics become highly inhomogeneous, performance is expected to degrade. In the limit, it is possible to fit a single LDS to the full video $\boldsymbol{y}_1^T$(which could have hundreds or thousands of frames, *e.g*. average T for THUMOS15 is 6620), rather than fitting various LDSs to subsequences of size τ. This is denoted the single-LDS descriptor and expected to be sensitive to inhomogeneous dynamics.

Figure 3 presents the results of this procedure on the three datasets. In all cases, performance increases with τ demonstrating the benefits of **medium-range** level modeling. However, performance starts to saturate for a value of τ that matches the temporal support of the dynamics in the video and decreases after that. This can be seen by the fact that, in all cases, the single-LDS descriptor underperforms the VLAD^3 of optimal τ. These experiments also confirm the categorization of the datasets in Table 1. On Olympic Sports and THUMOS15, whose video is composed of complex events and/or has inhomogeneous dynamics, the VLAD^3 of optimal τ clearly outperforms the single-LDS. However, on UCF101, which is a **short-term** dataset, the single-LDS is almost as good.

5.5. Comparison with the state-of-the-art

A comparison to state-of-the-art results in the literature can be difficult, since different methods use different components that are not always comparable. In fact, the best results are usually obtained by fusing different representations. For example, as noted in [31], hand-crafted features such as iDT [32] are complementary to the C3D feature - the iDT encodes low-level gradients by tracking op-

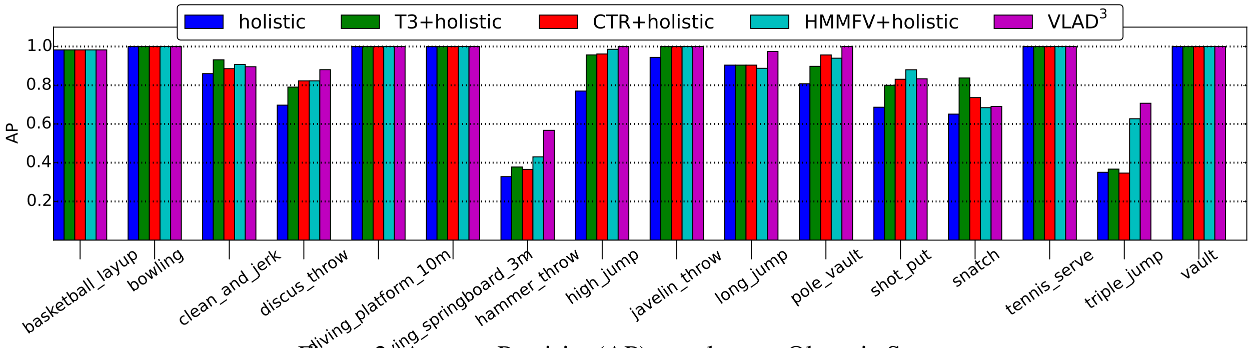
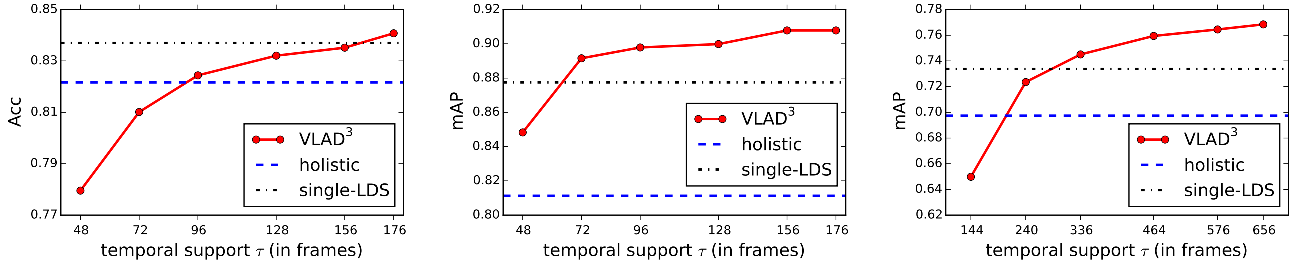


Figure 2: Average Precision(AP) per class on Olympic Sports.



(a) UCF101 (b) Olympic (c) THUMOS15

Figure 3: VLAD^3 performance as a function of the LDS temporal support τ.

UCF101 (Acc%)		Olympic Sports (mAP%)		THUMOS15(mAP%)	
Tran *et al.* [31]	90.4	Wang and Schmid [32]	91.1	Xu *et al.* [35]	74.6
Lan *et al.* [14]	89.1	Gaidon *et al.* [7]	85.5	Qiu *et al.* [23]	70.0
Simonyan and Zisserman[26]	88.0	Lan *et al.* [14]	91.4	Ning and Wu [12]	65.5
Wang *et al.* [33]	91.5	Peng *et al.* [22]	93.8	VLAD^3	**76.8**
VLAD^3 +iDT	90.9	VLAD^3 +iDT	**96.6**	VLAD^3 +iDT	**80.8**
VLAD^3 +iDT(fisher)	**92.2**				

Table 2: Comparison to state-of-the-art results.

tical flow while C3D features are more abstract and capture high level semantics. Inspired by this observation, we tested the late-fusion of the scores produced by VLAD^3 , which is built on top of C3D features, and iDT. Table 2 compares this approach to the state-of-the-art for the three datasets considered in this work. In all cases, the fusion of VLAD^3 and iDT achieves the best performance. For example, on UCF101, replacing the state-of-the-art trajectory-pooled deep descriptors (TDD+iDT (fisher)) of [33] by the VLAD^3+iDT (fisher) combination, improves the state-of-the-art from 91.5% to 92.2%. This is a non-trivial gain, given the amount of research that has addressed this dataset. Similarly, on Olympic Sports, the VLAD^3+iDT combination, outperforms approaches that attempt to stack multiple layers of feature representation. Since THUMOS15 is a relatively new dataset, we only list the results using the top three approaches (with only visual features) reported in the competition. Note that these approaches often require fusion of many different features, while VLAD^3 by itself can lead to better performance, clearly demonstrating the benefits of a representation that models long-range dynamics. Again, the VLAD^3+iDT achieves the state-of-the-art results by a big margin.

6. Conclusion

In this work, we proposed a new video representation, the VLAD^3, that models video dynamics at three hierarchical levels. The resulting encoding leverages discriminative deep features for short-term dynamics, the LDS model for medium-range dynamics, and a novel VLAD descriptor, derived from the LDS, for long-range dynamics. This enables it to model video whose dynamics are homogeneous over short and medium-range time scales, but inhomogeneous over long time spans. An evaluation on three datasets showed that explicit modeling of long-range dynamics is important for action recognition, and demonstrated superior performance of the proposed VLAD^3 compared to state-of-the-art methods.

7. Acknowledgments

This work was partially funded by NSF grant IIS-1208522. We also thank NVIDIA for GPU donation through their academic program.

References

[1] R. Arandjelovic and A. Zisserman. All about vlad. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1578–1585. IEEE, 2013. 2

[2] S. Bhattacharya, M. M. Kalayeh, R. Sukthankar, and M. Shah. Recognition of complex events: Exploiting temporal dynamics between underlying concepts. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2243–2250. IEEE, 2014. 2, 3, 6, 7

[3] A. B. Chan and N. Vasconcelos. Modeling, clustering, and segmenting video with mixtures of dynamic textures. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(5):909–926, 2008. 4

[4] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko. Long-term recurrent convolutional networks for visual recognition and description. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 2625–2634, June 2015. 3

[5] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109, 2003. 4

[6] J. L. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990. 2, 3

[7] A. Gaidon, Z. Harchaoui, and C. Schmid. Activity representation with motion hierarchies. *International journal of computer vision*, 107(3):219–238, 2014. 8

[8] A. Gorban, H. Idrees, Y.-G. Jiang, A. Roshan Zamir, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. 6

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2, 3

[10] T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing Systems*, pages 487–493, 1999. 4

[11] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1725–1732. IEEE, 2014. 1, 3

[12] F. W. Ke Ning. Zjudcd submission at thumos challenge 2015. In *CVPR workshop*, 2015. 8

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1

[14] Z. Lan, M. Lin, X. Li, A. G. Hauptmann, and B. Raj. Beyond gaussian pyramid: Multi-skip feature stacking for action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 204–212, June 2015. 8

[15] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition (CVPR), 2008 IEEE Conference on*, pages 1–8. IEEE, 2008. 3, 6

[16] W. Li, Q. Yu, H. Sawhney, and N. Vasconcelos. Recognizing activities via bag of words for attribute dynamics. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2587–2594. IEEE, 2013. 3, 4

[17] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3337–3344. IEEE, 2011. 3

[18] R. J. Martin. A metric for arma processes. *Signal Processing, IEEE Transactions on*, 48(4):1164–1170, 2000. 4

[19] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 4694–4702, June 2015. 3

[20] J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *Computer Vision–ECCV 2010*, pages 392–405. Springer, 2010. 3, 6

[21] 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. 2, 3, 6

[22] X. Peng, C. Zou, Y. Qiao, and Q. Peng. Action recognition with stacked fisher vectors. In *Computer Vision–ECCV 2014*, pages 581–595. Springer, 2014. 8

[23] Z. Qiu, Q. Li, T. Yao, T. Mei, and Y. Rui. Msr asia msm at thumos challenge 2015. In *CVPR workshop*, 2015. 8

[24] J. Sánchez and F. Perronnin. High-dimensional signature compression for large-scale image classification. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1665–1672. IEEE, 2011. 1

[25] R. H. Shumway and D. S. Stoffer. An approach to time series smoothing and forecasting using the em algorithm. *Journal of time series analysis*, 3(4):253–264, 1982. 3, 5

[26] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014. 1, 3, 6, 8

[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[28] K. Soomro, A. Roshan Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. In *CRCV-TR-12-01*, 2012. 6

[29] C. Sun and R. Nevatia. Active: Activity concept transitions in video event classification. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 913–920. IEEE, 2013. 2, 3, 6

[30] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 1–9, June 2015. 1

[31] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. C3d: generic features for video analysis. *arXiv preprint arXiv:1412.0767*, 2014. 1, 3, 6, 7, 8

[32] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3551–3558. IEEE, 2013. 1, 2, 3, 7, 8

[33] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 4305–4314, June 2015. 2, 3, 8

[34] Z. Xu, Y. Yang, and A. G. Hauptmann. A discriminative cnn video representation for event detection. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 1798–1807, June 2015. 2, 6

[35] Z. Xu, L. Zhu, Y. Yang, and A. G. Hauptmann. Uts-cmu at thumos 2015. In *CVPR workshop*, 2015. 6, 8