Unsupervised Universal Attribute Modeling for Action Recognition

Debaditya Roy, Kodukula Sri Rama Murty, and Chalavadi Krishna Mohan

***Abstract*—A fixed dimensional representation for action clips of varying lengths has been proposed in the literature using aggregation models like bag-of-words and Fisher vector. These representations are high dimensional and require classification techniques for action recognition. In this paper, we propose a framework for unsupervised extraction of a discriminative low-dimensional representation called action-vector. To start with, local spatio-temporal features are utilized to capture the action attributes implicitly in a large Gaussian mixture model called the universal attribute model (UAM). To enhance the contribution of the significant attributes in each action clip, a maximum *aposteriori* adaptation of the UAM means is performed for each clip. This results in a concatenated mean vector called super action vector (SAV) for each action clip. However, the SAV is still high dimensional because of the presence of redundant attributes. Hence, we employ factor analysis to represent every SAV only in terms of the few important attributes contributing to the action clip. This leads to a low-dimensional representation called action-vector. This entire procedure requires no class labels and produces action-vectors that are distinct representations of each action irrespective of the inter-actor variability encountered in unconstrained videos. An evaluation on trimmed action datasets UCF101 and HMDB51 demonstrates the efficacy of action-vectors for action classification over state-of-the-art techniques. Moreover, we also show that action-vectors can adequately represent untrimmed videos from the THUMOS14 dataset and produce classification results comparable to existing techniques.**

***Index Terms*—Action recognition, universal attribute model, unsupervised feature extraction, MAP adaptation, factor analysis, Gaussian mixture model.**

I. Introduction

HUMAN actions can be modeled in terms of a sequence of atomic attributes. For example, the act of *boxing* can be interpreted as a combination of attributes such as *right-hand forward punch and right-hand retraction*, followed by *a left-hand forward punch and left-hand retraction*. However, the definition of attributes is a subjective phenomenon and hence a manual annotation of attributes is highly inconsistent [38]. In addition, unconstrained videos have inter-actor variability or viewpoint differences which cause large deviations within the same attribute, making their explicit extraction difficult. Hence, we propose a framework for implicit attribute modelling using a universal attribute model (UAM). In order to represent an action, we should be able to determine the attributes that are responsible for that action. Since these attributes are implicitly modeled, we use factor analysis to discover them. Finally, we obtain a description containing only the contributions from the implicit attributes for that action. We refer to this representation as an action-vector that is both low-dimensional and distinct for different actions.

Recent literature regarding action representation focuses on long-term features which are extracted from the entire video clip. In [5], motion dynamics from a whole clip were captured using a long short term memory (LSTM) where each frame was represented as an output of a convolutional neural network (CNN). Another approach considered long frame sequences (60–100 frames) for capturing long-term temporal relationships for action description [25]. In [33], temporal segment networks were used to sample entire videos in order to produce a single feature vector representation. Although long-term features can summarize an entire video, it is computationally expensive to calculate such features for very long videos. Especially, in the case of temporally untrimmed videos which contain background movement, obtaining an adequate representation for the desired action is challenging.

There exists an extensive use of short-term features to represent short duration snippets in literature before the advent of long-term features. Improved dense trajectory [28] is the most popular amongst such features and it describes a set of points being tracked across several frames (generally considered to be 15). This description contains a) the histogram of oriented gradients (HOG) which defines the spatial structure of the neighbourhood of the point, b) the histogram of optical flow (HOF) descriptor that calculates the temporal derivative of the trajectory taken by the point, and c) the motion boundary histogram (MBH) which is the concatenation of horizontal and vertical spatial derivatives calculated on the temporal derivatives used for HOF. Another representation which has gained prominence is the 3D CNN features which are also extracted from 16 consecutive frames [24]. Both these features perform close to long-term features for action classification when combined with a suitable classifier like a support vector machine (SVM). However, the biggest challenge in using short-term features is

Manuscript received May 18, 2017; revised November 6, 2017, July 19, 2018, and October 4, 2018; accepted November 24, 2018. Date of publication December 14, 2018; date of current version June 21, 2019. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Zhu Li. *(Corresponding author: Debaditya Roy.)*

D. Roy and C. K. Mohan are with the Visual Learning and Intelligence Group, Department of Computer Science and Engineering, Indian Institute of Technology Hyderabad, Hyderabad 502205, India (e-mail: cs13p1001@iith.ac.in; ckm@iith.ac.in).

K. S. R. Murty is with the Department of Electrical Engineering, Indian Institute of Technology Hyderabad, Hyderabad 502205, India (e-mail: ksrm@iith.ac.in).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2018.2887021

that the number of features is dictated by the duration of the clip, leading to varying length patterns.

Attempts have been made to arrive at a fixed dimensional representation for each clip in order to overcome the varying number of short-term features extracted from each clip. In this regard, some methods consider an action as a sequence of features. Gaidon *et al.* [7] proposed that each action can be decomposed to be a sequence of atomic units called actoms. A histogram of visual features was extracted from each actom, and a sequence of these features was used to model the action. While the actoms were annotated manually during training, they were obtained automatically while testing. In another approach, actions were decomposed into actionlets and represented using Markov dependencies between these actionlets [15]. Although this method can capture long-term dependencies among actionlets, it nevertheless requires a manual identification of actionlets and explicit modelling. In [35], every action was divided into a set of events and the start of each event was associated with a latent variable. The action detection problem was then posed as a quadratic programming (QP) problem using these latent variables. However, this approach demands precise manual identification of start and end points of the events, and also requires fixed length representation of each event for formulating the QP. A similar problem is encountered in temporal clustering-based methods that rely mainly on change detection for identifying and clustering motion segments [37]. Moreover, the methods mentioned above do not model fluid actions like *blowing out candles* or *playing a flute* where marking the start and end of events within the action becomes challenging.

Apart from sequential modelling of features, there are many aggregation based frameworks like bag-of-words (BoW), Fisher vector and vector of locally aggregated descriptors (VLAD) which have been predominantly used for action representation [28]. To derive either of the descriptors mentioned above, the short-term features are clustered using k-means or Gaussian mixture models (GMM). Following which, the BoW descriptor is computed using the zeroth order statistics, VLAD requires the first-order statistics of the clusters and Fisher vectors are obtained by using both the first and second-order statistics. In [34], a BoW model was created using HOG and HOF features which were utilized for action classification. Similarly, Fisher vectors have also been extensively used as a feature for standard classifiers such as SVM [28] and feed-forward neural networks [2] to perform action recognition. Notably, Fisher vectors calculated with the motion descriptors such as HOF and MBH have shown good classification performance on large action datasets [26], [27]. A recent improvement in VLAD termed as VLAD^3 was used to provide a video-based representation [16] and was shown to perform better than Fisher vector on action datasets. In [8], an alternative descriptor called a super vector was computed using the maximum *aposteriori* (MAP) adaptation of the means of each mixture in the GMM. However, the resultant super vector is quite high-dimensional as the GMM contains many mixture components to accommodate all the actions and is also computationally expensive.

In the aggregation frameworks discussed above, either deliver a very high dimensional representation, or they are not specifically tuned for each action. So, in the proposed method, we aim to provide a distinct low-dimensional representation for each action. Also, most of the existing research demonstrates the use of some supervision in the form of manually annotated bounding boxes for feature extraction [26], [27]. In the proposed method, we extract HOG, HOF, and MBH features without using bounding boxes or body joint localization. Subsequently, we build a UAM to estimate the probability density function for implicit modelling of attributes across the actions. Using the UAM removes the need for manual annotation of attributes making it an excellent option even for fluid actions like *blowing out candles* and *playing a flute*. For the next step, a fixed-dimensional super action vector (SAV) is obtained by concatenating the adapted means of the UAM for a given clip. The SAV obtained is intrinsically low-dimensional because an action is composed of only a few attributes. So, to obtain a low-dimensional representation, we decompose the SAV using factor analysis. In the process, we get a low-dimensional representation for each clip to which we refer to as an action-vector. Finally, we demonstrate that even simple cosine scoring can be used for classifying action-vectors as they are found to be distinctive for each action. Unlike in most of the existing literature [2], [16], [28], this characteristic of the action-vectors eliminates the need for using class labels to build a classifier. Fig. 1 presents a block diagram of the entire process of action-vector extraction.

The main contributions of this paper are listed as follows.

- A universal model for representing actions in terms of their implicit attributes.
- Unsupervised extraction of a low-dimensional representation for each clip.
- Representation of an untrimmed clip using a single action-vector to highlight the relevant action while suppressing irrelevant background.

The rest of the paper is organized as follows. In Section II, action-vector extraction is discussed in detail through the process of UAM construction, SAV generation, and low-dimensional decomposition of SAV. Section III introduces the action-vector scoring mechanisms used in this work. The results and relevant discussion regarding both trimmed and untrimmed videos are covered in Section IV and the conclusions are presented in Section V.

II. Action-Vector Extraction

Each video clip can be considered as a sample function of the random process responsible for that action. In order to quantify the similarity between two action clips, we need to match the corresponding random processes. The random process of an action, in turn, involves sequence of random variables describing different attributes of the action. Hence, the similarity between two action clips depends on the parameters of the probability density functions (*pdf*) of the random variables associated with the random processes of those action. If we assume that the underlying *pdf* can be estimated using a GMM, then the number of mixtures must be large enough to accommodate the intra-action variability in the unconstrained videos. Unfortunately, a single clip does not have enough data points to estimate the

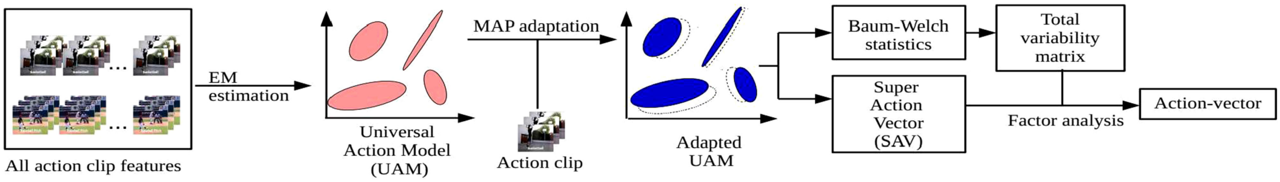


Fig. 1. Unsupervised attribute modeling and action-vector extraction.

pdf of the action. Further, research shows that training a GMM for every action is challenging especially for actions with few examples [8]. Hence, we propose to train a universal GMM using the clips of all the actions. We call this model the universal attribute model (UAM) which has a large number of mixtures for modelling the attributes of different actions spanning across datasets. However, it has been observed that even for a large number of actions spanning multiple datasets, the number of mixtures does not amount to be exceedingly large since they share attributes.

A. Universal Attribute Model (UAM)

The universal attribute model (UAM) can be represented as follows

$$p(\mathbf{x}_l) = \sum_{c=1}^{C} w_c \mathcal{N}(\mathbf{x}_l | \boldsymbol{\mu}_c, \boldsymbol{\sigma}_c), \tag{1}$$

where the mixture weights w_c satisfy the constraint $\sum_{c=1}^{C} w_c = 1$ and $\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c$ are the mean and covariance for mixture c of the UAM, respectively. A feature $\mathbf{x}_l$ is part of a clip $\mathbf{x}$ represented as a set of feature vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L$. This feature can be either a HOF or an MBH descriptor and we train a separate UAM for each during evaluation using standard EM estimation.

Since our goal is to find the *pdf* of the action process that generates a clip, we need to adapt the UAM parameters using the data in the clip. We perform a maximum *aposteriori* (MAP) adaptation similar to [8], [19] for obtaining the requisite *pdf* which describes the clip.

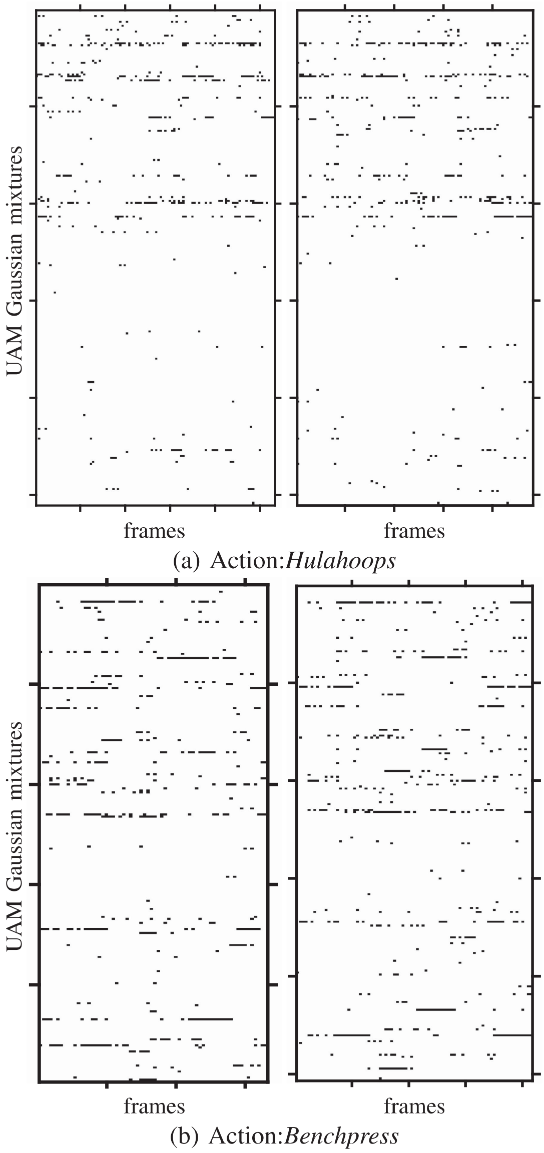
B. Super Action-Vector (SAV) Representation

In the MAP adaptation, the parameters of the UAM are adapted after observing each clip to enhance the contribution of the attributes present in that clip. The posterior probability of a mixture component (representing an attribute), given the feature vector from $\mathbf{x}_l \in \mathbb{R}^d$ from a clip, $p(c|\mathbf{x}_l)$ is given by

$$p(c|\mathbf{x}_l) = \frac{w_c p(\mathbf{x}_l|c)}{\sum_{c=1}^{C} w_c p(\mathbf{x}_l|c)}, \tag{2}$$

where $p(\mathbf{x}_l|c)$ is the likelihood of drawing a feature $\mathbf{x}_l$ from the cth mixture, and w_c is the prior probability of that mixture.

In Fig. 2 shows the posteriogram representation for two actions, viz., *Hulahoops* and *Benchpress*. The posteriogram is a 3-dimensional representation in which the intensity at a pixel (l,c) denotes the posterior probability of the cth Gaussian mixture for the lth feature vector $p(c|\mathbf{x}_l)$. Hence, the darker pixels correspond to the important attributes in the action. The posteriograms extracted from the action *Hulahoops*, performed by two different actors, shown in Fig. 2(a) show almost identical patterns. The slight changes in the patterns for two clips of the same action can be attributed to actor-specific and viewpoint-specific variations. A similar trend can be noticed for the *Benchpress* action as well, in Fig. 2(b).



(a) Action:*Hulahoops*

(b) Action:*Benchpress*

Fig. 2. Posteriograms (using 256 Gaussian mixtures) for two actions of UCF101: (a) *Hulahoops* and (b) *Benchpress*. Although the two clips of *Hulahoops* have variable number of frames, the sequence of Gaussian mixtures having the highest posterior probability (in black) is similar throughout the action. These mixtures represent the attributes which contribute to the action and the slight deviations may be caused by actor or viewpoint variability. Similar behavior can be observed in the clip of *Benchpress*.

The posterior probability $p(c|\mathbf{x}_l)$ computed in Equation (2) is then used to estimate the zeroth and first order Baum-Welch statistics [21] for a clip $\mathbf{x}$ given by

$$n_c(\mathbf{x}) = \sum_{l=1}^{L} p(c|\mathbf{x}_l), \text{ and} \tag{3a}$$

and

$$\mathbf{F}_c(\mathbf{x}) = \frac{1}{n_c(\mathbf{x})} \sum_{l=1}^{L} p(c|\mathbf{x}_l)\mathbf{x}_l, \tag{3b}$$

respectively. The MAP adapted parameters of a clip-specific model can be obtained as a convex combination of the UAM and the clip-specific statistics. For every mixture c of the UAM, the adapted weights and means are calculated as

$$\hat{w}_c = \alpha n_c(\mathbf{x})/L + (1-\alpha)w_c, \tag{4a}$$

$$\hat{\boldsymbol{\mu}}_c = \alpha\mathbf{F}_c(\mathbf{x}) + (1-\alpha)\boldsymbol{\mu}_c. \tag{4b}$$

The covariance matrices of the UAM are not adapted due to data insufficiency. The adapted means are concatenated to form a $(Cd \times 1)$-dimensional super action vector (SAV) for the clip as

$$\mathbf{s}(\mathbf{x}) = [\hat{\boldsymbol{\mu}}_1 \hat{\boldsymbol{\mu}}_2 \cdots \hat{\boldsymbol{\mu}}_C]^t. \tag{5}$$

While the SAV provides a fixed-dimensional representation for varying-length clips, it results in a high-dimensional representation. This representation though contains many of the attributes that do not contribute to the clip and hence are not changed from the original UAM. According to Equation (2), the likelihood $p(\mathbf{x}_l|c)$ is close to zero for mixtures in the UAM which do not model the attributes given by the features in the clip. This translates to posterior probability $p(c|\mathbf{x}_l)$ being close to zero for the same mixtures which leads the Baum-Welch statistics for these mixtures, $n_c(\mathbf{x})$ and $\boldsymbol{\mu}_c(\mathbf{x})$ to be close to zero. Finally, this leads to the MAP adapted weights $\hat{w}_c$ and means $\hat{\boldsymbol{\mu}}_c$ for these non-contributing mixtures to be the same as that in the UAM. Since each clip contains only a few of the total UAM mixtures (attributes), only those means are modified. Hence, the SAV is intrinsically low-dimensional, and through the use of a suitable decomposition, we can extract a representation of this kind, which has been referred to in this paper as an action-vector.

C. Action-Vector Representation

In order to arrive at a low-dimensional representation, the SAV $\mathbf{s}$ is decomposed as

$$\mathbf{s} = \mathbf{m} + \mathbf{T}\mathbf{w}, \tag{6}$$

where $\mathbf{m}$ is assumed to be an actor and viewpoint independent supervector. A supervector of this kind can be initialized using the UAM supervector. This is possible as the UAM is trained using large number of actors and viewpoints resulting in a distribution that is marginalized over views and actors. Also, $\mathbf{T}$ is a low-rank rectangular matrix known as the total variability matrix of size $Cd \times r$, and a r-dimensional random vector $\mathbf{w}$ whose prior distribution is assumed to be a standard Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$ [3]. We refer to this random vector as an *action-vector* which is a hidden variable and is defined by its posterior distribution $P(\mathbf{w}|\mathbf{x})$ after observing a clip $\mathbf{x}$ as $P(\mathbf{w}|\mathbf{x}) \propto P(\mathbf{x}|\mathbf{w})\mathcal{N}(\mathbf{0}, \mathbf{I})$

$$\begin{aligned}
&\propto \exp\left(\mathbf{w}\mathbf{T}^t\boldsymbol{\Sigma}^{-1}\tilde{\mathbf{s}}(\mathbf{x}) - \frac{1}{2}\mathbf{w}^t\mathbf{T}^t\mathbf{N}(\mathbf{x})\boldsymbol{\Sigma}^{-1}\mathbf{T}\mathbf{w} - \frac{1}{2}\mathbf{w}^t\mathbf{w}\right)\\
&= \exp\left(\mathbf{w}\mathbf{T}^t\boldsymbol{\Sigma}^{-1}\tilde{\mathbf{s}}(\mathbf{x}) - \frac{1}{2}\mathbf{w}^t\mathbf{M}(\mathbf{x})\mathbf{w}\right)\\
&= \exp\left(-\frac{1}{2}(\mathbf{w} - \mathbf{L}(\mathbf{x}))^t\mathbf{M}(\mathbf{x})(\mathbf{w} - \mathbf{L}(\mathbf{x}))\right) \times \mathit{constant}.
\end{aligned} \tag{7}$$

Here, $\boldsymbol{\Sigma}$ is a diagonal covariance matrix of dimension $Cd \times Cd$ and it models the residual variability that is left uncaptured by the total variability matrix $\mathbf{T}$. The matrix $\mathbf{L}(\mathbf{x})$ is defined as

$$\mathbf{L}(\mathbf{x}) = \mathbf{M}^{-1}(\mathbf{x})\mathbf{T}^t\boldsymbol{\Sigma}^{-1}\tilde{\mathbf{s}}(\mathbf{x}) \tag{8}$$

where $\tilde{\mathbf{s}}(\mathbf{x})$ is the centered supervector which appears because the posterior distribution of $\mathbf{w}$ is conditioned on the Baum-Welch statistics of the clip centered on the means of the UAM. The first order Baum-Welch statistics centered on the UAM mean can be obtained through

$$\tilde{\mathbf{F}}_c(\mathbf{x}) = \sum_{l=1}^{L} p(c|\mathbf{x}_l)(\mathbf{x}_l - \boldsymbol{\mu}_c). \tag{9}$$

We can now express $\tilde{\mathbf{s}}(\mathbf{x})$ as the concatenated first-order statistics given below

$$\tilde{\mathbf{s}}(\mathbf{x}) = \left[\tilde{\mathbf{F}}_1(\mathbf{x})\tilde{\mathbf{F}}_2(\mathbf{x}) \cdots \tilde{\mathbf{F}}_C(\mathbf{x})\right]^t. \tag{10}$$

Also, the matrix $\mathbf{M}(\mathbf{x})$ is defined as

$$\mathbf{M}(\mathbf{x}) = \mathbf{I} + \mathbf{T}^t\boldsymbol{\Sigma}^{-1}\mathbf{N}(\mathbf{x})\mathbf{T}, \tag{11}$$

where $\mathbf{N}(\mathbf{x})$ is a diagonal matrix of dimension $Cd \times Cd$ whose diagonal blocks are $n_c(\mathbf{x})\mathbf{I}$, for $c = 1, \ldots, C$ and $\mathbf{I}$ is the identity matrix of dimension $d \times d$.

From Equation (7), the mean and covariance matrix of the posterior distribution are given by

$$E[\mathbf{w}(\mathbf{x})] = \mathbf{M}^{-1}(\mathbf{x})\mathbf{T}^t\boldsymbol{\Sigma}^{-1}\tilde{\mathbf{s}}(\mathbf{x}) \tag{12a}$$

and

$$\mathrm{Cov}(\mathbf{w}(\mathbf{x}), \mathbf{w}(\mathbf{x})) = \mathbf{M}^{-1}(\mathbf{x}), \tag{12b}$$

respectively. Using EM algorithm [12], we iteratively estimate the posterior mean and covariance in the E-step and use the same to update $\mathbf{T}$ and $\boldsymbol{\Sigma}$ in the M-step.

In the first E-step of the estimation, $\mathbf{m}$ and $\boldsymbol{\Sigma}$ are initialized with the UAM mean and covariance, respectively. For the total variability matrix $\mathbf{T}$, a desired rank r is chosen, and the matrix is initialized randomly. Then $E[\mathbf{w}(\mathbf{x})]$ and $\mathrm{Cov}(\mathbf{w}(\mathbf{x}), \mathbf{w}(\mathbf{x}))$ calculated according to Equations (12a) & (12b).

In the M-step, the matrix $\mathbf{T}$ is calculated by solving

$$\sum_{\mathbf{x}} \mathbf{N}(\mathbf{x})\mathbf{T}E[\mathbf{w}(\mathbf{x})\mathbf{w}^t(\mathbf{x})] = \sum_{\mathbf{x}} \tilde{\mathbf{s}}(\mathbf{x})E[\mathbf{w}^t(\mathbf{x})], \tag{13}$$

which results in a system of r linear equations. The right hand side of Equation (13) contains $\tilde{\mathbf{s}}(\mathbf{x})$ which also accounts for the number of features in the clip. As $\mathbf{T}$ is the same for all the clips.

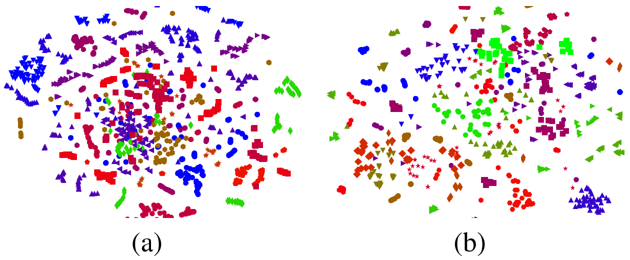


Fig. 3. t-SNE visualization on selected classes of UCF101 (a) MBH features (b) MBH action-vectors. Best viewed in color.

the left hand side is also multiplied by $\mathbf{N}(\mathbf{x})$ to account for the number of features in the clip.

For each $c = 1, \ldots, C$, the residual matrix $\boldsymbol{\Sigma}$ is estimated mixture-by-mixture as

$$\boldsymbol{\Sigma}_c = \frac{1}{n_c(\mathbf{x})}\left(\sum_{\mathbf{x}} \tilde{S}_c(\mathbf{x}) - \mathbf{M}_c\right) \tag{14}$$

where $\mathbf{M}_c$ denotes the cth diagonal block of the $Cd \times Cd$ matrix $\frac{1}{2}\sum_{\mathbf{x}} \tilde{\mathbf{s}}(\mathbf{x})E[\mathbf{w}^t(\mathbf{x})]T^t + TE[\mathbf{w}(\mathbf{x})]\tilde{\mathbf{s}}^t(\mathbf{x})$ and $\tilde{\mathbf{S}}_c(\mathbf{x})$ is the second-order Baum-Welch statistics of the clip centered on the means of the UAM calculated by

$$\tilde{\mathbf{S}}_c(\mathbf{x}) = diag\left(\sum_{l=1}^{L} p(c|\mathbf{x}_l)(\mathbf{x}_l - \boldsymbol{\mu}_c)(\mathbf{x}_l - \boldsymbol{\mu}_c)^t\right). \tag{15}$$

After the final M-step i.e. estimation of $\mathbf{T}$ and $\boldsymbol{\Sigma}$ matrices, the action-vector for a given clip can be represented using the mean of its posterior distribution as follows

$$\mathbf{w}(\mathbf{x}) = (I + \mathbf{T}^t\boldsymbol{\Sigma}^{-1}\mathbf{N}(\mathbf{x})\mathbf{T})^{-1}\mathbf{T}^t\boldsymbol{\Sigma}^{-1}\tilde{\mathbf{s}}(\mathbf{x}). \tag{16}$$

This process of obtaining the action-vector is known as *factor analysis* [12]. The $\mathbf{T}$-matrix contains the eigenvectors of the largest r eigenvalues of the total variability covariance matrix [3]. We hypothesize that these large eigenvalues arrive from the Gaussian mixture(s) which model the attributes in the clip. The original SAV can now be projected onto a r-dimensional action-vector based on T. The 2-D visualization of action-vectors ($r = 200$) is presented in Fig. 3(b) using t-distributed stochastic neighbor embedding (t-SNE) [17]. It follows, therefore, that most of the actions of UCF101 form easily identifiable clusters that which are in contrast to Fig. 3(a) where highly overlapping MBH features can be seen for the same actions. Hence, the proposed approach can effectively represent the video clip in a fixed dimensional space. The ability to obtain such a lower dimensional embedding confirms our hypothesis that SAVs are intrinsically low-dimensional. The visualization of action-vectors for the MBH feature (Fig. 3(b)) shows that there is a general distinction among actions which is obtained without the use of action labels. This visualization necessitates an exploration of the viability of action-vectors for the purpose of classification without explicitly training a classifier. Hence, we explore various scoring mechanisms like cosine scoring, linear discriminant analysis, and probabilistic linear discriminant analysis in the following section.

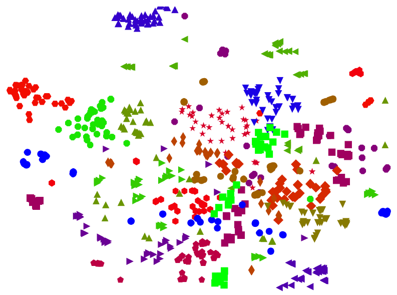


Fig. 4. t-SNE visualization of LDA projected MBH action-vectors on selected classes of UCF101. Best viewed in color.

III. Action-Vector Scoring

Started with the goal of comparing clips based on the underlying *pdf* of their actions and arrived at an action-vector representation. The *pdf* was estimated using a GMM where each mixture was assumed to learn an attribute. Hence, the action-vector extraction method of obtaining useful attributes to represent a clip is equivalent to obtaining the *pdf* for that clip. Once such a normalized representation (because of $\mathbf{N}(\mathbf{x})$) in Equation (16) has been obtained, we can directly compare the action-vectors of any two clips using cosine scoring.

Cosine scoring: For cosine scoring, the distance between a pair of action-vectors is expressed as

$$k(\mathbf{w}_1, \mathbf{w}_2) = \frac{\mathbf{w}_1^t\mathbf{w}_2}{\sqrt{\mathbf{w}_1^t\mathbf{w}_1}\sqrt{\mathbf{w}_2^t\mathbf{w}_2}}. \tag{17}$$

Linear Discriminant Analysis (LDA): While cosine scoring requires no class labels for evaluation, it cannot address intra-class variability. Such variability arises when recording the same action from different camera views which can reveal or hide certain action attributes. In a attempt to address this, we hypothesize that using class information in multi-class linear discriminant analysis can reduce the effect of intra-class variability. Hence, the coefficients of the projection matrix $\hat{\mathbf{A}}$ are chosen in a manner so as to maximize the ratio of the between-class scattering $\mathbf{S}_b$ to the within-class $\mathbf{S}_w$ scattering

$$\hat{\mathbf{A}} = \underset{\mathbf{A}}{\operatorname{argmax}} \frac{\mathbf{A}^t\mathbf{S}_b\mathbf{A}}{\mathbf{A}^t\mathbf{S}_w\mathbf{A}}. \tag{18}$$

The solution is to the above maximization problem is given by the following generalized eigenvalue problem

$$\mathbf{S}_b\mathbf{A} = \lambda\mathbf{S}_w\mathbf{A}. \tag{19}$$

There are at most $(m-1)$ generalized eigenvectors of $\mathbf{A}$ available to discriminate between m actions. Fig. 4 shows the LDA projected action-vectors onto 100 dimensions (highest available for 101 actions). Clearly, the different actions are better separated into clusters than Fig. 3(b).

Probabilistic Linear Discriminant Analysis (PLDA): Even though LDA solves the intra-class variability issue, the number of dimensions available for projection is always limited by the number of classes. In PLDA, there is no dimensionality constraint and it has been shown to produce better results than LDA for tasks like face recognition [1], [20] and object recognition [9]. Hence, we propose to use PLDA based action-vector scoring which is derived with a two-covariance model similar to LDA. The two-covariance model is known as a generative linear-Gaussian model, where latent vectors $\mathbf{y}$ representing actions are assumed to be distributed according to the prior distribution

$$p(\mathbf{y}) = \mathscr{N}(\mathbf{y}|\boldsymbol{\mu}, \mathbf{S}_b). \tag{20}$$

For a given action represented by a latent vector $\hat{\mathbf{y}}$, the distribution of action-vector $\mathbf{w}$ is assumed to be

$$p(\mathbf{w}) = \mathscr{N}(\mathbf{w}|\mathbf{y}, \mathbf{S}_w). \tag{21}$$

The maximum-likelihood estimates of the model parameters, $\boldsymbol{\mu}, \mathbf{S}_b$, and $\mathbf{S}_w$, can be obtained using EM algorithm. Now, in case of LDA, the projection of all the vectors using the projection matrix $\mathbf{A}$ would be followed by cosine scoring. In PLDA, the projection matrix $\mathbf{A}$ is not obtained explicitly and scoring is done using for every pair of action-vectors $(\mathbf{w}_1, \mathbf{w}_2)$ using the following two hypotheses

- *Null hypothesis* $\mathscr{H}_s$: A single latent vector $\hat{\mathbf{y}}$ representing the action is generated from the prior $p(\mathbf{y})$, for which both $\mathbf{w}_1$ and $\mathbf{w}_2$ are generated from $p(\mathbf{w}|\hat{\mathbf{y}})$.
- *Alternative hypothesis* $\mathscr{H}_d$: Two latent vectors representing two different actions are independently generated from $p(\mathbf{y})$. For each latent vector, either $\mathbf{w}_1$ or $\mathbf{w}_2$ is generated.

The score can now be calculated by means of a log-likelihood ratio between the two hypotheses $\mathscr{H}_s$ and $\mathscr{H}_d$ as

$$k(\mathbf{w}_1, \mathbf{w}_2) = \log \frac{p(\mathbf{w}_1, \mathbf{w}_2|\mathscr{H}_s)}{p(\mathbf{w}_1, \mathbf{w}_2|\mathscr{H}_d)} = \log \frac{\int p(\mathbf{w}_1|\mathbf{y})p(\mathbf{w}_2|\mathbf{y})}{p(\mathbf{w}_1)p(\mathbf{w}_2)}$$

where in the numerator, we integrate over the distribution of action-vectors to determine the likelihood of producing both action-vectors from the same action. For the hypothesis that action-vectors belong to separate actions, the product of the marginal likelihoods $p(\mathbf{w}_1)$ and $p(\mathbf{w}_2)$ is used.

IV. Experimental Results

In this section, we evaluate the performance of action-vector representation across different datasets and features. A detailed performance analysis of action-vectors across various features, different UAM configurations, and scoring mechanisms is performed on the UCF101 dataset. This is because UCF101 is one of the largest trimmed action dataset consisting of 13000+ clips of human actions spanning 101 classes [23]. Each video clip contains only the action of interest, and the average duration of each clip is around 7.21 seconds. Feature extraction on these clips in the form of HOG, HOF, and MBH descriptors is done (as part of the iDT framework [28]) without using any human annotations. Though the classification performance of iDT significantly improves when using human bounding boxes [27], annotations are either not available or expensive to acquire for most of the unconstrained videos.

A. Effect of Features and UAM Mixtures

Action-vector performance is evaluated for different UAMs and features on the UCF101 dataset. Every UAM is trained by pooling clips from one of the training splits mentioned in [23] for a particular feature. For each mixture in the UAM, diagonal covariance matrices are trained as there are a large number of mixtures and an insufficient number of features. Also, a similar number of feature vectors are chosen from each action to avoid the bias towards a particular action during training.

TABLE I
Classification Accuracy (%) for Various Classifiers on UCF101 Using SAV and Action-Vectors

Representation + Scoring technique	**HOG**	**HOF**	**MBH**
SAV + cosine	66.45	67.5	72.11
action-vector + cosine	87.17	88.80	90.67
action-vector + LDA	87.24	90.10	92.20
action-vector + SVM	88.54	91.24	93.88
action-vector + PLDA	88.84	92.73	**93.92**

Table I presents the classification performance of SAV and action-vectors using different scoring techniques, namely, (a) cosine, (b) LDA, (c) PLDA, and (d) SVM, over a 2048 mixture UAM. The scores obtained in Table I are produced by averaging over the officially provided test splits for UCF101. The action-vector dimension used is 200, and it is observed empirically that any further increase in feature dimension does not yield any noticeable increase in performance. For LDA scoring, 100 dimensions are used during projection. It can be seen from Table I that MBH consistently performs better than HOF and HOG. This is also true at feature level where MBH outperforms the other descriptors when used for classification with SVM [6], [27]. Also, action-vector representation performs much better compared to SAV which shows that removing redundant attributes results in a more discriminatory representation. The inherent capability of dissociation of action-vectors is evident as an unsupervised technique like cosine scoring shows comparable performance to supervised methods like SVM, LDA, and PLDA. As a scoring technique, PLDA consistently outperforms other scoring techniques like LDA and cosine scoring, and is marginally better than SVM. Hence, a projection matrix based on similarity or dissimilarity of action-vectors provides better discrimination than a single projection matrix for all the action-vectors.

Fig. 5 charts the performance of action-vectors across UAMs with varying number of mixtures. Action-vectors on 256 mixture UAM demonstrate low accuracy because the number of mixtures may not be enough to model all the attributes of the 101 actions in UCF101. The classification performance of the action-vector rises steadily for both SVM and PLDA classifiers with an increase in the number of the UAM mixtures. Hence, it is observed that adding mixtures leads to the representation of more distinctive attributes. However, there is no substantial improvement recorded beyond 2048 mixtures and in addition, training UAMs with 4096 mixtures increases computation time enormously. So, in order to further improve performance, we explore fusion of action-vectors for leveraging complimentary information.

B. Action-Vector Fusion

In [27], HOG, HOF, and MBH have been shown to extract complimentary information and they are combined in various ways to improve action recognition performance. So, we explore

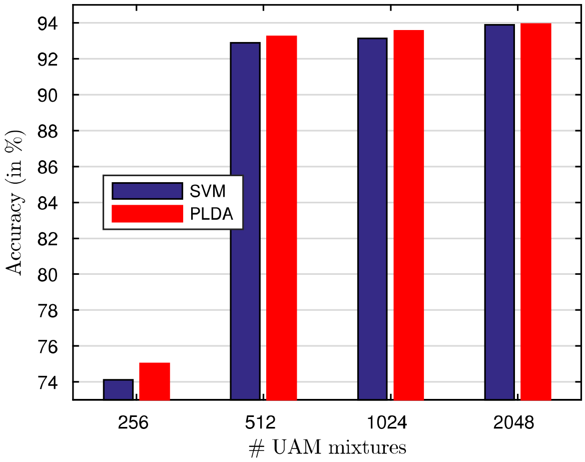


Fig. 5. Effect of number of UAM mixtures on action-vector extracted using MBH features for UCF101. Best viewed in color.

TABLE II
COMPARISON OF ACTION-VECTOR FUSION TECHNIQUES ON UCF101 (2048 MIXTURE UAM)

Feature combination	**Accuracy (in %)**			
	concat + cosine	**IF + SVM**	**PLDA-based ILDF**	**SF**
HOG + HOF	89.11	90.45	91.23	92.99
HOG + MBH	90.72	93.15	93.74	93.92
HOF + MBH	90.81	93.56	94.10	93.96
HOG + HOF + MBH	91.12	94.21	**95.13**	93.98

TABLE III
COMPARISON OF CLASSIFICATION ACCURACY OF PROPOSED APPROACH WITH EXISTING STATE-OF-THE-ART METHODS

Method	**Accuracy (in %)**		
	UCF101	**HMDB51**	**THUMOS14**
Supervised			
Spatio-Temporal CNN [39]	93.0	68.2	-
Multi-skip feature [14]	89.1	63.9	-
Long-term CNNs +iDT [25]	92.7	67.2	-
Two-stream CNN [22]	88.0	59.4	-
Temporal segment networks (TSN) [33]	94.3	69.4	-
HOF + MBH + Event model + BoW [35]	-	49.86	-
Objects + motion [10]	-	-	71.6
C3D features + iDT(fisher) [24]	90.4	-	-
Traj-pooled deep CNN + iDT(fisher) [31]	91.5	65.9	-
VLAD^3 + iDT(fisher) [16]	92.2	-	-
iDT-FV + DNN Hybrid [2]	92.4	70.4	-
iDT+CNN [30]	-	-	62.0
iDT+FV [29]	-	-	63.1
Temporal linear embedding (TLE) [4]	**95.6**	71.1	-
Action-vector (HOG + HOF + MBH) + SVM	94.3	81.0	70.9
Action-vector (HOG + HOF + MBH) + PLDA	95.1	**81.1**	**71.9**
Unsupervised			
Action-vector fusion (HOG + HOF + MBH) + cosine	91.1	79.3	67.8

different fusion techniques like: (a) concatenation (concat) of action-vectors with cosine scoring, (b) intermediate fusion (IF) of action-vectors scored with SVM classifier, (c) intermediate latent dimension fusion (ILDF) using PLDA, and (d) score fusion (SF) using PLDA scores. In Table II, we present the results using these fusion techniques on action-vectors. To perform intermediate fusion (IF), the action-vectors of different features for the same clip are concatenated and classified using an SVM. In case of IDLF, a PLDA model is trained on the concatenated action-vectors of the training set, following which, the concatenated action-vectors of the test clips are classified using this model. For score fusion (SF), a convex combination of PLDA scores is used which is optimized for accuracy. In all cases, the action-vectors and PLDA classification scores are obtained on 2048 mixture UAMs trained separately for each feature. It is observed that an action-dependant latent projection technique like PLDA performs better than either concatenation of action-vectors (intermediate fusion scored with SVM) or score fusion. For our next step, we present action-vector performance on other benchmark datasets and compare our performance with state-of-the-art techniques.

C. Comparison With State-of-the-Art

To explore the generalization capability of action-vectors, we present results on two other challenging action datasets: HMDB51 and THUMOS14. The HMDB51 dataset consists of 51 classes of actions containing 6766 clips [13]. On the other hand THUMOS14 is an untrimmed dataset [11] containing 1574 clips of the 101 actions from the UCF101 dataset. As the clips are untrimmed, the background activities affect the recognition rate of most algorithms. For extracting the action-vectors of clips in HMDB51, separate UAMs and corresponding $\mathbf{T}$-matrices are trained for all the three features. The action-vectors for untrimmed videos in THUMOS14 are extracted using the best performing UAM (2048 mixtures for all features) and corresponding $\vec{T}$-matrix on the UCF101 dataset. This is done in accordance to the THUMOS14 challenge where the UCF101 dataset is used as the training set for the THUMOS14 test clips [11].

Table III compares the classification performance of the proposed method with the state-of-the-art techniques used for action recognition on the UCF101, HMDB51, and THUMOS14 datasets. Some techniques like [14], [22], [25], [39] use long-term features to represent the entire video with a single feature. Other techniques use aggregation models like bag-of-words (BoW) [35], Fisher vector [2], [10], [24], [31] or VLAD [16]. Notably, many of the methods augment CNN features with iDT features to attain the high classification accuracy [25], [31]. Some recent methods either use attention [18], saliency [36], or weak supervision [32] to classify untrimmed clips. Among the methods that use long-term features, temporal linear embedding (TLE) networks [4] perform the best on UCF101. The proposed action-vectors produces comparable performance to TLE on UCF101 and comfortably outperforms it by 10% on the slightly more challenging HMDB51 (according to [13]).

For the untrimmed THUMOS14 dataset in particular, the state-of-the-art approaches [10], [10], [30] use fixed-sized windows to process untrimmed videos. Fixed-sized windows cannot handle high variations in the duration of actions. To mitigate this issue, fixed-sized windows of different temporal resolutions are used on the same clip their results are aggregated for the final decision [31]. Such approaches add to computational overhead as the same video is processed multiple times whereas in the proposed approach we compute a single action-vector for the entire video which outperforms these methods.

D. Analysis on Untrimmed Videos

In Fig. 6, we show the similarity between action-vectors obtained for the same action in two different datasets i.e. UCF101 and THUMOS14. The reason behind the similarity is that the

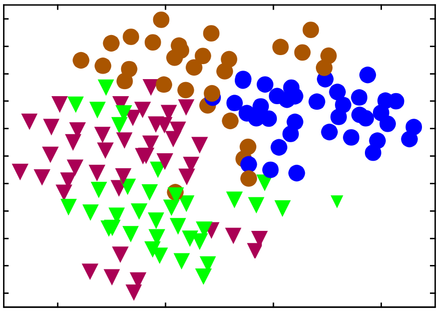


Fig. 6. t-SNE plot shows similarity in action-vectors of UCF101 and THUMOS14 across two classes—WritingOnBoard (UCF101,▼) (THUMOS14,▼) and WallPushups (UCF101,●) (THUMOS14,●).

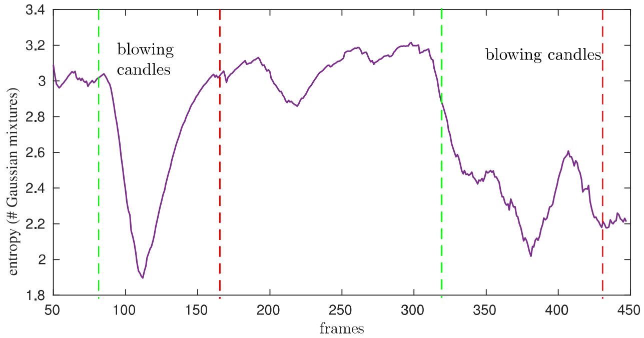


Fig. 7. Entropy plot for the number of UAM posteriors (using MBH features) for an untrimmed clip of *blowing candles* in THUMOS14.

total variability matrix $\mathbf{T}$ can faithfully reconstruct only the foreground actions that have been learned through the UAM and not the arbitrary background movements. This is important as the average length of the clips in THUMOS14 is around 3 minutes, but the duration of relevant actions is only around 4–5 seconds.

In order to understand how action-vectors enhance the mixtures belonging to the attributes of a specific action, an untrimmed clip containing the action, *blowing out candles* is presented in the form of an entropy plot of UAM posteriograms in Fig. 7. It is observed that during the action of interest, the entropy value is low and this denotes that only a few of the UAM mixtures get activated. These mixtures represent the attributes which form a part of the action. Apart from the actions of interest, the video also contains activities which are not in the 101 actions of UCF101 and have not been modeled by the UAM. During background activities of this kind, higher entropy values are observed which shows that an arbitrarily large number of mixture components get activated simultaneously. Behavior of this kind is expected from any action that is not modeled by the UAM because in that case there will be no particular set of attributes (components) which are affiliated with the action. It can also be observed that the entropy values change gradually when the clip transitions from the action to the background or vice-versa which is a result of MBH features being extracted over a period of 15 frames.

V. Conclusion

In this paper, we presented a technique to develop a compact and low-dimensional representation of actions called action-vectors. We first constructed a universal attribute model (UAM) for all the different actions across datasets. Following this, we used MAP adaptation on the UAM to retrieve super action-vector (SAV) representation for each clip. Subsequently, we employed factor analysis for obtaining action-vectors from SAV. The action-vector extraction process does not require labels and performs on-par with supervised techniques on both trimmed datasets HMDB51 and UCF101. Compared to existing fixed dimensional representations like Fisher vectors and VLAD, action-vectors perform better at action classification. The efficacy of action-vectors for classification of untrimmed videos of THUMOS14 shows that it is suitable even when the action is present for a relatively small amount of time in the entire clip. In addition, we also demonstrated state-of-the-art performance with intermediate latent dimension fusion of action-vectors using PLDA. In future, we would like to incorporate sequence information by replacing the UAM with a time-dependent hidden Markov model or recurrent neural networks.

References

[1] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun, "Bayesian face revisited: A joint formulation," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 566–579.

[2] C. R. de Souza, A. Gaidon, E. Vig, and A. M. López, *Sympathy for the Details: Dense Trajectories and Hybrid Classification Architectures for Action Recognition*. Cham, Switzerland: Springer International Publishing, 2016, pp. 697–716.

[3] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, May 2011.

[4] A. Diba, V. Sharma, and L. Van Gool, "Deep temporal linear encoding networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1541–1550.

[5] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 677–691.

[6] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "ActivityNet: A large-scale video benchmark for human activity understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 961–970.

[7] A. Gaidon, Z. Harchaoui, and C. Schmid, "Temporal localization of actions with actoms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2782–2795, Nov. 2013.

[8] N. Inoue and K. Shinoda, "A fast and accurate video semantic-indexing system using fast MAP adaptation and GMM supervectors," *IEEE Trans. Multimedia*, vol. 14, no. 4, pp. 1196–1205, Aug. 2012.

[9] S. Ioffe, *Probabilistic Linear Discriminant Analysis*. Berlin, Germany: Springer, 2006, pp. 531–542.

[10] M. Jain, J. C. van Gemert, and C. G. M. Snoek, "What do 15,000 object categories tell us about classifying and localizing actions?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 46–55.

[11] H. Idrees *et al.*, "The THUMOS challenge on action recognition for videos "in the wild"," *Comput. Vis. Image Understand.*, vol. 155, pp. 1–23, 2017.

[12] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 3, pp. 345–354, May 2005.

[13] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2556–2563.

[14] Z. Lan, M. Lin, X. Li, A. G. Hauptmann, and B. Raj, "Beyond gaussian pyramid: Multi-skip feature stacking for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 204–212.

[15] K. Li, J. Hu, and Y. Fu, "Modeling complex temporal composition of actionlets for activity prediction," in *Proc. 12th Eur.Conf. Comput. Vis.—Vol. Part I*, 2012, 286–299.

[16] Y. Li, W. Li, V. Mahadevan, and N. Vasconcelos, "VLAD3: Encoding dynamics of deep features for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1951–1960.

[17] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.

[18] Y. Peng, Y. Zhao, and J. Zhang, "Two-stream collaborative learning with spatial-temporal attention for video classification," *IEEE Trans. Circuits Syst. Video Technol.*, to be published, doi: 10.1109/TCSVT.2018.2808685.

[19] F. Perronnin, "Universal and adapted vocabularies for generic visual categorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 7, pp. 1243–1256, Jul. 2008.

[20] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, IEEE, 2007, pp. 1–8.

[21] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Process.*, vol. 10, no. 1, pp. 19–41, 2000.

[22] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., New York, NY, USA: Curran Associates, Inc., 2014, pp. 568–576.

[23] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," Dec. 2012, arXiv:1212.0402.

[24] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4489–4497.

[25] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," in *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1510–1517, Jun. 1, 2018, doi: 10.1109/TPAMI.2017.2712608.

[26] G. Varol and A. A. Salah, "Extreme learning machine for large-scale action recognition," in *Proc. Eur. Conf. Comput. Vis. Workshop*, vol. 7, 2014, pp. 8–16.

[27] H. Wang, D. Oneata, J. Verbeek, and C. Schmid, "A robust and efficient video representation for action recognition," *Int. J. Comput. Vis.*, vol. 119, no. 3, pp. 219–238, Jul. 2015.

[28] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 3551–3558.

[29] H. Wang and C. Schmid, "LEAR-INRIA submission for the thumos workshop," in *Proc. Int. Conf. Comput. Vis. Workshop Action Recognit. Large Number Classes*, vol. 2, 2013, pp. 8–11.

[30] L. Wang, Y. Qiao, and X. Tang, "Action recognition and detection by combining motion and appearance features," in *Proc. THUMOS Action Recognit. Challenge*, vol. 1, no. 2, 2014, pp. 2–10.

[31] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4305–4314.

[32] L. Wang, Y. Xiong, D. Lin, and L. V. Gool, "UntrimmedNets for weakly supervised action recognition and detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 6402–6411.

[33] L. Wang *et al.*, *Temporal Segment Networks: Towards Good Practices for Deep Action Recognition*. Cham, Switzerland: Springer International Publishing, 2016, pp. 20–36.

[34] S. Wang *et al.*, "Semi-supervised multiple feature analysis for action recognition," *IEEE Trans. Multimedia*, vol. 16, no. 2, pp. 289–298, Feb. 2014.

[35] J. Wu and D. Hu, "Learning effective event models to recognize a large number of human actions," *IEEE Trans. Multimedia*, vol. 16, no. 1, pp. 147–158, Jan. 2014.

[36] Y. Zhao and Y. Peng, "Saliency-guided video classification via adaptively weighted learning," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2017, pp. 847–852.

[37] F. Zhou, F. D. l. Torre, and J. K. Hodgins, "Hierarchical aligned cluster analysis for temporal clustering of human motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 582–596, Mar. 2013.

[38] Z. Zhou, F. Shi, and W. Wu, "Learning spatial and temporal extents of human actions for action detection," *IEEE Trans. Multimedia*, vol. 17, no. 4, pp. 512–525, Apr. 2015.

[39] Y. Zhu and S. Newsam, *Depth2Action: Exploring Embedded Depth for Large-Scale Action Recognition*. Cham, Switzerland: Springer International Publishing, 2016, pp. 668–684.

Debaditya Roy (S'15) received the B.Tech. degree from the West Bengal University of Technology, West Bengal, India, in 2011, the M.Tech. degree from the National Institute of Technology Rourkela, Odisha, India, in 2013, and the Ph.D. degree from the Indian Institute of Technology Hyderabad, Hyderabad, India, in 2018. He is currently a Postdoctoral researcher with the Department of Transportation Systems Engineering, Nihon University, Tokyo, Japan. His research interests include deep learning, computer vision, and traffic surveillance analysis.

Kodukula Sri Rama Murty received the B.Tech. degree in electronics and communications engineering from Jawaharlal Nehru Technological University, Hyderabad, India, in 2002, and the Ph.D. degree from the Indian Institute of Technology Madras, Chennai, India, in 2009.

In 2009, he joined, as an Assistant Professor, the Department of Electrical Engineering, Indian Institute of Technology Hyderabad, Hyderabad, India, where he is currently an Associate Professor. His research interests include signal processing, speech analysis, and machine learning.

Chalavadi Krishna Mohan (M'14) received the bachelor's in science education (B.Sc.Ed) degree from the Regional Institute of Education, Mysore, India, in 1988, the master's degree in computer applications from the S. J. College of Engineering, Mysore, India, in 1991, the master's degree in technology in system analysis and computer applications from the National Institute of Technology Surathkal, India, in 2000 and the Ph.D. degree in computer science and engineering from the Indian Institute of Technology Madras, Chennai, India, in 2007. He is currently a Professor with the Department of Computer Science and Engineering, Indian Institute of Technology Hyderabad, India. His research interests include video content analysis, pattern recognition, and neural networks.