Action-Vectors: Unsupervised movement modeling for action recognition

Debaditya Roy *‡, K. Sri Rama Murty †‡ and C. Krishna Mohan *‡
*Visual Learning and Intelligence Group (VIGIL),
Department of Computer Science and Engineering
†Department of Electrical Engineering
‡Indian Institute of Technology Hyderabad, India
{cs13p1001, ksrm, ckm}@iith.ac.in

***Abstract*—Representation and modelling of movements play a significant role in recognising actions in unconstrained videos. However, explicit segmentation and labelling of movements are non-trivial because of the variability associated with actors, camera viewpoints, duration etc. Therefore, we propose to train a GMM with a large number of components termed as a universal movement model (UMM). This UMM is trained using motion boundary histograms (MBH) which capture the motion trajectories associated with the movements across all possible actions. For a particular action video, the MAP adapted mean vectors of the UMM are concatenated to form a fixed dimensional representation referred to as "super movement vector" (SMV). However, SMV is still high dimensional and hence, Baum-Welch statistics extracted from the UMM are used to arrive at a compact representation for each action video, which we refer to as an "action-vector". It is shown that even without the use of class labels, action-vectors provide a more discriminatory representation of action classes translating to a 8 % relative improvement in classification accuracy for action-vectors based on MBH features over naïve MBH features on the UCF101 dataset. Furthermore, action-vectors projected with LDA achieve 93% accuracy on the UCF101 dataset which rivals state-of-the-art deep learning techniques.**

***Keywords*—action recognition, unsupervised learning, fixed-dimensional representation**

I. Introduction

Most human actions can be decomposed into movements which make up the actions. For example, *boxing* can be decomposed into smaller movements like right hand punching forward, right hand retracting, left hand punching forward and left hand retracting. However, in case of some actions like *cricket batting, applying eye makeup* etc., such crisp definition of movements and is challenging. This issue is further compounded in unconstrained videos because of viewpoints changes, varying duration of actions and different actors performing the same action. To address this issue, an extension of the bag-of-features model known as "actoms" are proposed [1]. In this representation, an action is a sequence of histograms of actom-anchored visual features. Another approach [2] proposes to decompose complex activities into "actionlets" and the large and small order Markov dependencies between these actionlets are represented using a probabilistic suffix tree. In both these methods, the requirement of labelling the action units and/or explicit modeling of dependency between the action units has to be performed manually. To overcome this issue, we propose an unsupervised movement model (UMM) to discover the various movements from the actions independent of each other in order to avoid both labeling and dependency estimation.

Some other approaches involve the use of additional information (like the exact position of the limbs) to estimate movements and then modeling the action as a sequence of such movements [3]. Even movement recognition in 3D videos is perfomed with depth-map based information which could pinpoint the location of limbs [4]. In both these instances, the core problem of pose estimation of the human body for movement recognition could be bypassed because of availability of limb positions. However, for unconstrained videos such information is difficult to obtain and we present a solution which utilizes short-term motion information to discover movements.

Another parallel trend in action recognition advocates the modeling of long-term dynamics [5], [6]. The short-range dynamics (from CNNs), medium-range (from linear dynamical systems) and long-range dynamics (from pooling the VLAD descriptors on short-term motion) are fused for action classification [5]. On a relevant note, long-term temporal convolutions (LTC) till 100 frames have been proposed for action description and the best performance is obtained with a optical flow based LTC network with pooled output from both 60 and 100 frames [6]. Notably, the short-term dynamics were shown to be perform close to the long-range dynamics. This motivates the use of highly overlapping short-term dynamics like motion boundary histograms (MBH) [7]. However, raw MBH features cannot be directly used for matching action clips, as they depend on the duration of the video leading to varying length patterns. This is solved by producing a super movement vector (SMV) for each action clip after MAP adaptation of the UMM model.

Generally, a UMM model contains a lot of mixture components to accommodate all the movements in the action classes which leads to a very high-dimensional super movement vector. In the past, dimensionality reduction and encoding has been performed to get a low-dimensional representation [7], [8]. In most of the approaches though, dimensionality reduction and encoding have been investigated separately. Especially in [8], PCA is followed by GMM based clustering which is encoded as Fisher vectors to obtain the desired representation. In this paper, we show that dimensionality reduction and encoding need to be performed in tandem to achieve the best representation of the super movement vector. The encoded vector thus obtained is termed "action-vector" and it considers first and second-order statistics of feature vectors in

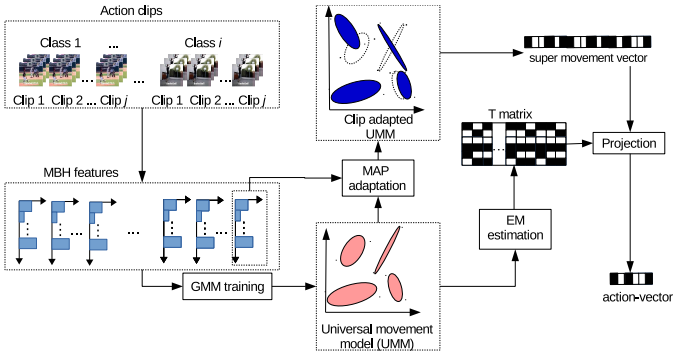


Fig. 1. Block diagram of action-vector extraction

addition to the zeroth order statistics (used in VLAD and Fisher vector). These "action-vectors" are compact representations of SMV and are shown to extract class-based information by grouping the movements of the same class together even in the absence of class labels. The entire procedure is shown as a block diagram in figure 1 and the details are discussed in the subsequent sections.

II. Universal Movement Model (UMM)

Given an action clip consisting of L feature vectors $\boldsymbol{x} = \boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_l}$ which represent the various movements present in the action, the task is to find whether $\boldsymbol{x}$ belongs an action class a [9]. This single-action detection problem can be represented as a basic hypothesis test between:

$$H_0 : \boldsymbol{x} \text{ is from the action class } a$$
$$\text{and}$$
$$H_1 : \boldsymbol{x} \text{ is not from the action class } a$$

To decide between the two hypotheses, a likelihood ratio test is devised as follows:

$$\frac{p(\boldsymbol{x}|H_0)}{p(\boldsymbol{x}|H_1)} \begin{cases} \geq \theta & accept \quad H_0 \\ < \theta & reject \quad H_0 \end{cases} \tag{1}$$

where $p(\boldsymbol{x}|H_i), i = \{0, 1\}$, is the probability density function for the hypothesis H_i evaluated for the observed action clip x, also referred to as the likelihood of the hypothesis H_i given the action clip and θ is the decision threshold for accepting or rejecting H_0.

However, instead of training a class-dependent model for each action class to evaluate the likelihood $p(\boldsymbol{x}|H_0)$, a class-independent universal movement model (UMM) is trained by pooling a balanced subset of all the feature vectors obtained from all the action classes which is then adapted for each action class. The likelihood term $p(\boldsymbol{x}|H_1)$ represents the UMM likelihood and need not be calculated separately as it same for all the action classes. Also, instead of a decision threshold θ, the class-adapted UMM for which $p(\boldsymbol{x}|H_0)$ attains the maximum value determines the class of $\boldsymbol{x}$.

Given a UMM and training vectors of an action clip $\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_L\}$ as before, at first, the probabilistic alignment of the training vectors into the UMM mixture components is calculated. For every mixture c in the UMM, the likelihood $p(c|\boldsymbol{x}_l)$ is computed as follows:

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

where $\boldsymbol{x}_l$ is a $d \times 1$ feature vector, $p_c(\boldsymbol{x})$ is a uni-modal Gaussian density parametrized by a mean $d \times 1$ vector, $\boldsymbol{\mu}_c$, and a $d \times d$ covariance matrix, Σ_c and is represented as:

$$p(\boldsymbol{x}_l|c) = \frac{1}{(2\pi)^{d/2}|\Sigma_c|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}_l - \boldsymbol{\mu}_c)^t(\Sigma_c)^{-1}(\boldsymbol{x}_l - \boldsymbol{\mu}_c)\right\}. \tag{3}$$

and the mixture weights satisfy the constraint $\sum_{c=1}^{C} w_c = 1$. In practice, diagonal covariance $\boldsymbol{\sigma}_c$ is used instead of the full covariance Σ_c.

The computed likelihood $p(c|\boldsymbol{x}_l)$ is then used to calculate the Baum-Welch statistics for weights, mean and variance parameters:

$$n_c(\boldsymbol{x}) = \sum_{l=1}^{L} p(c|\boldsymbol{x_l}) \tag{4a}$$

$$\boldsymbol{\mu}_c(\boldsymbol{x}) = \frac{1}{n_c(\boldsymbol{x})} \sum_{l=1}^{L} p(c|\boldsymbol{x}_l)\boldsymbol{x}_l \tag{4b}$$

$$\boldsymbol{\sigma}_c(\boldsymbol{x}) = \frac{1}{n_c(\boldsymbol{x})} \sum_{l=1}^{L} p(c|\boldsymbol{x}_l)\boldsymbol{x}_l\boldsymbol{x}^t. \tag{4c}$$

These new statistics obtained from the training data are then used to update the old UMM statistics for mixture c to create the adapted parameters for mixture c as follows:

$$\hat{w}_c = [\alpha_i^w n_c(\boldsymbol{x})/L + (1 - \alpha_i^w)w_c]\gamma \tag{5a}$$

$$\hat{\boldsymbol{\mu}}_c = \alpha_i^m \boldsymbol{\mu}_c(\boldsymbol{x}) + (1 - \alpha_i^m)\boldsymbol{\mu}_c \tag{5b}$$

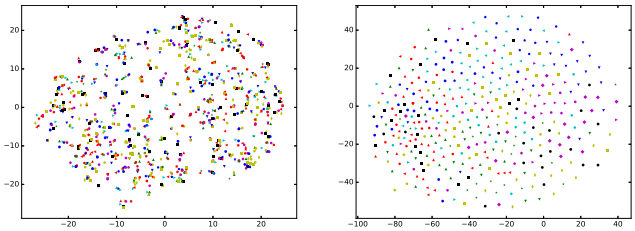
$$\hat{\boldsymbol{\sigma}}_c^2 = \alpha_i^v \boldsymbol{\sigma}_c(\boldsymbol{x}) + (1 - \alpha_i^v)(\boldsymbol{\sigma}_c^2 + \boldsymbol{\mu}_c^2) - \hat{\boldsymbol{\mu}}_c^2. \tag{5c}$$

The adaptation coefficients $\{\alpha_i^w, \alpha_i^m, \alpha_i^v\}$ control the balance between new and old estimates for weights, means and variances, respectively. Further, a scale factor γ is calculated over all adapted mixture weights to ensure they sum up to unity. Apart from equation 5a, rest of the adaptations can be obtained from general MAP estimate for a GMM. Also, the update is for the statistics and not for the derived parameters like variance etc.

The means of the adapted UMM for each action clip are then concatenated to compute a $Cd \times 1$ dimensional supervector termed as a super movement vectors (SMV) $\boldsymbol{v}$. Obtaining a fixed-dimensional representation normalizes the effect of varying durations across action clips. A visualization of SMVs is shown in figure 2(b) for the 101 classes of UCF101 dataset [10]. It can be observed that the overlap between many of the classes as present for the MBH features (figure 2(a)) is carried over here as well. This is expected as many of the classes like *golf swing, cricket shot* and *hammer throw* share similar movements. It also shows that each movement is modelled using one/more Gaussians in the UMM model and they are shared across actions. Also, being an unsupervised learning technique, no explicit labeling for movements is required for obtaining this behaviour of SMVs.

III. Extraction of Action-Vectors

In addition to the action-specific movements, viewpoint variations and the duration also add undesired variability to the action clips. In order to extract action-vectors, which are representatives of particular actions, the variability introduced by viewpoint and duration have to be normalized. The effect



(a) (b)

Fig. 2. t-SNE visualization on UCF101 dataset: (a) MBH features (b) super movement vectors

of varying duration is absorbed while obtaining a fixed dimensional super movement vector (SMV). In order to arrive at a viewpoint invariant representation, SMV can be decomposed into actor and viewpoint subspaces. This is motivated from speaker recognition where a speech utterance is decomposed into channel and speaker subspaces using joint factor analysis (JFA) [11]. However, more recently, a total variability space which jointly models both the factors was found to be more effective than JFA for speaker recognition [12]. Accordingly, we represent each SMV $\boldsymbol{v}$ as:

$$\boldsymbol{v} = \boldsymbol{m} + T\boldsymbol{w} \tag{6}$$

where $\mathbf{m}$ is the actor and viewpoint independent mean vector which can be obtained from the UMM and T is a low-rank total variability matrix which captures actor and viewpoint variability. The reduced dimensional coefficient vector $\boldsymbol{w}$ is referred to as the "action-vector" and is estimated by enforcing a Gaussian distribution. The action-vector is characterised by $\boldsymbol{m}$, T and a diagonal covariance matrix Σ, of dimension $Cd \times Cd$, which captures the residual error in approximating $\boldsymbol{v}$ [12].

The total variability matrix T and the residual error covariance Σ are estimated using expectation maximization (EM). In the E-step of the EM estimation of T, for each action clip $\boldsymbol{x}$, the parameters of the posterior distribution of $\boldsymbol{w}(\boldsymbol{x})$ are estimated using the current estimates of $\boldsymbol{m}$, T and Σ. In the M-step, T and Σ are updated by solving a set of linear equations in which the $\boldsymbol{w}(\boldsymbol{x})$'s calculated in the previous step act as explanatory variables. The detailed derivations for the E-step and M-steps can be found in [13]. The EM algorithm requires centered Baum-Welch statistics which can be obtained from equations 4b and 4c as follows:

$$\tilde{\boldsymbol{\mu}}_c(\boldsymbol{x}) = \sum_{l=1}^{L} p(c|\boldsymbol{x}_l)(\boldsymbol{x}_l - \boldsymbol{\mu}_c) \tag{7a}$$

$$\tilde{\boldsymbol{\sigma}}_c(\boldsymbol{x}) = diag\left(\sum_{l=1}^{L} p(c|\boldsymbol{x}_l)(\boldsymbol{x}_l - \boldsymbol{\mu}_c)(\boldsymbol{x}_l - \boldsymbol{\mu}_c)^t\right) \tag{7b}$$

In the first E-step of the estimation, $\boldsymbol{m}$ and Σ are initialized with the UMM mean and covariance, respectively. For the total variability matrix T, a desired rank r is chosen and the matrix is initialized randomly. The action-vector $\boldsymbol{w}$ is initialized with a standard normal distribution. In order to estimate the posterior distribution of $\boldsymbol{w}$ after observing an action clip $\boldsymbol{x}$, let us define a matrix $M(\boldsymbol{x})$ as:

$$M(\boldsymbol{x}) = I + T^t\Sigma^{-1}N(\boldsymbol{x})T \tag{8}$$

where $N(\boldsymbol{x})$ is a diagonal matrix of dimension $Cd \times Cd$ whose diagonal blocks are $n_c(\boldsymbol{x})I$ for $c = 1, ..., C$. Then the posterior distribution of $\boldsymbol{w}(\boldsymbol{x})$ is a Gaussian with mean and covariance matrix defined as:

$$\boldsymbol{\mu}(\boldsymbol{w}(\boldsymbol{x})) = M^{-1}(\boldsymbol{x})T^t\Sigma^{-1}\tilde{\boldsymbol{\mu}}(\boldsymbol{x}) \tag{9a}$$

$$\Sigma(\boldsymbol{w}(\boldsymbol{x})) = M^{-1}(\boldsymbol{x}) \tag{9b}$$

Here, $\tilde{\boldsymbol{\mu}}(\boldsymbol{x})$ is a supervector of dimension $Cd \times 1$ obtained by concatenating all first-order BaumWelch $\tilde{\boldsymbol{\mu}}_c(\boldsymbol{x})$ statistics obtained in equation 7a for a given utterance $\boldsymbol{x}$.

In the M-step, some additional statistics are accumulated over all action clips such as:

$$n_c = \sum_{\boldsymbol{x}} n_c(\boldsymbol{x}), \tag{10a}$$

$$\mathscr{A}_c = \sum_{\boldsymbol{x}} n_c(\boldsymbol{x}) \left\{\Sigma(\boldsymbol{w}(\boldsymbol{x})) + \boldsymbol{\mu}(\boldsymbol{w}(\boldsymbol{x}))\boldsymbol{\mu}(\boldsymbol{w}^t(\boldsymbol{x}))\right\} \tag{10b}$$

$$\mathscr{C} = \sum_{\boldsymbol{x}} \tilde{\boldsymbol{\mu}}(\boldsymbol{x})\boldsymbol{\mu}(\boldsymbol{w}^t(\boldsymbol{x})), \tag{10c}$$

$$N = \sum_{\boldsymbol{x}} N(\boldsymbol{x}). \tag{10d}$$

Finally, the matrices T and Σ are updated as follows:

$$T(i,:)\mathscr{A}_c = \mathscr{C}_i, \tag{11a}$$

$$\Sigma = N^{-1}\left(\sum_{\boldsymbol{x}} \tilde{\Sigma}(\boldsymbol{x}) - diag(\mathscr{C}T^t)\right), \tag{11b}$$

where $i = (c-1)d + j$ and $j = 1, ..., d$, d being the dimension of the feature vector, c is the number of mixture components in the GMM and $\tilde{\Sigma}(\boldsymbol{x})$ is a block diagonal matrix of dimension $Cd \times Cd$ whose diagonal blocks are $\boldsymbol{\sigma}_c(\boldsymbol{x})(c = 1, ..., C)$. After the estimation of the T matrix, action-vector $\boldsymbol{w}$ for a given action clip $\boldsymbol{x}$ can be obtained using the following equation:

$$\boldsymbol{w}(\boldsymbol{x}) = (I + T^t\Sigma^{-1}N(\boldsymbol{x})T)^{-1}T^t\Sigma^{-1}\tilde{\boldsymbol{\mu}}(\boldsymbol{x}). \tag{12}$$

The total variability matrix T (used for projection of $\boldsymbol{w}(\boldsymbol{x})$) suppresses the intra-class variability across actors and viewpoints and projects each action clip belonging to the same class closer to each other in the action-vector space. Such a visualization of action-vectors (200 dimensional) is presented in 3(a) and it can be noticed that most of the action classes of UCF101 form clusters which can be easily identified. This is in stark contract to figures 2(a) and 2(b) and demonstrates the strength of action-vector in retaining the unique signature of each action among its clips without the use of labels. Thus, by utilizing the first and second-order statistics, w uncovers the hidden patterns in the original features in a lower-dimensional space. This is especially relevant in action recognition as the actual space of actors' movements and viewpoints in which the actions lie is generally very sparse which is exploited in sparse modeling of actions [14].

Further, action-vector represents each action clip as a fixed dimensional feature vector which can then be efficiently used for classification using any scoring mechanism. One such scoring mechanism that is used, among others is linear discriminant analysis (LDA). Figure 3(b) shows the LDA projected action-vectors onto 100 planes (highest available for 101 classes).

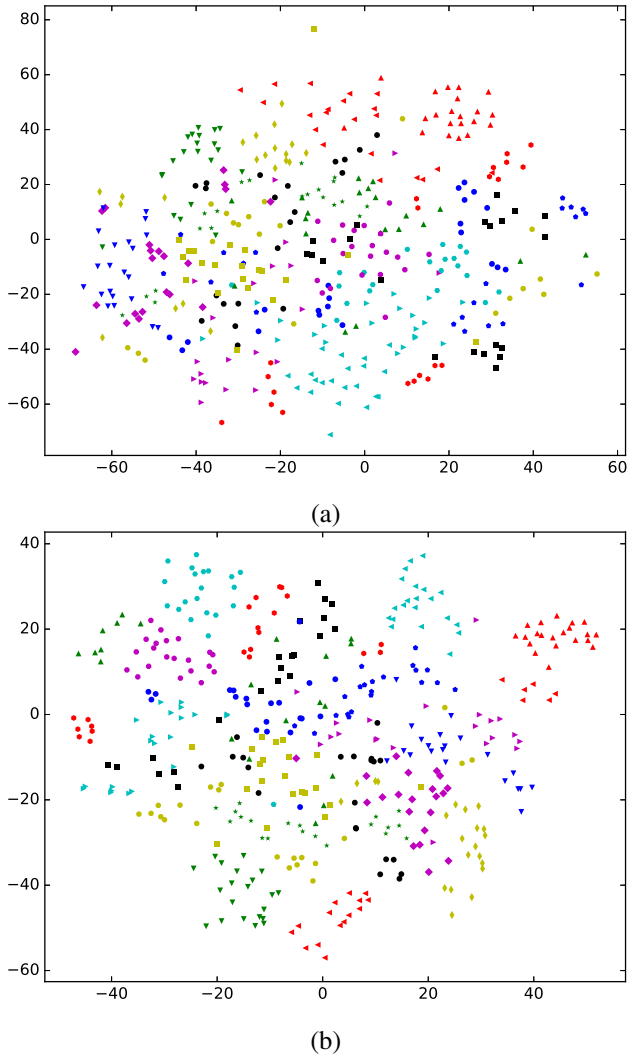


Fig. 3. t-SNE visualization UCF101: (a) action-vectors (b) LDA projected action-vectors

It can be observed that LDA works even better with action-vectors and makes the clusters even more separated resulting in improvement of classification performance over cosing scoring as shown in table I.

IV. Experiments and Results

The experiments for action recognition were performed on the UCF101 action recognition dataset [10]. The UCF101 dataset is one of the biggest annotated datasets for human action recognition with 13000+ clips covering 101 actions. The feature representation which is used to derive the action-vector representation in this paper is motion boundary histogram (MBH). MBH divides the optical flow (temporal derivative of the position of trajectory points in consecutive frames) $\omega = (u, v)$ into its horizontal u and vertical v components by computing separate spatial derivatives which reduce irrelevant background motion.

Table I lists the performance of the proposed action-vector representation with different number of UMM components. For LDA scoring, 100 dimensions were used for the projection (maximum available for 101 classes). It can be seen that the classification performance of action-vector rises steadily for all the 3 scoring mechanisms as the number of UMM components increase. As action-vectors by themselves cluster the classes without the use of labels, simple cosine scoring shows comparable performance to other projection methods like LDA and probabilistic LDA (PLDA) [15]. This shows that given enough number of components for representation, action-vectors can extract a fixed dimensional representation of each of the action clips while maintaining discriminative information.

TABLE I. Classification accuracy of action-vectors (%) vs. UMM components

# components	Cosine	LDA	PLDA
256	90.5	79.8	73.4
512	90.7	91.9	92.5
1024	90.8	92.1	92.7
2048	90.7	92.2	93.0

Table II shows a comparison of the classification proposed method with the state-of-the-art techniques used for action recognition on the UCF101 dataset. While [16], [17] and [18] [19] all use CNN based feature extraction methods, [5] and [19] use improved dense trajectory (iDT) features in conjunction with CNN features. On the other hand, the proposed method uses motion boundary histogram (MBH) based features, which though are used for camera calibration, they also estimate the gross motion information along the x-axis and y-axis separately. This information empirically seems to provide more information about human action which is useful for classification. Further, without using any of the other "complimentary" information about structure (HOG), motion of the tracked point (HOF) and direction of the tracked point (trajectory), MBH proves to be more than adequate for classification which is in-line with the observation made about MBH in [20].

TABLE II. Comparison of classification accuracy with state-of-the-art

Method	Accuracy (%)
HOG+HOF+MBH [21]	86.0
C3D features [16]	90.4
Multi-skip feature stacking [17]	89.1
Two-stream CNN [18]	88.0
Trajectory-pooled deep CNN [19]	91.5
$VLAD^3$ + iDT(fisher) [5]	92.2
Long-term temporal CNNs +iDT [6]	92.7
Action-vector + PLDA	**93.0**

V. Conclusion

In this paper, we presented an approach to arrive at fixed dimensional feature vectors for representing human actions by training a universal movement model (UMM) that captures the motion trajectories movements across all the possible actions. The high dimensional super movement vector obtained from the UMM was then encoded to a lower dimensional action-vector. These action-vectors were shown to be highly discriminative and further so using LDA based projection. Our experiments on the 101 classes of the UCF101 dataset show that action vectors with LDA rival the performance of state-of-the-art deep learning techniques. In future, combination of different feature vectors can be explored for action-vector based encoding.

REFERENCES

[1] A. Gaidon, Z. Harchaoui, and C. Schmid, "Temporal localization of actions with actoms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2782–2795, Nov 2013.

[2] K. Li, J. Hu, and Y. Fu, "Modeling complex temporal composition of actionlets for activity prediction," in *Proceedings of the 12th European Conference on Computer Vision - Volume Part I*, ser. ECCV'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 286–299.

[3] N. P. Cuntoor, B. Yegnanarayana, and R. Chellappa, "Activity modeling using event probability sequences," *IEEE Transactions on Image Processing*, vol. 17, no. 4, pp. 594–607, April 2008.

[4] S. R. Fanello, I. Gori, G. Metta, and F. Odone, "Keep it simple and sparse: real-time action recognition." *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2617–2640, 2013.

[5] Y. Li, W. Li, V. Mahadevan, and N. Vasconcelos, "Vlad3: Encoding dynamics of deep features for action recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[6] G. Varol, I. Laptev, and C. Schmid, "Long-term Temporal Convolutions for Action Recognition," *arXiv:1604.04494*, 2016.

[7] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3551–3558.

[8] G. Varol and A. A. Salah, "Extreme learning machine for large-scale action recognition," in *ECCV workshop*, 2014.

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

[10] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, vol. abs/1212.0402, 2012. [Online]. Available: http://arxiv.org/abs/1212.0402

[11] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 5, pp. 980–988, 2008.

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

[13] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1435–1447, 2007.

[14] A. Castrodad and G. Sapiro, "Sparse modeling of human actions from motion imagery," *International journal of computer vision*, vol. 100, no. 1, pp. 1–15, 2012.

[15] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[16] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "C3D: generic features for video analysis," *CoRR*, vol. abs/1412.0767, 2014. [Online]. Available: http://arxiv.org/abs/1412.0767

[17] Z. Lan, M. Lin, X. Li, A. G. Hauptmann, and B. Raj, "Beyond gaussian pyramid: Multi-skip feature stacking for action recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[18] 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. Curran Associates, Inc., 2014, pp. 568–576. [Online]. Available: http://papers.nips.cc/paper/5353-two-stream-convolutional-networks-for-action-recognition-in-videos.pdf

[19] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4305–4314.

[20] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[21] H. Wang, D. Oneata, J. Verbeek, and C. Schmid, "A robust and efficient video representation for action recognition," *International Journal of Computer Vision*, vol. 119, no. 3, pp. 219–238, Jul. 2015. [Online]. Available: https://hal.inria.fr/hal-01145834