A COMPACT PAIRWISE TRAJECTORY REPRESENTATION FOR ACTION RECOGNITION

*Qingya Huang, Shan Sun, Feng Wang**

Shanghai Key Laboratory of Multidimensional Information Processing
East China Normal University

ABSTRACT

Dense trajectories are widely used in human action recognition. However, the relationships among trajectories are rarely exploited and a large mount of useful information is missing. In this paper, we propose a novel approach to employ the space-time relationships between different trajectories for action recognition. In our approach, each trajectory is paired up with several neighbors which are spatially and temporally close to it. A GMM (Gaussian Mixture Model) is then trained and Fisher Vector is employed to quantify the pairwise trajectories. In this way, the local spatial and temporal structure information around each trajectory is explored for feature representation, which improves the discriminative ability of the features. The experimental results on several benchmark datasets show that our pairwise trajectory representation outperforms the state-of-the-art approaches.

Index Terms— Pairwise trajectory, action recognition.

1. INTRODUCTION

Action recognition [1, 2] in videos is playing an increasingly important role in computer vision. Although there have been numerous works [3, 4, 5, 6] on this issue, it is still one of the challenging problems in computer vision due to the background clutter, occlusions, viewpoint variations, inter- and intra-class variations, and subject differences.

Recently, a common solution using improved dense trajectory (IDT) and fisher vector (FV) shows the state-of-the-art performance on several popular benchmarks [7, 8, 9, 10]. In this solution, Gaussian Mixture Model (GMM) based codebook is built to represent the distribution of local features in training videos, which is then used to encode the local features of a given video. In this approach, all trajectories are encoded disorderly, while the relationship between different trajectories is ignored. For instance, in Figure 1, assuming there are two videos A and B of two different action categories, the upper part shows the extracted trajectories and their distributions in two videos respectively. As can be observed in Figure 1, for each trajectory in A, there exists a similar trajectory in B. With the traditional encoding approach, the results for A and B are similar and thus it is hard to discriminate the two actions. Actually, the spatial-temporal structures and distributions of the trajectories are different in two videos, which present different motion patterns of two actions. To effectively discriminate them, the relationship between trajectories should be considered so as to capture their spatial-temporal structures and distributions in the videos.

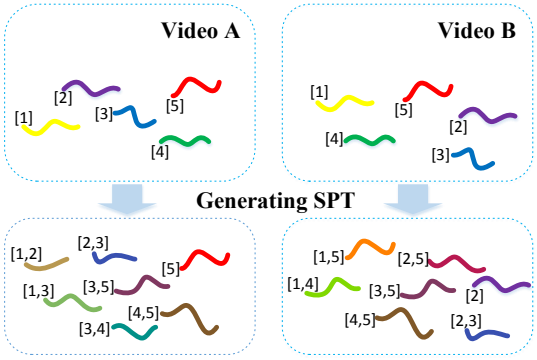


Fig. 1: Illustration of our proposed pairwise trajectory compared to the traditional IDT features. The number [i] represents trajectory T_i, and [i , j] represents the pairwise trajectory generated by T_i and T_j.

Similar problem exists in the popularly used Bag-of-Words (BoW) model in image classification. There have been a number of works that discover the higher-order structure inherence according to the spatial relationship of features to improve the classification performance [11, 12, 13]. Morioka et al. [13] propose Local Pairwise Codebook (LPC), which pairs up the image features within a small region to generate a high-order codebook and has shown its outstanding performance. Inspired by these works, we propose a novel approach to exploit the space-time relationships among video trajectories for action recognition.

In our approach, we employ space-time pairwise trajectories (SPT) to discover the local structures and distributions of trajectories in videos. An example is illustrated in Figure 1. For instance, in videos A and B, the neighboring trajectories of trajectory T_1 are different. This implies different feature structures and motion patterns. Thus, for the description and encoding of the trajectories, we take into account the neighboring trajectories to distinguish them. For trajectory T_1, by combining its neighbors, we generate the

*Corresponding author.

The work described in this paper was supported by the National Natural Science Foundation of China (No. 61103127 and No. 61375016).

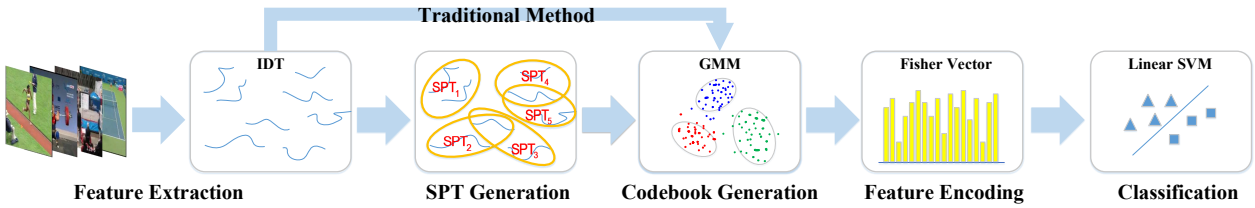


Fig. 2: The pipeline of our approach. Different from the traditional IDT+FV framework, we pair up the trajectories spatially and temporally to formulate pairwise trajectories after trajectory extraction.

trajectory pairs $(T_1, T_2), (T_1, T_3)$ to express T_1 with the local information around it. By exploiting the local structure information around the trajectories, these trajectory pairs have stronger discriminative ability than the single trajectory. In our approach, first we define a trajectory distance to measure the space-time relationship between two trajectories. Second, multiple strategies are proposed to find the optimal trajectory pairs. Finally, we generate the Space-time Pairwise Trajectories (SPT) by taking the mean of the paired trajectories as shown in the lower part of Figure 1.

2. RELATION TO PRIOR WORKS

As discussed in Section 1, the traditional IDT + FV framework encodes the trajectories disorderly, and ignores the local structure of the feature distribution. To solve this problem, a number of works focus on employing the space-time locations and distributions of trajectories for action recognition.

In [14, 15], Spatio-Temporal Pyramid (STP) is proposed to embed the local structure information by dividing a video with fixed grids and pooling the features locally into each grid. This generates much higher dimensions than the original representations. In [16], the Spatial Fisher vector (SFV) [17] is adopted which computes the means and the variances for the spatio-temporal locations of the assigned trajectories to encode the spatio-temporal location information. The feature vectors are extended with the spatio-temporal locations. Compared with STP, similar or slightly better results are achieved with much lower dimensions. In [18], the Space-Time Extended Descriptor (STED) is proposed which extends the feature vectors with the normalized mean of the spatio-temporal locations instead of the mean and the variance. This kind of works directly use the spatio-temporal locations to describe the distribution of trajectories in videos. However, the relationships between different trajectories are not considered.

Inspired by the success in object recognition, mid-level video representation approaches are proposed in action recognition. In [5], graphical models are utilized to represent the relationship between different trajectories. The GANC algorithm [19] is then used to put the trajectories into a number of groups to construct a latent variables model and train the models discriminatively. In order to enhance the discriminative ability of the groups, Ni et al. [20] adopt an optimization algorithm to assign the discriminativeness weights of each trajectory group to achieve trajectory group selection. Atmosukarto et al. [21] model a structured bag of trajectory groups with latent class variables, and learn discriminative trajectory groups by employing multiple instance learning (MIL) based Support Vector Machine (SVM). This kind of approaches group the features according to the similarity between them to form a higher-level representation, and then train a model to capture the relationship between the groups. The performance is heavily dependent on the selection of the groups.

In this paper, we propose a novel approach to discover and describe pairwise trajectories for action recognition. By describing each trajectory with its neighbors, we can capture the local structure information around it. Furthermore, instead of building complex models or higher-level representations, we focus on rebuilding the low-level features. This makes our feature representations more compact and robust.

3. PAIRWISE TRAJECTORY REPRESENTATION FOR ACTION RECOGNITION

The pipeline of our approach is shown in Figure 2. First, we extract improved dense trajectories from the videos. Second, we generate the pairwise trajectories by defining the spatio-temporal distance between trajectories. Third, a GMM codebook is trained and the fisher vector encoding is employed to quantify the pairwise trajectories for the final video representation. Finally, a linear SVM is used for classification.

3.1. Trajectory Distance

To find the neighbors for each trajectory, first we define the distance between trajectories. Let V = $\{T_1, T_2, T_3, \cdots, T_N\}$ be a set of trajectories extracted from a video. Each trajectory T_i can be represented as

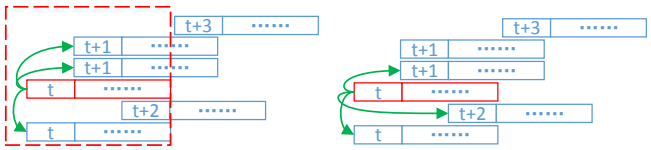
$$T_i = \{\phi_i, P_i, \tau_i\}$$
$$P_i = \{(x_{i1}, y_{i1}), (x_{i2}, y_{i2}), \cdots, (x_{iL}, y_{iL})\}$$

where τ_i is the index of the frame where T_i starts, P_i is the trajectory coordinates, L is the length of the trajectory, and ϕ_i is the appearance/motion description which consists of Trajectory Shape, HOG, HOF, MBHx, and MBHy.

Given two trajectories T_i and T_j, we define the distance between them as

$$\text{dist}(T_i, T_j) = d_s(T_i, T_j) \cdot ln(d_t(T_i, T_j) + e) \quad (1)$$
$$d_s(T_i, T_j) = \|P_i - P_j\|_2 \quad (2)$$
$$d_t(T_i, T_j) = |\tau_i - \tau_j| \quad (3)$$

where d_s, d_t are the spatial and the temporal distances between two trajectories respectively, $\|\cdot\|_2$ denotes the ℓ_2 distance, and the natural constant e is used for smoothness.



(a) (b)

Fig. 3: (a) KNN-based trajectory paring by selecting the k closest trajectories within a fixed temporal distance. (b) Stacked trajectory pairing by selecting the closest trajectory for each temporal distance scale.

3.2. Generation of Pairwise Trajectories

In our approach, we employ pairwise trajectories to capture the local structure information around each trajectory. One problem is how many pairs should be generated for each trajectory. One option is to pair each trajectories with all the others. However, this will produce a series of issues. First, assuming there are N trajectories, there will be $\frac{N(N+1)}{2}$ trajectory pairs which is potentially very large even if the size of N is moderate. Second, two trajectories which are far away from each other are usually unrelated. Several feature selection literatures [11, 12] have shown that most of them do not help to improve the performance of classification and may cause the over-fitting problem. In this section, we propose two strategies to avoid these problems by only pairing trajectories which are spatially and temporally close to each other.

3.2.1. KNN-based Trajectory Pairs

As shown in Figure 3(a), one straightforward way is to pair a trajectory T_i with its k nearest neighbors according to the distance defined in Equation (1). To avoid the extreme case where two trajectories are spatially overlapped ($d_s \to 0$) and temporally far away, we pair up two trajectories T_i and T_j only when their temporal distance $d_t(T_i, T_j) \leq \theta_t$. Similarly, we also filter out the trajectory pairs when $d_s(T_i, T_j) > \theta_s$. For the isolated trajectories which have no neighbors, we put it in the SPT set directly, i.e. we pair it with itself.

3.2.2. Stacked Trajectory Pairs

By pairing up two trajectories with different spatial coordinates, we can capture the spatial structure or distribution of the trajectories. Similarly, pairing up trajectories with different temporal coordinates can help us capture the temporal structure of the trajectories. KNN-based approach cannot guarantee that we capture both spatial and temporal structures of the trajectories. To solve this problem, we propose stacked trajectory pairing.

As illustrated in Figure 3(b), given a trajectory T_i, for each temporal distance $\lambda = 1, 2, \cdots, r$, we select a trajectory T_j such that $|\tau_i - \tau_j| = \lambda$ and $dist(T_i, T_j)$ is minimum. Thus, for each time distance scale λ, we get one trajectory pair for T_i. We filter out the pairs using the distance threshold θ_s as in KNN-based approach. Inspired by multi-scale representation [22], we stack these different pairs as a new pair set. For instance, we get pair sets S_0, S_1, and S_2 when $\lambda = 0, 1, 2$ respectively. The isolated trajectories are put into an unpaired set ψ. Finally, the stacked space-time pairwise trajectories are expressed as $\Phi_2 = \{S_0, S_1, S_2, \psi\}$.

3.3. Representation of Pairwise Trajectories

As described in section 3.1, each trajectory is represented as a tuple of (ϕ_i, P_i, τ_i). To describe the pairwise trajectories generated in Section 3.2, we adopt the average pooling strategy. A pairwise trajectory (T_i, T_j) is represented as

$$\text{SPT}_c = \left(\frac{\phi_i + \phi_j}{2}, \frac{P_i + P_j}{2}, \frac{\tau_i + \tau_j}{2} \right)$$

which has the same format with the standard IDT descriptors. This representation encodes the local spatial-temporal structure information around each trajectory and is more discriminative for action recognition.

4. EXPERIMENTS

4.1. Experimental Settings

For the extraction of improved dense trajectories, we use the default settings in [3]. After trajectory extraction, we find pairwise trajectories using the algorithms proposed in section 3 and set the distance threshold $\theta_s = 250$. For KNN-based trajectory pairing, we set $\theta_t = 5$. PCA (Principle Component Analysis) is used to reduce the dimensionality of Trajectory Shape, HOG, HOF, MBHx, MBHy by a factor of 2. Following the implementation in [3], we apply the square-root trick on all descriptors except for Trajectory Shape.

For Fisher Vector encoding, we randomly sample 256,000 samples to train a GMM with 256 Gaussians. Power ($\alpha = 0.5$) and L2 normalization are used before concatenating different types of descriptors into the final representation. Another Power ($\alpha = 0.5$) and L2 normalization are used after the concatenation. For classification, we use a linear SVM classifier by fixed $C = 100$ as suggested by [3] and the one-versus-all approach is used for multi-class classification.

Table 1: The effects of K size in KNN-based trajectory pairing.

K	Olympic Sports (mAP%)	HMDB51 (mAcc%)	UCF50 (mAcc%)	UCF101 (mAcc%)
1	89.96	59.85	92.12	87.00
2	89.84	59.87	92.38	87.08
3	**91.57**	**60.57**	**92.41**	**87.12**
4	91.23	60.39	-	-
5	91.06	60.46	-	-

Table 2: The effects of S in stacked trajectory pairing.

S	Olympic Sports (mAP%)	HMDB51 (mAcc%)	UCF50 (mAcc%)	UCF101 (mAcc%)
0	90.42	60.22	92.32	86.69
1	**91.44**	**60.72**	**92.58**	**86.98**
2	90.90	60.59	92.55	86.93
3	91.22	60.85	92.50	-
4	89.51	60.46	-	-

Table 3: Comparison of our proposed approaches to the state-of-the-arts.

Olympic Sports	(mAP%)	**HMDB51**	(mAcc%)	**UCF50**	(mAcc%)	**UCF101**	(mAcc%)
Oneata et al [23]	89.0	Oneata et al [23]	54.8	Oneata et al [23]	90.0	Sapienza et al [24]	82.3
Wang & Schmid [3]	91.1	Wang & Schmid [3]	57.2	Wang & Schmid [3]	91.2	Wang et al [25]	85.9
Wang &Oneata [16]	90.4	Wang &Oneata [16]	60.1	Wang &Oneata [16]	91.7	Wang &Oneata [16]	86.0
IDT+FV	89.5	IDT+FV	59.6	IDT+FV	92.3	IDT+FV	86.5
SPT-s (S=1)	**91.4**	SPT-s (S=1)	**60.7**	SPT-s (S=1)	**92.6**	SPT-s (S=1)	**87.0**
SPT-k (K=3)	**91.6**	SPT-k (K=3)	**60.6**	SPT-k (K=3)	**92.4**	SPT-k (K=3)	**87.1**
STED [18]	89.8	STED [18]	62.1	STED [18]	93.0	STED [22]	87.3
SPT-s (S=1)+STED	**89.3**	SPT-s (S=1)+STED	**63.1**	SPT-s (S=1)+STED	**93.3**	SPT-s (S=1)+STED	**87.8**
SPT-k (K=3)+STED	**89.3**	SPT-k (K=3)+STED	**63.3**	SPT-k (K=3)+STED	**93.0**	SPT-k (K=3)+STED	**88.1**

4.2. Datasets

Four typical datasets for action recognition are used in our experiments.

HMDB51 dataset [8]: This dataset has 51 action classes and 6,766 video clips from digitized movies and YouTube. Both original videos and stabilized ones are provided. We only use the original videos in this paper. The standard splits with mean accuracy (mAcc) are used for performance evaluation.

Olympic Sports dataset [7]: This dataset contains 783 video clips of 16 sports action categories from Youtube. We use the test-train splits provided by the dataset and the mAP (mean Average Precision) over all the classes for evaluation.

UCF50 dataset [9]: The UCF50 dataset contains 6,618 clips of 50 action categories. We apply the Leave-One-Group-Out cross-validation (25 cross-validations) as suggested by the authors and report mAcc over all classes.

UCF101 dataset [10]: The UCF101 dataset is extended from UCF50 with 51 additional action categories. In total, there are 13,320 video clips. We follow the evaluation guideline from the THUMOS13 workshop [26] using three train-test splits and report mAcc over the three splits for evaluation.

4.3. Experimental Results

First, for KNN based trajectory pairing approach proposed in Section 3.2.1, we evaluate the effects of different K sizes on the classification performance. The results are shown in Table 1. The highest performance is achieved when $K = 3$ for all datasets. For Olympic Sports and HMDB51 datasets which have relatively sparse trajectories in videos, significant difference is observed when $K = 3$ compared with other sizes. For UCF50 and UCF101 datasets which have relatively dense trajectories, only slight difference is observed. On one hand, when K is too small, the local structure information around a trajectory cannot be fully captured. On the other hand, as discussed in Section 3.2, pairing up one trajectory with too many trajectories does not promise better results since most of them are unrelated and will cause the over-fitting problem especially when the trajectories are sparse in videos. Thus, a moderate k size achieves the best performance.

Second, for the stacked trajectory pairing approach proposed in Section 3.2.2, we evaluate the effects of difference stack scales S on the classification performance in Table 2. When $S = 1$ (i.e. given a trajectory, we select the closest trajectory from the current frame and the next frame), the performance is almost the highest. When we increase S, the performance tends to be stable. Considering the efficiency issue, $S = 1$ is a good choice.

Finally, we compare our approach with several state-of-the-arts approaches in Table 3. According to the results in Tables 1 and 2, we set $K = 3$ for KNN-based trajectory pairing (SPT-k) and $S = 1$ for stacked trajectory pairing (SPT-s). In the upper part of Table 3, the SPT are described by Trajectory Shape, HOG, HOF, MBHx, MBHy. Compared with the traditional IDT + FV and other approaches, our proposed approaches improves the performance for all datasets. This is because the pairwise trajectories contain the local structure information around each trajectory and thus improves the discriminative ability. In the lower part of Table 3, we further extend the representations of the pairwise trajectories with the spatial-temporal location information as in [18]. As can be seen in Table 3, this improves the performances on HMDB51, UCF50, and UCF101 datasets. Compared with STED approach [18], our proposed pairwise trajectory coding approach also performs better for HMDB51, UCF50 and UCF101 datasets. For Olympic Sports dataset, our approach without STED performs the best. By further comparing STED with other approaches, the extended spatio-temporal location information does not help much for this specific dataset. As discussed in [18], in the Olympic Sports dataset, each class only contains 8 testing examples on average, and the improvement may not be statistically meaningful. Overall, the spatio-temporal location information can also be used to describe the pairwise trajectory proposed in this paper.

5. CONCLUSION

We have presented our space-time pairwise trajectory representations for action recognition. By embedding the local structure information into the trajectory representation, the discriminative ability is improved. Both two strategies for trajectory pairing have shown significant performance improvement. The compact pairwise trajectory representation proves to be effective in capturing the local structure information of the trajectory distributions in videos. For future work, we will investigate more effective representations for pairwise trajectories and local structure information embedding.

6. REFERENCES

[1] Jake K Aggarwal and Michael S Ryoo, "Human activity analysis: A review," *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, pp. 16, 2011.

[2] Pavan Turaga, Rama Chellappa, Venkatramana S Subrahmanian, and Octavian Udrea, "Machine recognition of human activities: A survey," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1473–1488, 2008.

[3] Heng Wang and Cordelia Schmid, "Action recognition with improved trajectories," in *IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 3551–3558.

[4] Xiaojiang Peng, Limin Wang, Xingxing Wang, and Yu Qiao, "Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice," *Computer Vision and Image Understanding*, 2016.

[5] Michalis Raptis, Iasonas Kokkinos, and Stefano Soatto, "Discovering discriminative action parts from mid-level video representations," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1242–1249.

[6] Hongyang Li, Jun Chen, Zengmin Xu, Huafeng Chen, and Ruimin Hu, "Multiple instance discriminative dictionary learning for action recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016.

[7] Juan Carlos Niebles, Chih-Wei Chen, and Li Fei-Fei, "Modeling temporal structure of decomposable motion segments for activity classification," in *European Conference on Computer Vision (ECCV)*, pp. 392–405. Springer, 2010.

[8] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre, "Hmdb: a large video database for human motion recognition," in *IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 2556–2563.

[9] Kishore K Reddy and Mubarak Shah, "Recognizing 50 human action categories of web videos," *Machine Vision and Applications*, vol. 24, no. 5, pp. 971–981, 2013.

[10] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, vol. abs/1212.0402, 2012.

[11] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce, "A maximum entropy framework for part-based texture and object recognition," in *IEEE International Conference on Computer Vision*, 2005, vol. 1, pp. 832–838.

[12] David Liu, Gang Hua, Paul Viola, and Tsuhan Chen, "Integrated feature selection and higher-order spatial feature extraction for object categorization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[13] Nobuyuki Morioka and Shinichi Satoh, "Building compact local pairwise codebook with joint feature space clustering," in *European Conference on Computer Vision (ECCV)*, pp. 692–705. Springer, 2010.

[14] Ivan Laptev, Marcin Marszałek, Cordelia Schmid, and Benjamin Rozenfeld, "Learning realistic human actions from movies," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[15] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, vol. 2, pp. 2169–2178.

[16] Heng Wang, Dan Oneata, Jakob Verbeek, and Cordelia Schmid, "A robust and efficient video representation for action recognition," *International Journal of Computer Vision*, pp. 1–20, 2015.

[17] Pyry Matikainen, Martial Hebert, and Rahul Sukthankar, "Representing pairwise spatial and temporal relations for action recognition," in *European Conference on Computer Vision*, 2010, pp. 508–521.

[18] Zhenzhong Lan and Alexander G Hauptmann, "Beyond spatial pyramid matching: Space-time extended descriptor for action recognition," *arXiv preprint arXiv:1510.04565*, 2015.

[19] Seyed Salim Tabatabaei, Mark Coates, and Michael Rabbat, "Ganc: Greedy agglomerative normalized cut for graph clustering," *Pattern Recognition*, vol. 45, no. 2, pp. 831–843, 2012.

[20] Bingbing Ni, Pierre Moulin, Xiaokang Yang, and Shuicheng Yan, "Motion part regularization: Improving action recognition via trajectory selection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3698–3706.

[21] Indriyati Atmosukarto, Narendra Ahuja, and Bernard Ghanem, "Action recognition using discriminative structured trajectory groups," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2015, pp. 899–906.

[22] Zhengzhong Lan, Ming Lin, Xuanchong Li, Alex G Hauptmann, and Bhiksha Raj, "Beyond gaussian pyramid: Multi-skip feature stacking for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 204–212.

[23] Dan Oneata, Jakob Verbeek, and Cordelia Schmid, "Action and event recognition with fisher vectors on a compact feature set," in *IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 1817–1824.

[24] Michael Sapienza, Fabio Cuzzolin, and Philip HS Torr, "Feature sampling and partitioning for visual vocabulary generation on large action classification datasets," *arXiv preprint arXiv:1405.7545*, 2014.

[25] Heng Wang and Cordelia Schmid, "Lear-inria submission for the thumos workshop," in *ICCV workshop on action recognition with a large number of classes*, 2013, vol. 2, p. 8.

[26] Y.-G. Jiang, J. Liu, A. Roshan Zamir, I. Laptev, M. Piccardi, M. Shah, and R. Sukthankar, "THUMOS challenge: Action recognition with a large number of classes," http://crcv.ucf.edu/ICCV13-Action-Workshop, 2013.