CLUSTER ENCODING FOR MODELLING TEMPORAL VARIATION IN VIDEO

Negar Rostamzadeh ⋆, Jasper Uijlings ◇, Ionuţ Mironică †, Mojtaba Khomami Abadi ⋆, Bogdan Ionescu †, Nicu Sebe ⋆

⋆ University of Trento, Italy
◇ CALVIN group, University of Edinburgh, UK
† LAPI, University Politehnica of Bucharest, Romania

ABSTRACT

Classical Bag-of-Words methods represent videos by modeling the variation of local visual descriptors throughout the video. In this approach they mix variation in time and space indiscriminately while these dimensions are fundamentally different. Therefore, in this paper we present a novel method for video representation which explicitly captures temporal variation over time. We do this by first creating frame-based features using standard Bag-of-Words techniques. To model the variation in time over these frame-based features, we introduce Hard and Soft Cluster Encoding, novel techniques to model variation inspired by the Fisher Kernel [1] and VLAD [2]. Results on the Rochester ADL [3] and Blip10k [4] datasets show that our method yields improvements of respectively 6.6% and 7.4% over our baselines. On Blip10k we outperform the state-of-the-art by 3.6% when using only visual features.

***Index Terms*—** modeling temporal variation in video, temporal Fisher Kernel encoding, temporal VLAD encoding, video classification.

1. INTRODUCTION

Videos change over time. They generally consist of different shots which vary wildly in appearance, while within a single shot the changes from frame to frame are more subtle. For automatic video classification, ideally such variation should be modeled. In this paper we propose a novel video representation in which we model the temporal variation in a video.

Currently, there are two main approaches for modeling video: (1) Bag-of-Words models [5, 6, 7] sample spatio-temporal video patches at specific locations in the video, from which local appearance or motion descriptors are extracted. Then techniques such as the Fisher Kernel [1] are used to *model* or *encode* the variation of these descriptors, where temporal and spatial variation is indiscriminately mixed together. This seems suboptimal since spatial and temporal dimensions are fundamentally different; (2) Some works model the temporal *order* within a video by using Hidden Markov Models [8, 9]. However, such models are generally slow at both training and testing time.

In this paper we take another approach. Unlike Bag-of-Words, we want to explicitly model the variation in time. However, instead of modeling temporal *order*, we only model the temporal *variation*. In particular, we first create frame-based features which model the appearance or motion at a specific point in time. Afterwards, we model the variation of these frame-based features, which means the temporal order is lost but we explicitly model the variation in time. The resulting representation is richer than the classical Bag-of-Words approach, while at the same time resulting in a representation which is fast to create and easy to use. We demonstrate the benefits of our framework on two datasets: ADL Rochester [3] for activity recognition and Blip10k [4] for genre classification.

The approach of modelling only the temporal variation was earlier proposed in [10]. In this paper, we improve their work in two important ways: (1) whereas [10] used global frame features (i.e., one frame is described by a single large HoG/HoF descriptor), we use the more powerful Bag-of-Words features to represent a single frame; (2) furthermore, we introduce two novel temporal encoding techniques which are effective in modeling the variation of frame-based Bag-of-Words features.

2. RELATED WORK

The current dominant method of creating video features is the Bag-of-Words method [11, 12]. This approach samples local spatial-temporal video patches on either space-time interest points [5, 13], a regular grid [7, 14], or dense trajectories [6]. From these patches local descriptors are extracted such as Histograms of Oriented Gradients (HoG) [15], Histograms of Optical Flow (HoF) [5, 16], and Motion Boundary Histograms (MBH) [16]. These descriptors are subsequently aggregated into a fixed length representation by counting codewords of a visual vocabulary (e.g., [11, 12, 17]), the Fisher Kernel [1], or VLAD [18]. These approaches all model the variation of the descriptors, but make no distinction between variation in time and variation in space. For videos consisting of multiple, wildly different shots, this is likely suboptimal. In this paper, we use these classical techniques to create

Part of this work was supported under InnoRESEARCH POSDRU/159/1.5/S/132395

frame-based Bag-of-Word features. However, afterwards we perform a separate aggregation step to model the variation of these frame-based features in time. This means our representation explicitly models temporal variation.

Several approaches explicitly model the temporal *order* of frames within a video by using Hidden Markov Models [8, 9, 19]. Revaud, et al. [20] encode frame descriptors jointly in the frequency domain while keeping the temporal order. Other work employs temporal rules with high-level concepts [21]. Such work is usually time consuming, since a pre-temporal-segmentation or a temporal constraint is required to be applied. In this paper we propose to drop the temporal *order* but keep the temporal *variation*. This leads to a simpler yet faster architecture with excellent performance.

3. METHOD

We create video representations in a two-step sequence: (1) we use a standard Bag-of-Words method to create frame-based features. These features model the spatial variation of local descriptors in the video at a specific time; (2) we use standard and new methods for aggregating features within a video in order to model the variation of frame–based features in time.

3.1. Creating Frame-based Visual Features

We use standard methods to create frame-based visual features. As local visual descriptors, we extract densely sampled HoG and HoF features using the software[1] and recommended settings of [7].

We experiment with three types of aggregation methods: hierarchical k–means (HKmeans) plus codebook assignment (e.g., [7, 12]), the Fisher Kernel [1], and VLAD [2], for which we use the VLfeat toolbox[2] [22]. Importantly, we use these methods to create *frame-based* features. That is, in contrast to normal approaches which use Bag-of-Words to represent all descriptors in the video, we create a representation at each point in time for which we have descriptors[3].

For all Bag-of-Words methods we use the recommended settings. We normalize our frame-based Bag-of-Words representations for HKmeans using the square root followed by L_1. For VLAD and FK we apply the square root while keeping the sign followed by L_2.

For future reference, let us represent a set of N videos as $\{V_1, V_2, ..., V_N\}$. We denote the number of frame-based features in a video V_j by η_j. For the m^{th} time–stamp ($m \in \{1, ..., \eta_j\}$) of video V_j, $\phi_{j,m}$ represents the frame-based vocabulary assignment.

[1] http://huppelen.nl/publications/RealtimeHofHogReleaseV1.0.zip

[2] http://www.vlfeat.org

[3] Local descriptors actually span multiple frames but have a time–stamp in the middle of these frames. We only aggregate features with equivalent time-stamps. So the term "frame-based" features is not 100% accurate but it captures the spirit of our work the best.

3.2. Temporal Encoding of Frame-based Features

We want to explicitly capture temporal variation over the frames within a video. We use two classical methods to encode the temporal variation over frame-based features, the Fisher Kernel and VLAD. Then we propose a novel method inspired by these models that outperforms both.

Temporal Fisher Kernel Encoding (TFK). We use the Fisher Kernel [1] to encode the temporal variation over frame-based features. This means that for each video V_j all frame-based features $\phi_j = \bigcup_{m=1}^{\eta_j}\{\phi_{j,m}\}$, are assigned to N_c clusters with a Gaussian Mixture Model (GMM).

Let μ_i and σ_i be the mean and the standard deviation of the i^{th} Gaussian component, $\gamma(i)$ the soft assignment of $\phi_{j,m}$ to Gaussian i, and ω_i the mixture weight of Gaussian i. Let D denote the dimensionality of ϕ_j. Now $G_{\mu,i}^{\phi_j}$ and $G_{\sigma,i}^{\phi_j}$ are respectively the D-dimensional gradient for the mean (μ_i) and standard deviation (σ_i) of Gaussian i. Mathematical derivations [1] lead to:

$$G_{\mu,i}^{\phi_j} = \frac{1}{\eta_j\sqrt{\omega_i}} \sum_{m=1}^{\eta_j} \gamma_m(i) \frac{\phi_{j,m} - \mu_i}{\sigma_i} \quad (1)$$

$$G_{\sigma,i}^{\phi_j} = \frac{1}{\eta_j\sqrt{2\omega_i}} \sum_{m=1}^{\eta_j} \gamma_m(i) \left[\frac{(\phi_{j,m} - \mu_i)^2}{{\sigma_i}^2} - 1 \right] \quad (2)$$

where the divisions between vectors is a term-by-term operation. The final gradient vector G^{ϕ_j} is the concatenation of the $G_{\mu,i}^{\phi_j}$ and $G_{\sigma,i}^{\phi_j}$ vectors, for $i = 1, \ldots, N$ and has a dimensionality of $2DN$. Following [1], we normalize this vector by taking the square root while keeping the sign, followed by L_2.

Temporal VLAD Encoding (TVLAD). VLAD encoding [2] is introduced as a simplified alternative to FK.

In our VLAD temporal encoding, the frame-based features $\phi_j = \bigcup_{m=1}^{\eta_j}\{\phi_{j,m}\}$ for video V_j are assigned to $N_c = \{c_1, c_2, ..., c_n\}$ vocabularies that are obtained by using *K-means*. Features are assigned only to the nearest cluster centre: $\kappa_m(i) = 1$ if the feature belongs to cluster m, and 0 otherwise. This yields:

$$R_{\mu,i}^{\phi_j} = \sum_{m=1}^{\eta_j} \kappa_m(i) \left[\phi_{j,m} - \mu_i\right] \quad (3)$$

The final VLAD vector R^{ϕ_j} is the concatenation of $R_{\mu,i}^{\phi_j}$ for $i = 1, \ldots, N$ and has a dimensionality of DN. We normalize this vector by taking the square root while keeping the sign, followed by L_2, as recommended by [2].

Temporal Hard Cluster Encoding (THC). The VLAD and Fisher Kernel both alter the feature space with respect to the cluster centres. While this has proven to work well on local descriptors, it is unclear if that is good for frame–based features as well. When our frame–based features are created using a codebook assignment, the resulting features are histograms. For such histograms, measuring distances using

Histogram Intersection (or, equivalently, Manhattan distance) is natural and has proven to work well (e.g., [7]). Hence, we propose an alternative encoding which does not alter the original feature space, but which accumulates the features within each cluster. As in VLAD, we use *K-means* to create a visual vocabulary and let $\kappa_m(i)$ denote the hard assignment to a cluster. Let N_m denote the number of assigned features to cluster m. Now we model the average of the features within a cluster as:

$$S_{\mu,i}^{\phi_j} = \frac{1}{N_m} \sum_{m=1}^{\eta_j} \kappa_m(i) \phi_{j,m} \tag{4}$$

We model the standard deviation of these features as:

$$S_{\sigma_m,i}^{\phi_j} = \frac{1}{N_m} \sum_{m=1}^{\eta_j} \kappa_m(i) \left[(\phi_{j,m} - S_{\mu,i}^{\phi_j})^2 \right] \tag{5}$$

As before, the final representation is obtained by concatenation. Note that unlike VLAD and Fisher, these formulas do not use the location of the cluster centre of the original k–means clustering. We normalize using the L_1 norm, as this preserves frame-based features which are based on codebook counts.

Temporal Soft Cluster Encoding (TSC). The assignment of a feature to a single cluster may be quite crude, especially when there are few clusters. Hence, we also propose a soft assignment version of our encoding scheme, as it is also done in the Fisher Kernel and in [17]. For this soft assignment $\gamma_m(i)$, we assume that all clusters created by the k–means clustering algorithm have an isotropic covariance $\sigma = \lambda I_D$, where λ is a parameter we optimize and I_D is the D-dimensional identity matrix. This leads to:

$$T_{\mu,i}^{\phi_j} = \frac{1}{N_m} \sum_{m=1}^{\eta_j} \gamma_m(i) \phi_{j,m} \tag{6}$$

$$T_{\sigma_m,i}^{\phi_j} = \frac{1}{N_m} \sum_{m=1}^{\eta_j} \gamma_m(i) \left[(\phi_{j,m} - T_{\mu,i}^{\phi_j})^2 \right] \tag{7}$$

As before, we normalize using the L_1 norm.

4. EXPERIMENTAL SETUP

We evaluate our temporal encoding schemes on the Rochester ADL [3] and Blip10k [4] datasets. We first describe these datasets and then give details on our experimental setup.

4.1. Datasets

Rochester Activities of the Daily Living (ADL) dataset. This dataset consists of ten complex fine-grained human activities. Each activity is performed three times by five different people, with different ethnicity, appearance and manner of performing the actions. Each clip is in the range of 3-50s. In total the dataset contains 150 videos.

Blip10k dataset. Blip10k contains videos from Blip.tv [4]. The dataset contains 14832 episodes with the running time of 3288 hours. Each video is labeled according to 26 web specific video genre categories, e.g., art, autos and vehicles, business, comedy, etc. The dataset was used for the 2010–2012 MediaEval benchmarking campaigns [4].

4.2. Experimental Setup

Baseline. Our *baseline* uses the simplest temporal encoding of frame-based features. We aggregate by taking the mean and standard deviation of all features in a video. So the baseline models the temporal variation by a single Gaussian distribution. Note that this corresponds to taking a single cluster for our THC and TSC encodings.

Temporal encodings. We model the distribution of our frame-based features using 5 different *Temporal* encodings: (i) *Hard Cluster Encoding (THC)* (ii) *Soft Cluster Encoding (TSC)* (iii) *Fisher Kernel Encoding (TFK)* [1] (iv) *VLAD Encoding* with the $kmeans$-based vocabulary *(TVLAD-K)* [2] and (v) *VLAD Encoding* with the $GMMs$-based vocabulary *(TVLAD-G)* [23].

Optimization of λ. For the TSC encoding, λ is chosen from $\lambda_{set} = \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ to preserve the softness of the representation using an inner-loop cross-validation over the training data. For Blip10k and Rochester ADL we found the optimal values to be respectively $\lambda = 10^{-2}$ and $\lambda = 10^{-4}$.

How many clusters for temporal encoding. This is one of the main questions of this paper. We vary the number of clusters from 1 to 5. Preliminary experiments showed no significant improvements for more clusters, while more clusters significantly increase the dimensionality of our video representations (it scales linearly, but initial dimensionality is already substantial).

5. RESULTS AND DISCUSSIONS

Rochester ADL dataset [3]. We first perform our evaluation on the ADL dataset using leave-one-person-out cross-validation, a standard procedure for this dataset. We have evaluated all the possible combinations of frame-based features, temporal encoding approaches and SVM kernels (Linear, RBF, Histogram Intersection), but given the space limitation, we only present the most relevant results. Figure 1(a) shows a comparison between several temporal encoding methods, where our frame-based features are HOF+HKmeans. Note for the SVM kernel, we selected for each method its best kernel: a linear kernel for VLAD and FK, corresponding to recommendations of [1, 2]; the Histogram Intersection kernel for THC and TSC as anticipated in Section 3.2.

First of all, we observe that all temporal encodings benefit from having multiple clusters. Intuitively, this means that the visual variation in the videos over time is too high to be captured by a single cluster only. However, the classical VLAD

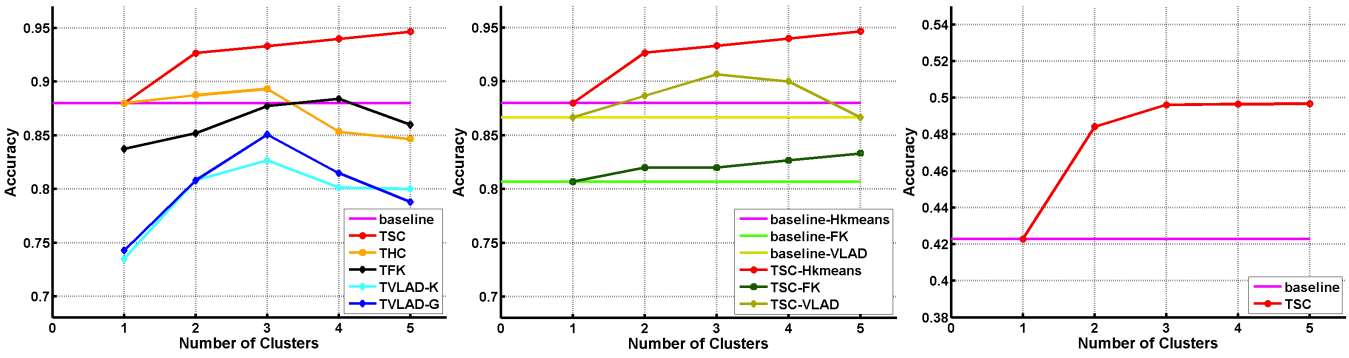

Fig. 1: Experiments on the Rochester ADL dataset: (a) the performance of different encoding approaches with a fixed *BoW* extraction method; (b) the performance when using a fixed encoding method (TSC) and different frame-based representations. The performance using the best pipeline on the Blip10k dataset is shown in (c). All the graphs are shown when the vocabulary size changes from 1 (no temporal variation) to 5 (highest temporal variation).

Table 1: Accuracy results for different approaches stating explicitly the used features. For the Blip10k dataset, V denotes visual features. The results obtained by our approaches are depicted in bold.

Rochester ADL dataset	
Features	Acc.
Our method-HoF	**94.6%**
Our Baseline-HoF	**88.0%**
HoF+HoG [24]	88.6%
HoF+HoG [25]	85.0%
HoF [26]	80.0%
HoF+FG+PoseDets [27]	98.8%
HoF+PoseDets[10]	97.3%
HoF+HoG+ContextFtrs [25]	96.0%
KPT+color+FaceDets [3]	89.0%
HoF+HoG+KPT [28]	82.6%

Blip10k dataset	
Features	Acc.
Our method-HoG (V)	**49.6%**
Our Baseline-HoG (V)	**42.2%**
G-HoG+Color (V) [10]	46.0%
Color+RgbSIFT (V)[4][4]	35.0%
Audio+Video [10]	55.0%
Audio [10]	47.5%
Audio [4][4]	19.2%

and Fisher Kernel encodings do not work well on the frame-based Bag-of-Words features: these are worse than the baseline which models the frame-based features by a single Gaussian distribution. Still, our novel Soft Cluster Encoding yields significant improvements: accuracy goes up from 88.0% to 94.6%, an improvement of 6.6 percentage points in accuracy.

In the next experiment, we keep the best temporal encoding method, i.e., Soft Cluster Encoding, but instead change the frame-based features. Here we create frame-based features not only using HKMeans encoding, but also using VLAD and Fisher Kernels. The results in Figure 1(b) show that for all frame-based features the temporal encoding improves the accuracy by 3-7%. We can also note that HKmeans are the best frame-based features.

We also compare our results with the state-of-the-art in Table 1. The best results are obtained by methods which use complex features such as Body-Parts [10, 27] or by a method which models contextual interaction between interest points [25]. However, if we compare our results to approaches relying only on fast local visual descriptors, we obtain significantly better results: the best method [24] has 88.6% using both HoG and HoF, similar to our baseline. In contrast, we obtain 94.6% accuracy using only HoF. This shows that our explicit coding of temporal variation is very effective.

Blip10k [4]. Since the Blip10k dataset is huge, we only evaluate the framework that gave the best results on the Rochester ADL dataset. We replaced however the HoF descriptors with HoG descriptors to reduce the computation time. Consequently, as frame-based features we have a BoWs representation using HoG descriptors modeled by a HK-means codebook. We model the temporal variation using Soft Cluster Encoding.

The results are calculated by mean Average Precision (mAP) which is a standard evaluation metric on Information retrieval tasks including the Blip10k genre retrieval task. Results are presented in Figure 1(c). As before, the results go up drastically by properly modeling the temporal variation. Results improve from 42.2% to 49.6%, an increase of 7.4 percentage points in mean Average Precision.

Table 1 shows the comparison with the state-of-the-art. Here, the best results are obtained by [10] while using a combination of visual and audio features. However, for visual features only, we outperform [10] by 3.6%, even if we use fewer visual features. We conclude that our explicit modeling of temporal variation in video is very effective.

6. CONCLUSION

We presented a framework in which we explicitly model variation in time in video. First we create frame-based features based on Bag-of-Words. For modeling their variation in time (but not their order) we introduced Hard and Soft Cluster Encoding, novel encoding techniques inspired by the Fisher Kernel and VLAD. Results show significant improvements of respectively 6.6% and 7.4% accuracy over our baselines. Furthermore, comparing our results on the Rochester ADL dataset to other articles which use only local visual HoG and HoF descriptors, we show accuracy improvements of 6%. On Blip10k, we outperform the state-of-the-art when using only visual features by 3.6%.

[4]The results presented for [4] are the best results reported in the MediaEval competition.

7. REFERENCES

[1] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher kernel for large-scale image classification," in *ECCV*, 2010.

[2] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *CVPR*, 2010.

[3] R. Messing, C. Pal, and H. Kautz, "Activity recognition using the velocity histories of tracked keypoints," in *CVPR*, 2009.

[4] S. Schmiedeke, P. Xu, I. Ferrané, M. Eskevich, C. Kofler, M. Larson, Y. Estève, L. Lamel, G. Jones, and T. Sikora, "Blip10000: A social video dataset containing spug content for tagging and retrieval," in *ACM Multimedia Systems*, 2013.

[5] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *CVPR*, 2008.

[6] H. Wang, A. Kläser, C. Schmid, and C. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision*, vol. 103, pp. 60–79, 2013.

[7] J. Uijlings, I. Duta, N. Rostamzadeh, and N. Sebe, "Realtime video classification using dense HOF/HOG," in *ICMR*, 2014.

[8] H. Kuehne, D. Gehrig, T. Schultz, and R. Stiefelhagen, "Online action recognition from sparse feature flow," in *VISAPP*, 2012.

[9] G-J. Qi, X-S. Hua, Y. Rui, J. Tang, T. Mei, M. Wang, and H-J. Zhang, "Correlative multilabel video annotation with temporal kernels," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 5, no. 1, 2008.

[10] I. Mironica, J. Uijlings, N. Rostamzadeh, B. Ionescu, and N. Sebe, "Time matters!: Capturing variation in time in video using fisher kernels," in *ACM Multimedia*, 2013.

[11] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *ICCV*, 2003.

[12] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Int. Workshop on Statistical Learning in Computer Vision*, 2004.

[13] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *VS-PETS*, 2005.

[14] H. Wang, M. Ullah, A. Kläser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *BMVC*, 2009.

[15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.

[16] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *ECCV*, 2006.

[17] J. van Gemert, J-M. Geusebroek, C. Veenman, and A. Smeulders, "Kernel codebooks for scene categorization," in *ECCV*, 2008.

[18] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *Pattern Analysis and Machine Intelligence, PAMI*, vol. 34, no. 9, pp. 1704–1716, 2012.

[19] C. Snoek and M. Worring, "Concept-based video retrieval," *Foundations and Trends in Information Retrieval*, 2008.

[20] J. Revaud, M. Douze, C. Schmid, and H. Jégou, "Event retrieval in large video collections with circulant temporal encoding," in *CVPR*, 2013.

[21] K-H. Liu, M-F. Weng, C-Y. Tseng, Y-Y. Chuang, and M-S. Chen, "Association and temporal rule mining for post-filtering of semantic concept detection in video," *IEEE Transactions on Multimedia*, vol. 10, no. 2, pp. 240–251, 2008.

[22] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," 2008.

[23] D. Picard and P-H. Gosselin, "Improving image similarity with vectors of locally aggregated tensors," in *ICIP*, 2011.

[24] P. Banerjee and R. Nevatia, "Pose filter based hidden-crf models for activity detection," in *ECCV*, 2014.

[25] J. Wang, Z. Chen, and Y. Wu, "Action recognition with multiscale spatio-temporal contexts," in *CVPR*, 2011.

[26] S. Satkin and M. Hebert, "Modeling the temporal extent of actions," in *ECCV*, 2010.

[27] N. Rostamzadeh, G. Zen, I. Mironică, J. Uijlings, and N. Sebe, "Daily living activities recognition via efficient high and low level cues combination and fisher kernel representation," in *ICIAP*, 2013.

[28] M. Raptis and S. Soatto, "Tracklet descriptors for action modeling and video analysis," in *ECCV*, 2010.