An end-to-end generative framework for video segmentation and recognition

Hilde Kuehne
University of Bonn
kuehne@iai.uni-bonn.de

Juergen Gall
University of Bonn
gall@iai.uni-bonn.de

Thomas Serre
Brown University
thomas_serre@brown.edu

Abstract

We describe an end-to-end generative approach for the segmentation and recognition of human activities. In this approach, a visual representation based on reduced Fisher Vectors is combined with a structured temporal model for recognition. We show that the statistical properties of Fisher Vectors make them an especially suitable front-end for generative models such as Gaussian mixtures. The system is evaluated for both the recognition of complex activities as well as their parsing into action units. Using a variety of video datasets ranging from human cooking activities to animal behaviors, our experiments demonstrate that the resulting architecture outperforms state-of-the-art approaches for larger datasets, i.e. *when sufficient amount of data is available for training structured generative models.*

1. Introduction

The growing need for automated video monitoring and surveillance systems is quickly reshaping our research landscape. Much of the current research on action recognition has focused on semi-realistic problems such as categorizing short clips consisting of one single action (*e.g.* kick, pour, throw, pick). However, many real-world applications will require methods that can solve more realistic problems including the recognition and parsing of complex activities in long continuous recordings, often consisting of sequences of goals and sub-goals.

Most successful approaches to action recognition have typically relied on unstructured models of video sequences. A holistic visual representation is usually computed over an entire video clip and then passed to a discriminative classifier to yield a single categorization label per video. These methods have been successful for the recognition of single-action video clips (see *e.g.* [34]). However, they do not appear to be well suited for the recognition of daily activities that require the modeling of complex behavior sequences.

Several extensions of these unstructured models have been proposed to try to address this challenge. One popular approach relies on sliding (temporal) windows whereby videos are decomposed into a sequence of shorter segments that can be individually classified with discriminative approaches [23, 5, 1]. However, these approaches have, for the most part, only been tested on a handful of relatively small datasets that do not capture the rich and diverse nature of daily activities. As we will show, these approaches are not competitive on more challenging activity datasets.

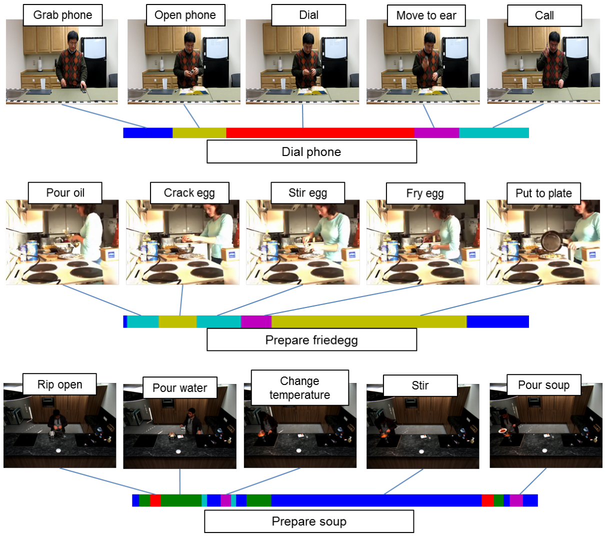

Figure 1: Segmentation and recognition of human activities with a) the ADL dataset ("dial phone"), b) the Breakfast dataset ("prepare fried eggs") and c) the MPII cooking dataset ("prepare soup").

Structured temporal models, on the other hand, have reached an impressive level of maturity in several engineering domains and speech recognition [36] in particular. These models would appear more appropriate than their unstructured counterparts for the recognition of human activities. Somewhat surprisingly, relatively little effort has been devoted to adapting these approaches to human action recognition (but see *e.g.* [4, 12]). One of the main reasons why structured generative methods have not found more widespread acceptance in action recognition is that, unlike for speech analysis where large annotated corpora are available, video databases have been comparatively limited in size [12].

With the emergence of larger video datasets (*e.g.* CRIM13 [2] and Breakfast [12]), these models are more likely to start exhibiting competitive performance. For instance, encouraging results were obtained in [12] using Hidden Markov Models (HMMs) combined with a context-free grammar to learn cooking activities. One of the main limitations associated with standard HMM toolboxes (such as the HTK used in [12]) is the use of Gaussian mixtures, which typically require input data to be normally-distributed as well as low-dimensional to prevent overfitting. Standard visual representations such as Bag-of-Words or Fisher Vectors (FVs) thus constitute a poor choice for HMMs and other generative approaches, because they typically yield sparse and high-dimensional visual representations.

Here, we describe an approach for the construction of reduced FVs which is particularly amenable to structured temporal models. FVs have been shown to achieve state-of-the-art accuracy in action recognition [18]. They have also been shown to maintain good classification accuracy when used in conjunction with dimension reduction techniques [6, 11]. Hence, this makes them good candidates for modeling by Gaussian mixtures. As we will show, the proposed approach yields a very substantial improvement in recognition accuracy on a variety of activity segmentation and recognition tasks, ranging from the recognition of human daily activities to the segmentation of rodent social interactions.

To summarize, we describe an approach to improve the efficiency of state-of-the-art feature encoding methods [6, 11] that are especially amenable to generative models. We systematically evaluate the proposed approach using a variety of standard activity datasets and demonstrate significant improvements for datasets that contain sufficient training data.

2. Related work

2.1. Fisher vectors

Fisher kernel methods were originally proposed as a way to derive kernels for discriminative classifiers from generative models [9]. They were later adapted to represent feature sets used for image classification [19]. The application of an L_2 norm and power normalizations combined with a method for sampling FVs based on a spatial pyramid were then shown to significantly improve their accuracy [20]. More recently, FVs have been shown to yield not only higher classification accuracy, but also much more compact feature vectors [11].

The application of FVs to action recognition was first explored in [35], where the authors used a standard video descriptor (HOGHOF) to compare different encoding methods on two different datasets. It was shown that FVs often outperform other methods, a result that was further replicated in a separate study [31]. The combination of FVs and Dense Trajectory Features (DTFs) was also demonstrated to work exceedingly well for the recognition of actions [34, 17]. All the aforementioned approaches are based on discriminative classification methods trained on (short) single-action pre-segmented video clips. We are not aware of previous work focusing on the statistical properties of FVs in the context of a generative action recognition models.

2.2. Structured temporal models

Most early approaches for action recognition with structured temporal models relied on either motion capture data [8, 28] or hand-labeled trajectories [22]. Several temporally structured models have been applied since on video data including generative mixture models [14], Bayes Networks [25] and an HMM/SVM combination [4].

More recent work has focused on the problem of detecting and segmenting human activities in videos. In [27], a semantic scene label map was built as context for agent actions to automatically learn AND-OR grammars from videos. In [1], Linear Dynamical Systems theory was used to detect events in complex video datasets. Long-term relations were also considered in the “sequence memorizer” described in [5], which uses a Bayesian nonparametric model to simultaneously detect and classify events within a video stream. A similar idea is proposed in the work of [12], using a context free grammar in combination with HMMs to model longer temporal sequences of smaller action units. In [7], activity models were based on the detection of changes in state-specific regions of interest (*e.g.* the lid of a coffee jar for ’opening coffee jar’ and ’closing coffee jar’ actions). The authors used SVM-based state detectors to detect the beginning and end of short task-oriented action units such as “hold spoon” or “stir coffee”.

A higher-level representation based on stochastic context-free grammars was used in [32] where body pose information (*i.e.* hand positions) was used for classification. A closely related approach was proposed in [21] where action units were combined with a set of production rules to build a grammar to model the hierarchical temporal structure of human activities. The system was able to learn and parse action units derived from the Olympic sport dataset.

Here, we build on our earlier work [12] using HMMs combined with a simple grammar to model complex human activities as sequences of action units.

3. System description

3.1. Fisher vectors

We briefly review the key steps involved in FV computation and frame-based action recognition. We refer the reader to [26] for a more detailed description. The main assumption behind FVs is that local feature descriptors may be modeled by a probability density function. Here, we consider a Gaussian mixture Model (GMM) with K components defined by the associated mixture weights, mean vectors μ_k and variances σ_k. FVs characterize how a feature set X $= \{x_t | t = 1, \ldots, T\}$ deviates from a learned distribution. For each feature set X, the resulting gradients $\mathcal{G}^{\mathrm{x}}_{\mu_k}$ and $\mathcal{G}^{\mathrm{x}}_{\sigma_k}$ each have the dimensionality D of the original feature descriptor and they are computed for each mixture of the GMM as described in [6].

The concatenation leads to an overall $2 \times D \times K$ dimensional FV representation $\hat{x}$ of the original feature set X with $\hat{x} = [\mathcal{G}^{\mathrm{x}}_{\mu,k}, \mathcal{G}^{\mathrm{x}}_{\sigma,k}]'$. Following [20], we applied an L2-normalization to these vectors. Additionally, the authors in [20] observed that the more Gaussian components are used, the sparser the FVs become. We followed their suggestion to use a power normalization ($g(\hat{x}) = sign(\hat{x})\sqrt{\hat{x}}$) to reduce the sparsity of the FVs. As the resulting FVs are too high dimensional to be processed in a generative framework, we used PCA to reduce the overall dimensionality of the feature vector [11] and to further whiten the data.

3.2. Normality test

The HTK recognition framework used here (see section 3.3), like most other systems for automated speech recognition, relies on HMMs with observation probabilities modeled by Gaussian mixtures. Higher dimensional Gaussian mixtures are prone to overfitting, especially when given only a limited amount of training data. This can be compensated to a certain extent by reducing the number of mixtures used. In general, we found that best results were obtained with one Gaussian per state which is consistent with the practice reported in [12]. It is thus highly desirable for input data to be normally distributed.

In order to test the normality of FVs for video data, we considered different normality tests. To evaluate how dimensionality reduction using PCA affects the normality of the resulting feature vector, we randomly sampled data along each dimension of the feature vectors and test the skewness and kurtosis of the resulting distributions using the Lilliefors [13] and the Jarque-Bera test [10], respectively. We tested the null hypothesis that a given dimension is normally distributed and estimated the number of dimensions for which the null hypothesis is valid (for decreasing significance levels in the range 0.5–0.001). We applied this test to FV samples before and after PCA. Results shown in Figure 2 confirm that PCA yields distributions that are closer to a normal distribution. For instance, at a significance level of $\alpha = 0.001$, a mere 0.53% of the original FV dimensions pass the Lillifors test (none for Jaque-Bera), whereas 84.3% (79.6% Jaque-Bera) of the PCA-reduced data dimensions pass significance. This is quite evident already when we consider the first dimension of the feature vector (before and after PCA), as shown in Figure 2. For comparison, we also tested the BoWs as used in our previous work [12]. Here, the null hypothesis was always rejected, irrespective of the significance level, suggesting that none of the dimensions are normally distributed.

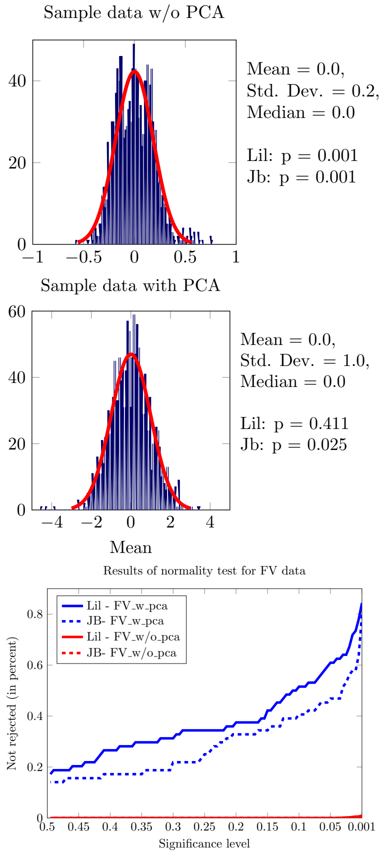

Figure 2: Distribution of FV samples before and after PCA and results of normality test (Lil = Lilliefors, Jb = Jarque-Bera) with decreasing significance levels for FV samples before and after PCA.

Overall, PCA helps to build a feature vector that better fit the normality assumption of the proposed HMM-based model. As we will show in Section 4.2, this yields significant gains in activity recognition accuracy.

3.3. A generative recognition pipeline

In the following, we briefly give an overview of the pipeline used (see Figure 3). We used an improved version [34] of the Dense Trajectory Features (DTFs) [33] for datasets with camera motion. The dimensionality of the feature descriptors was first reduced from 426 dimensions to 64 dimensions by PCA, following the procedure described in [17].

We sampled 200,000 random features to fit the GMMs. FVs were computed using 50,000 frames sampled from the

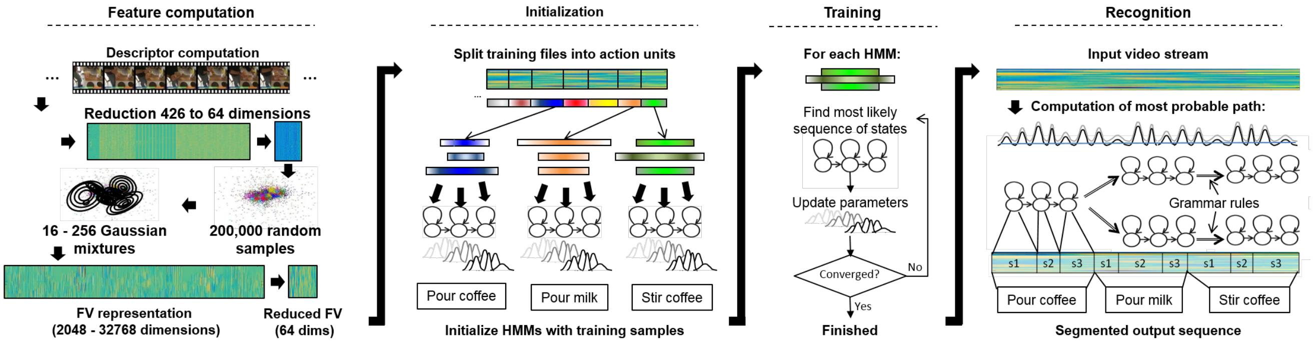
Feature computation
Descriptor computation
Reduction 426 to 64 dimensions
16 - 256 Gaussian mixtures
200,000 random samples
FV representation (2048 - 32768 dimensions)
Reduced FV (64 dims)
Initialization
Split training files into action units
Pour coffee
Pour milk
Stir coffee
Initialize HMMs with training samples
Training
For each HMM:
Find most likely sequence of states
Update parameters
Converged?
No
Yes
Finished
Recognition
Input video stream
Computation of most probable path:
Grammar rules
s1
s2
s3
Pour coffee
Pour milk
Stir coffee
Segmented output sequence

Figure 3: Overview of the recognition pipeline: DT features are computed and the corresponding descriptor is reduced to 64 dimensions. A total of 200,000 features are randomly sampled and fitted to GMMs ($K = 16, 32, 64, 128$ or 256). An FV representation is computed for each frame of the video. The corresponding representation is further reduced from 2048 – 32,768 down to 64 dimensions. During training, each HMM is initialized with action unit samples. State boundaries are re-estimated and the GMMs are updated according to the new state boundaries until convergence. During recognition, HMMs are combined with a learned context-free grammar and the most probable sequence of action units is determined.

training data. For each reference frame, FVs were computed over a 20-frames sliding window. The dimensionality of the resulting vector was further reduced to 64 dimensions using PCA (see section 3.1). Thus, each frame is then represented by a 64-dimensional FV. We further applied an L2-normalization to each feature dimension separately for each video clip.

The proposed recognition system contains two main components: a set of HMMs is used to model all possible action units found in the dataset and a grammar is used to model possible sequences of those units. The number of hidden states for each HMM was set to $1/10$ of the mean length of the corresponding action units. All HMMs are based on a left-to-right feed-forward topology, allowing only self-transitions and transitions to the next state. The initial state transitions probabilities were set to default values (self: $p = 0.6$, next $p = 0.4$). To initialize the state distribution, we subdivided each action unit evenly over time and associated each subdivision to a hidden state. Thus, frames at the beginning or end of an action unit get always associated to the first and last states due to the left-to-right topology.

During training, unit states were re-estimated using the Baum-Welch algorithm, *i.e.* by finding the HMM parameters that maximise the probability of a given set of observations. For details concerning the training and recognition with HTK, we refer the reader to to [36, 12] for details. As the number of samples per class or in our case, per action unit follows a long-tail distribution, with few classes being frequent and a large number of classes being relatively rare, we enforced a minimum and maximum number of training samples (see Table 1) for a balanced training data set across classes. When needed, artificial samples were generated by synthetic minority over-sampling to guarantee a minimum number of samples.

During recognition, we followed the approach described in by [12] formalizing activity recognition and segmentation as the problem of finding the most probable sequence of action units from an observed input sequence. A context free grammar was built automatically using available annotations. For the CRIM13 dataset [2]), we favored a bi-gram model which defines the transition probability to the next possible units instead of absolute paths. This is a richer model and it is more appropriate for modeling animal behavior which tends to be relatively stochastic compared to the human activities found in other datasets.

The Viterbi algorithm was used to find the most probable sequence of action units. The output of the algorithm includes the best matching sequence of action units, their beginning and end frames, and the corresponding observation probabilities (see [36]).

4. Evaluation

4.1. Datasets

Recent years have seen a significant increase in the availability of public activity datasets. To evaluate the proposed architecture, we considered complex activity datasets (as opposed to single task-oriented action) that are labeled at one or more levels of granularity. The datasets found suitable for this evaluation included: ADL [14], Olympics [16], ToyAssembly [32], CMU-MMAC [29], MPIICooking [23], 50Salads [30], Breakfast [12], and CRIM13 [2]. Sample frames for each of these datasets are shown in Figure 4.

The recognition tasks for these datasets typically include activity classification, action unit detection and segmenta-

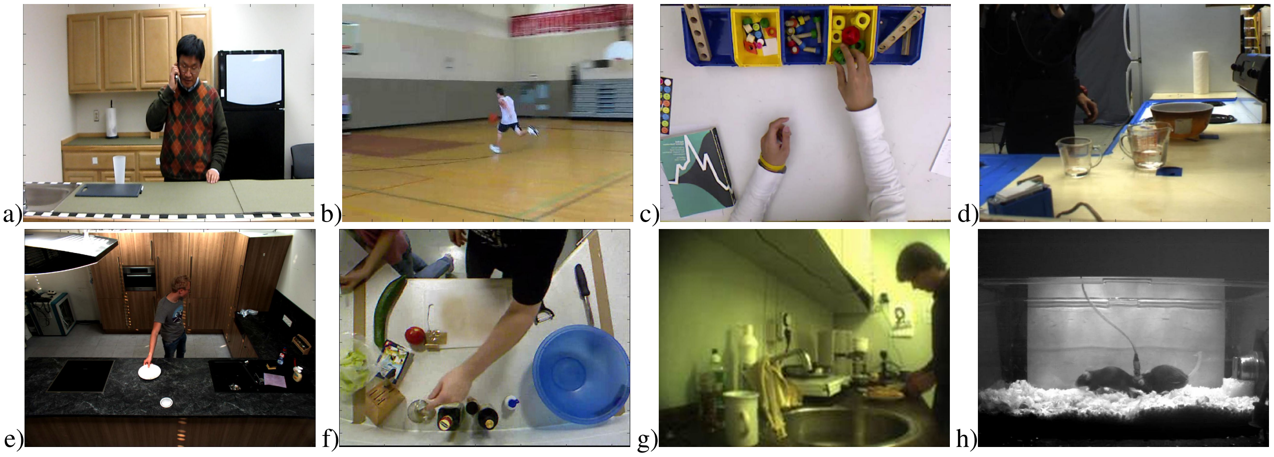

Figure 4: Sample frames from the datasets used for performance evaluation: a) ADL [14], b) Olympic [16], c) ToyAssembly [32], d) CMU-MMAC [29], e) MPIICooking [23], f) 50Salads [30], g) Breakfast [12], and h) CRIM13 [2].

	Duration	Train samples used per class
ADL	40 min	12-30 samples
Olympics	90 min	70-80 samples
Toy	64 min	15-20 samples
CMU	265 min	30-40 samples
MPII	490 min	12-30 samples
50Salads	320 min	30-35 samples
BF	66.7 h	50-70 samples
CRIM13	32.4 h	80-100 samples

Table 1: Overall duration of the different datasets and number of samples available for training.

Breakfast dataset - FV						
	GMMs =	16	32	64	128	256
1) SVM+DTF w/o PCA		52.0	52.6	48.7	39.6	23.2
2) SVM+DTF w PCA	$D' = 64$	42.0	42.5	42.8	40.3	41.2
3) HTK+HOGHOF w PCA	$D' = 64$	62.3	61.1	62.2	60.7	60.2
4) HTK+DTF w PCA	$D' = 64$	**73.2**	**74.8**	**75.4**	**70.2**	**67.9**

Table 2: Comparison between HTK vs. SVM and HOGHOF vs. DTFs for activity recognition (in combination with FV-based encoding on the Breakfast dataset).

tion. The only exception is the Olympic Sport dataset, where no action unit labeling exists. For this dataset, we manually labeled 10 clips per class and used these annotations for initializing the system. We then applied the recognition scheme to the remaining training clips and used the system outputs as labels for the training phase.

Some of the selected datasets provide additional benefits such as multi-modal signals or multi-view settings. For this evaluation however, we only considered video data. All videos were separately processed and evaluated and we did not apply any method for combining camera input from different views. The duration of the datasets and the number of samples used for training is shown in Table 1.

4.2. System evaluation

We first compare the accuracy of the proposed reduced FVs against that of our previous work using HTK in combination with HOGHOF for the Breakfast dataset (Table 2). Replacing HOGHOF with DTFs already improves the overall system accuracy by $\sim 10 - 14\%$ (Table 2, HTK+HOGHOF w PCA compared to HTK+DTF w PCA).

To evaluate the impact of the reduced FVs on a generative vs. a discriminative framework, we compared the proposed pipeline against one where the HTK classification stage was replaced with an SVM (for both the full FV representation with 2,048–32,768 dimensions and K= 16–256 GMMs and the reduced FV representation with 64 dimensions). Classification was based on the libSVM software library [3] using a linear kernel. We used identical features and GMM clusters as in the proposed HTK-based system. Note, however, that for the SVM baseline, features were sampled from the entire video sequence because we found it to work better than a frame-based sampling as used for HTK.

As Table 2 shows, SVM-based classification performs better when using the full FV representation for classification compared to the reduced FV representation. However, the accuracy of the SVM-based classification remains significantly below the accuracy of the system based on HTK by $\sim 20 - 30\%$ with identical features. Our results show that, compared to the baseline reported in [12], reduced FVs improve the recognition accuracy by $\sim 20\%$ for HOGHOF and $\sim 30\%$ for DT.

Segmentation								
GMM=	ADL	Oly.	Toy	CMU	MPII	50Salad	BF	CRIM13
16	53.4	62.4	50.3 / *64.3*	53.8 / *60.8*	46.5 / *58.5*	81.6	36.2 / *54.2*	52.6
32	54.5	66.1	48.6 / *63.1*	53.7 / *60.7*	53.9 / *68.5*	80.4	36.9 / *54.4*	**53.5**
64	55.7	67.5	56.7 / *67.5*	53.0 / *60.3*	51.6 / *63.9*	**83.8**	**38.1 / *56.3***	53.4
128	58.9	65.9	60.5 / *70.8*	52.5 / *60.4*	53.9 / *66.8*	82.0	34.0 / *51.2*	52.6
256	54.4	63.7	63.5 / *72.2*	**58.8 / *67.1***	**57.3 / *71.7***	**83.8**	32.7 / *50.7*	53.3
Best	–	–	– / ***91.0*** [32]	– / *59.0* [32]	– / *54.3* [15]	67.6 [30]	– / *28.8* [12]	39.1 [2]

Table 3: Overview of the segmentation results for all datasets. Accuracy is computed as the mean over all classes. For comparison, we also report the frame-based accuracy (*italic*) for the Toy, CMU and BF dataset, and midpoint hit accuracy (*also italic*) for the MPII dataset as used by the authors in the original studies.

4.3. Segmentation

To evaluate the segmentation accuracy of the proposed system, we consider eight different datasets (see section 4.1 for details). As the original benchmarks for these datasets are based on different measurements, we report multiple accuracy measures for fair comparison to these baseline systems. One measure reported uses the mean accuracy over all classes (corresponding to the mean accuracy computed over the diagonal of the corresponding confidence matrix) as used in [23, 30, 2]. In addition, we also report the frame-based accuracy (corresponding to the mean proportion of correctly classified frames) for the Toy, CMU and Breakfast dataset as used in [32, 12]. For the MPII dataset, we also report the mid-point hit accuracy as defined in [23].

Segmentation results for the proposed system and available benchmarks are reported in Table 3. Note that for the two smallest datasets (ADL and Olympics), no benchmark is available as no segmentation results have been previously reported for these datasets. It is pretty clear that the proposed approach under-performs the best segmentation results obtained for the Toy assembly dataset (which remains a small video dataset with about one hour of video). For large datasets (8 hours or more of video), the system significantly outperforms the state of the art in terms of segmentation accuracy (*e.g.* BF +27.5%). Note that for the CRIM13 dataset, the benchmark approach is based on spatial-temporal features [2]. For this evaluation, we only considered side-view videos and report the accuracy of the benchmark system for the same set of videos as reported in the original study. Sample segmentation results are shown in Figure 5 for the ADL, MPII and Breakfast datasets.

4.4. Activity classification

Here, we evaluated the accuracy of the proposed system for activity classification (Table 4). We only considered datasets that provide multiple activity classes (*i.e.* ADL, Olympic and Breakfast datasets). Consistent with earlier experiments, the accuracy of the proposed system is below the state of the art for smaller datasets (*e.g.* ADL and Olympic Sports) but outperforms the state of the art when enough training samples are available (*e.g.* Breakfast dataset).

Activity classification			
GMM=	ADL	Olympics	BF
16	86.0	74.4	73.1
32	86.7	76.8	74.8
64	91.3	77.6	**75.4**
128	94.7	77.2	70.2
256	87.3	74.4	67.9
Best	**98.7** [24]	**90.2** [34]	40.5 [12]

Table 4: Activity classification results.

5. Conclusion

In this paper, we studied how different feature representations affect the performance of a structured generative (temporal) model based on the HTK framework. We performed a systematic evaluation of the proposed approach and compared the accuracy of the resulting system against the state of the art for both activity segmentation and classification. Our results showed that combining a compact video representation based on Fisher Vectors with Hidden Markov Models yields very significant gains in accuracy for both the recognition of goal-oriented activities and their parsing at the level of task-oriented action units. Indeed, when sufficient training data was available, we found that structured generative temporal models outperform the state of the art. These results are consistent with recent trends in other areas of computer vision suggesting that, as datasets are becoming increasingly large, structured models are starting to outperform the state of the art.

6. Acknowledgment

This work was supported by the DFG Emmy Noether program (GA 1927/1-1) to HK and by DARPA young faculty award (N66001-14-1-4037) and NSF early career award (IIS-1252951) to TS.

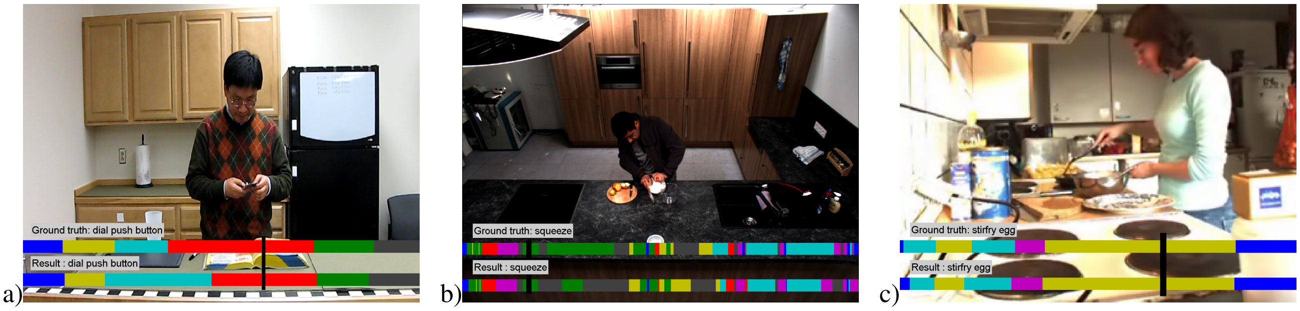
a)
Ground truth: dial push button
Result : dial push button
b)
Ground truth: squeeze
Result : squeeze
c)
Ground truth: stirfry egg
Result : stirfry egg

Figure 5: Sample segmentation results for a) the ADL dataset ("dial phone"), b) the MPII cooking dataset ("prepare cold drink"), and c) the Breakfast dataset ("prepare scrambled eggs"). The upper/lower color bars correspond to ground-truth/system outputs, respectively.

References

[1] S. Bhattacharya, M. Kalayeh, R. Sukthankar, and M. Shah. Recognition of complex events: Exploiting temporal dynamics between underlying concepts. In *CVPR*, 2014. 1, 2

[2] X. Burgos-Artizzu, P. Dollár, D. Lin, D. Anderson, and P. Perona. Social behavior recognition in continuous videos. In *CVPR*, 2012. 2, 4, 5, 6

[3] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Trans. on Intelligent Systems and Technology (TIST)*, 2(3):27 – 27, 2011. 5

[4] C. Chen and J. Aggarwal. Modeling human activities as speech. In *CVPR*, pages 3425–3432, 2011. 1, 2

[5] Y. Cheng, Q. Fan, S. Pankanti, and A. Choudhary. Temporal sequence modeling for video event detection. In *CVPR*, 2014. 1, 2

[6] G. Csurka and F. Perronnin. Fisher vectors: Beyond bag-of-visual-words image representations. In *Computer Vision, Imaging and Computer Graphics. Theory and Applications*, volume 229, pages 28–42. Springer, 2011. 2, 3

[7] A. Fathi and J. Rehg. Modeling actions through state changes. In *CVPR*, 2013. 2

[8] G. Guerra-Filho, C. Fermüller, and Y. Aloimonos. Discovering a language for human activity. In *AAAI Symposium on Anticipatory Cognitive Embodied Systems*, 2005. 2

[9] T. S. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *NIPS*, pages 487–493, 1998. 2

[10] M. Jarque, A. K. Bera, C. M. Jarque, and A. K. Bera. A test for normality of observations and regression residuals. *Internat. Statist. Rev*, pages 163–172, 1987. 3

[11] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. *PAMI*, 34(9):1704–1716, Sept 2012. 2, 3

[12] H. Kuehne, A. Arslan, and T. Serre. The Language of Actions: Recovering the Syntax and Semantics of Goal-Directed Human Activities. In *CVPR*, 2014. 1, 2, 3, 4, 5, 6

[13] H. W. Lilliefors. On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, 62(318):399–402, Jun 1967. 3

[14] R. Messing, C. Pal, and H. Kautz. Activity recognition using the velocity histories of tracked keypoints. In *CVPR*, 2009. 2, 4, 5

[15] B. Ni, V. Paramathayalan, and P. Moulin. Multiple granularity analysis for fine-grained action detection. In *CVPR*, 2014. 6

[16] J. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*. 2010. 4, 5

[17] D. Oneata, J. Verbeek, and C. Schmid. Action and Event Recognition with Fisher Vectors on a Compact Feature Set. In *ICCV*, 2013. 2, 3

[18] X. Peng, C. Zou, Y. Qiao, and Q. Peng. Action recognition with stacked fisher vectors. In *ECCV*, 2014. 2

[19] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007. 2

[20] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, 2010. 2, 3

[21] H. Pirsiavash and D. Ramanan. Parsing videos of actions with segmental grammars. In *CVPR*, 2014. 2

[22] C. Rao, A. Yilmaz, and M. Shah. View-Invariant Representation and Recognition of Actions. *IJCV*, 50(2):203–226, 2002. 2

[23] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *CVPR*, 2012. 1, 4, 5, 6

[24] N. Rostamzadeh, G. Zen, I. Mironica, J. Uijlings, and N. Sebe. Daily living activities recognition via efficient high and low level cues combination and fisher kernel representation. In *Image Analysis and Processing (ICIAP)*, volume 8156 of *LNCS*, pages 431–441. Springer, 2013. 6

[25] M. S. Ryoo and J. K. Aggarwal. Semantic Representation and Recognition of Continued and Recursive Human Activities. *IJCV*, 82(1):1–24, 2009. 2

[26] J. Sanchez, F. Perronnin, T. Mensink, and J. Verbeek. Image Classification with the Fisher Vector: Theory and Practice. *IJCV*, 105(3):222–245, Dec. 2013. 2

[27] Z. Si, M. Pei, B. Yao, and S.-C. Zhu. Unsupervised learning of event and-or grammar and semantics from video. In *ICCV*. 2

[28] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Conditional models for contextual human motion recognition. In *ICCV*, 2005. 2

[29] E. H. Spriggs, F. De la Torre, and M. Hebert. Temporal Segmentation and Activity Classification from First-person Sensing. In *Proc. of IEEE Workshop on Egocentric Vision, CVPR*, June 2009. 4, 5

[30] S. Stein and S. J. McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *UbiComp*. ACM, 2013. 4, 5, 6

[31] C. Sun and R. Nevatia. Large-scale web video event classification by use of fisher vectors. In *WACV*, 2013. 2

[32] N. Vo and A. Bobick. From Stochastic Grammar to Bayes Network: Probabilistic Parsing of Complex Activity. In *CVPR*, 2014. 2, 4, 5, 6

[33] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *IJCV*, 103(1):60–79, 2013. 3

[34] H. Wang and C. Schmid. Action Recognition with Improved Trajectories. In *ICCV*, 2013. 1, 2, 3, 6

[35] X. Wang, L. Wang, and Y. Qiao. A comparative study of encoding, pooling and normalization methods for action recognition. In *ACCV*, 2013. 2

[36] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland. *The HTK Book, version 3.4*. Cambridge University Engineering Department, 2006. 1, 4