Making Action Recognition Robust to Occlusions and Viewpoint Changes

Daniel Weinland[1], Mustafa Özuysal[2,*], and Pascal Fua[2]

[1] Deutsche Telekom Laboratories, TU Berlin, Germany
daniel.weinland@tu-berlin.de
[2] Computer Vision Laboratory, EPFL, Switzerland
{mustafa.oezuysal,pascal.fua}@epfl.ch

Abstract. Most state-of-the-art approaches to action recognition rely on global representations either by concatenating local information in a long descriptor vector or by computing a single location independent histogram. This limits their performance in presence of occlusions and when running on multiple viewpoints. We propose a novel approach to providing robustness to both occlusions and viewpoint changes that yields significant improvements over existing techniques. At its heart is a local partitioning and hierarchical classification of the 3D Histogram of Oriented Gradients (HOG) descriptor to represent sequences of images that have been concatenated into a data volume. We achieve robustness to occlusions and viewpoint changes by combining training data from all viewpoints to train classifiers that estimate action labels independently over sets of HOG blocks. A top level classifier combines these local labels into a global action class decision.

1 Introduction

Action recognition has applications in video surveillance, human computer interaction, and multimedia retrieval, among others. It is also very challenging both because the range of possible human motions is so large and because variations in scene, viewpoint, and clothing add an additional layer of complexity.

Most state-of-the-art approaches compute image-sequence descriptors based on variants of either *sparse interest points* [3,11,17,20,24] or *dense holistic features* [9,13,19,22,23]. They integrate information over space and time into a global representation, *bag of words* or a *space-time volume*, and use a classifier, such as an SVM, to label the resulting representation.

These approaches achieve nearly perfect results on the well-known KTH and Weizmann datasets [20,1]. These, however, are relatively easy because subjects are seen from similar viewpoints and against uniform backgrounds. Furthermore, the motions in the test and training set look very similar, so that test-motions are well explained as small variations of training ones. Most of the above-mentioned publications do not report results on difficult multiview datasets, such as the

[*] This work was funded in part by Swiss National Science Foundation project.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part III, LNCS 6313, pp. 626–639, 2010.

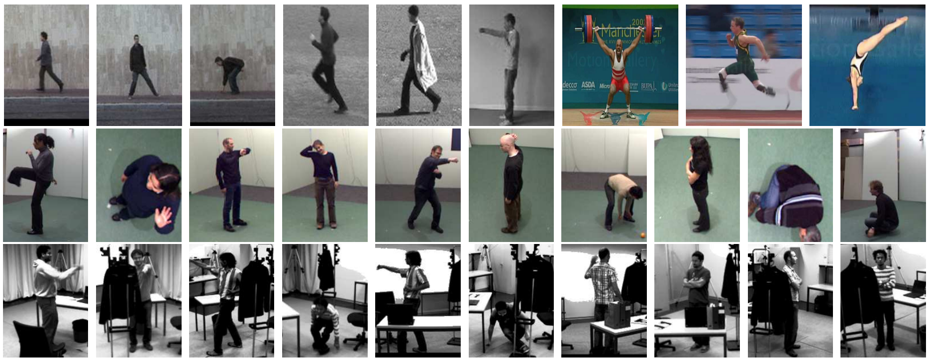


Fig. 1. We evaluate our approach on several datasets. **(Top)** Weizmann, KTH, and UCF datasets. **(Middle)** IXMAS dataset, which contains strong viewpoint changes. **(Bottom)** Finally, to measure robustness to occlusions, we evaluate the models learned from the IXMAS dataset on a new dataset that contains substantial occlusions, cluttered backgrounds, and viewpoint variations.

IXMAS [27] one, which includes subjects seen from arbitrary viewpoints. Nor do they discuss what happens when the subjects are partially occluded so that none of the training samples resembles the observation for the whole body. One must note that some of these approaches have been tested on the even more challenging Hollywood [12] dataset. However, the recognition rates on the Hollywood dataset are much lower, and the action classes contain scene context cues that can be exploited by scene classification techniques. Such discriminative scene context is not always present depending on the set of actions and also for tasks that require action classification in the same scene, such as surveillance or HCI.

To handle occlusions, an alternative to global models is to use part based ones to make independent decisions for individual body parts and to fuse them into a global interpretation [7]. However, robustly tracking body parts remains an open problem, especially in the presence of occlusions. As a result, these methods have not been tested on sequences containing substantial occlusion.

In this paper, we propose a hybrid approach that uses a local partitioning of a dense 3DHOG representation in a hierarchical classifier, which first performs local classification followed by global, to provide robustness to both viewpoint changes and occlusions. Not only can it handle sequences with substantial occlusions such as in Fig. 1, it also yields significant improvements on the IXMAS dataset [27] against recent methods explicitly designed with view-invariance in mind [5,10,27,28]. This is achieved without any performance loss on the Weizmann, KTH, and UCF datasets [1,20,18].

2 Related Work

Early attempts at view independent action recognition [15,16] required individual body parts being detected or feature points being tracked over long sequences. However, in a typical single-camera setup, it is difficult both to track individual limbs and to find feature points in images of people wearing normal clothes. Current approaches proceed differently and can be partitioned into two classes depending on whether they represent the spatio-temporal information densely or sparsely.

Sparse Representations. Many approaches rely on 3D interest points, also known as space-time corners and represent them using SIFT-like 2D descriptors [3,11,17,20,24]. These descriptors are often incorporated into a single histogram to be used for classification purposes using a Bag-of-Words (BoW) approach.

These approaches depend neither on background subtraction nor on exact localization of the person. They perform particularly well with periodic actions, such as walking or running that produce many space-time corners. A major limitation, however, is that all geometric information is lost during the BoW step and we will show that this results in a drop in performance. Furthermore, they are not suitable for action sequences that do not contain enough repeatable space time corners such as aperiodic motions.

Dense Representations. The requirement for space time corners can be eliminated by replacing sparse representations with dense ones, such as those provided by HMAX [21] or HOG [2]. These descriptors can represent 2D gradients, optical flow, or a combination thereof. For instance [22] encodes video sequences into histograms of 2D HOG descriptor and the biologically inspired approaches of [9,19] use 2D Gabor-filter responses combined with optical flow. Such dense representations avoid some of the problems discussed above but require a region of interest (ROI) around the human body, which is usually obtained by using either a separate human body detector or background subtraction followed by blob detection. Nevertheless, they have shown much better performance on the Weizmann [1] and KTH [20] datasets than sparse representations. Interestingly, improved performance on some datasets was obtained using BoW-based representations when the interest point detection was replaced by dense sampling [24].

View Independence. The above described methods have not been designed with view-independence in mind. To achieve it, several avenues have been explored. In [5], the change of silhouettes and optical flow with viewpoint is learned and used to transfer action models from a single *source-view* into novel *target-views.* This requires source and target views that record the same action, which severely limits the applicability of this technique. By contrast, in [10], actions are learned from arbitrary number of views by extracting view-invariant features based on frame-to-frame similarities within a sequence, which yields very stable features under difficult viewing conditions. However, discarding all absolute view information results in a loss of discriminative power. For instance, a moving arm or leg might produce exactly the same self-similarity measures.

Another class of techniques relies on recovering the 3D body orientation from silhouettes. For example, in [27], 3D models are projected onto 2D silhouettes with respect to different viewpoints and, in [28], 2D features are detected and back-projected onto action features based on a 3D visual hull. Such approaches require a search over model parameters to find the best match between the 3D model and the 2D observation, which is both computationally expensive and known to be relatively fragile. As a result, these techniques are usually only deployed in very constrained environments.

Occlusion Handling. The above mentioned approaches for action recognition have not been demonstrated on partially occluded action sequences. Recently, [25] tried to infer occlusion maps from a global HOG-SVM classifier for pedestrian detection by analyzing the individual contribution of each HOG block to the classifier response. However, this approach requires estimation of deterministic local occlusion labels based on a globally trained classifier. By contrast, we directly learn local SVM classifiers, each one tuned to a specific region of the HOG feature and combine the results without the need of hard decisions.

3 Recognition of Action Classes

Our approach is depicted by Fig. 2. It relies on the 3D extension of the HOG descriptor [2] to represent image sequences that have been concatenated into a data volume. The volume is subdivided into equally spaced overlapping blocks and information within each block is represented by a histogram of oriented 3D spatio-temporal gradients [11]. The resulting block descriptors are embedded temporally [26] at each spatial location, providing a discriminative representation that has fixed dimension independent of the duration of a sequence and hence can be easily fed to a classifier. By contrast to HOG and BoW, the feature descriptors are not spatially integrated into a global representation, i.e. by concatenating the blocks into a single vector (HOG) or by computing a location independent histogram of the blocks (BoW). Instead each location is individually encoded using a set of location dependent classifiers. Preserving location dependent information introduces additional discriminative power. Moreover, the local classifiers let us also estimate probabilities for occlusion, which we use to filter out contributions from cluttered and occluded regions when finally combining the local action assignments into a global decision.

In our experiments, we will demonstrate this additional robustness to occlusion over using the standard HOG and BoW. Moreover, and even though our representation is *not* view-independent, if trained using samples from different viewpoints such as those in the IXMAS dataset [27], our experiments also demonstrate strong robustness to realistic viewpoint variations. Surprisingly, our approach not only outperforms similar learning based approaches, but also those specially designed with view-independence in mind. While our approach can not generalize to view orientations that are significantly far away from all training samples, the performance of our approach does not degrade much trained on the IXMAS data and tested on new recordings acquired in a different setup and with a wide range of different viewpoints depicted by Fig. 1.

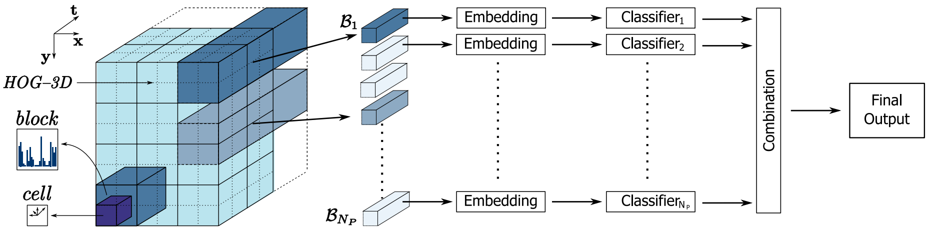


Fig. 2. We use a 3D HOG descriptor to represent a video sequence. Temporal information at each grid location is integrated over time using temporal embedding, and classified using location dependent classifiers. Finally the local results are combined into a global decision.

3.1 3D Histograms of Oriented Gradients

We use the 3DHOG descriptor introduced in [11]. In difference to [11] we compute the descriptor not at previously detected interest locations, but at densely distributed locations within a ROI centered around the person. Computing the descriptor involves the following steps: First, the region to be characterized is partitioned into regular *cells* and a histogram $\mathbf{h}$ of 3D gradient orientations is computed in each one. This compactly represents temporal and spatial texture information and is invariant to local deformations. Histograms for all cells in a small neighborhood are then concatenated into a *block* descriptor $B = L_2\text{-}\operatorname{clip}\left([\mathbf{h}_1, \ldots, \mathbf{h}_{N_C}]\right)$, to which SIFT-like L_2 normalization with clipping is applied to increase robustness. Since the *blocks* overlap with each other, this yields a redundant representation, which increases discriminative power because normalization emphasizes different bins in different blocks.

Finally, let

$$\mathcal{B}_p = [B_1, \ldots, B_{N_T}] \quad , \quad p = 1, \ldots, N_P \tag{1}$$

be the sequence of blocks computed at spatial location p along the time axis, where N_T is the number of overlapping blocks that fit within the duration of the sequence, and N_P is the number of blocks that fit within the ROI centered around the subject.

As will be discussed in the following Sections, these blocks are the primitives that we will feed first to the embedding and then to the local classifiers for recognition purposes. Such individual treatment of HOG blocks is what sets us apart from the original HOG and BoW computation that combine all blocks into a global representation, as discussed above. We will show it to be critical for occlusion handling.

3.2 Block Embedding and Classification

In this Section, we present an effective way to compute the probability that a block represents a specific action using information from all subsequences along the temporal axis. To this end, we create a set $\mathcal{V} = \{V_1, \ldots, V_{N_V}\}$ of N_V prototype descriptors by randomly sampling the HOG blocks computed for the training subsequences. Given an action sequence and the block descriptors of Eq.1, we create an N_V-dimensional vector made of the distances of each one of the V_i to the closest block within the sequence. In the case of a sequence belonging to the training set, some of these distances will be exactly zero since some elements of $\mathcal{V}$ are contained in its set of block descriptors but they may not be the only one to be small. Prototypes that do not belong to the sequence but resemble one of the blocks will also be assigned a small value. This *Sequence Embedding*, which is inspired by *max-pooling* of action descriptors [9] and *exemplar-based embedding* [26], makes the training and recognition much more effective. We discuss it in more details below.

Let $\mathcal{B}_p$ be a sequence of blocks at spatial location p partitioned into N_T overlapping blocks, as defined in Eq.1. We represent $\mathcal{B}_p$ in terms of minimum-distances to the set $\mathcal{V}$ of N_V prototype descriptors introduced above. We take the distance of the sequence to each V_i to be

$$d_i^*(\mathcal{B}_p) = \min_t d(B_t, V_i) \quad , \quad B_t \in \mathcal{B}_p \quad , \tag{2}$$

where d represents the distance between orientation histograms. We compute it as the χ^2-distance

$$d(B, V) = \frac{1}{2} \sum_k \frac{(h_k - v_k)^2}{h_k + v_k} \quad , \tag{3}$$

which we experimentally found to be more suited for our purposes than both the squared-Euclidean-distance and Kullback-Leibler divergence. Fig 3 illustrates the embedding for an action sequence.

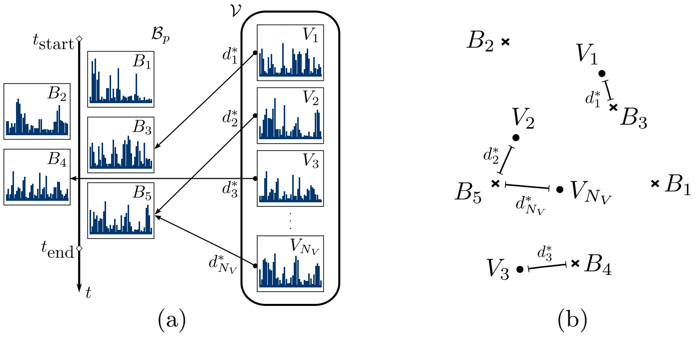


Fig. 3. Embedding of HOG block sequence. **(a)** Each exemplar V_i is compared against all blocks extracted from the sequence using the χ^2-distance and the minimum distance d_i^* is stored in a feature vector that we use for classification. The blocks extracted from the HOG descriptors overlap to minimize quantization error. **(b)** Same set of blocks and exemplars visualized in the space of histograms.

We then take the resulting set of d^* distances

$$D_p^* = (d_1^*(\mathcal{B}_p), \dots, d_{N_V}^*(\mathcal{B}_p))^\top \in \mathbb{R}^{N_V}, \tag{4}$$

as input to a classifier trained for location p.

An alternative is to use a local BoW approach that performs the bagging along the time axis. Each HOG descriptor in the sequence can vote for the closest words in the vocabulary and a histogram over the vocabulary can be input to the SVMs. Since N_V is much larger than N_T in a typical sequence, every descriptor must vote for multiple words in the vocabulary to avoid quantization effects and sparse histograms. This can be facilitated by votes that decay exponentially with the distance between B_t and V_i. Optimizing the rate of decay for each dataset yields comparable performance to the embedding method. However we prefer the embedding method because it is simpler and does not involve adjusting an additional parameter to each dataset.

We pick the exemplars V_i from the training set by random sampling. We experimented with a selection strategy as in [26]. This gave better results with a small number of exemplars, however using a sufficiently large number (500) the performance of both approaches was very close. We therefore report results for random selection since it is simpler.

Finally, we use L2-regularized logistic regression [4] to produce probability estimates $p(c|D_p^*, \Theta_p)$ for each class $c = 1, \dots, N_C$, where D_p^* is the descriptor of Eq. 4 and Θ_p is the learned logistic regression weights at position p.

3.3 Occlusion Handling

The overall framework that we propose resembles that of a global HOG representation that is well known for being sensitive to occlusions [25]. We have introduced the local partitioning and embedding of the 3DHOG descriptor to preserve the advantages of HOG, while at the same time making it robust to occlusions. This is achieved by individually classifying each embedded block descriptor and then combining the classification responses from all blocks in a final stage as detailed in the next Section.

To further improve occlusion robustness, we learn at each location in addition to the N_c actions a separate class. Thus $p(c = N_C + 1|D_p^*, \Theta_p)$ represents the probability of region p being occluded. If a region is occluded with high probability, and because the probability distribution normalizes to one, the probabilities for all other classes will be reduced. Hence when fusing the results as discussed in the next Section, such a region will carry reduced weight.

To generate a large variety of potential occlusions during training, we artificially hide parts of the training images, as shown in Fig. 4. These occluders are placed so that approximately either the lower part of the body, the right or left side is occluded. We then calculate for each region the amount of overlap with the occluding object; if it is higher than a predefined threshold the corresponding HOG block is labeled as belonging to the *occluded* class during training. In practice, we found that setting the threshold to 90% yields the best results.

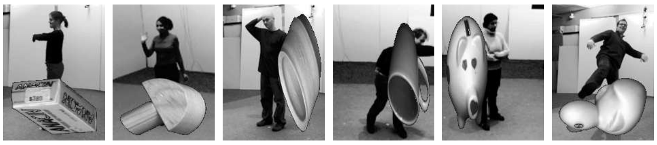


Fig. 4. Sample images from artificially occluded training data used to introduce additional robustness against occlusions in learned classifiers

3.4 Classifier Combination

The previously described local classifiers produce action probabilities at uniformly distributed locations of the HOG window. We have evaluated the following strategies to fuse these results into a single decision.

Product Rule. Our classifiers produce probabilities $p(c|D_p^*, \Theta_p)$. Thus if independence can be assumed, the natural choice is to combine these by the product rule $p(c|D_{1:N_P}^*, \Theta_{1:N_P}) = \prod_p p(c|D_p^*, \Theta_p)$. Note that we choose the sigmoid parameters [4] that are used to convert the classifier outputs into probabilities so that the resulting probability estimates are not overly confident.

Sum Rule. It is also possible to compute a score for each class by averaging the probabilities of the individual classifiers, i.e. $f(c) = \sum_p p(c|D_p^*, \Theta_p)$, which can produce better results than the product rules, when the probabilities are not accurately estimated.

Weighted Sum. Not every region of the HOG window carries equally discriminative information for each action. Thus, when summing the individual probabilities from each region they can be weighted accordingly. One way to choose the weights is via conditional error probabilities $p(\tilde{c}|c, p)$, which represent the probability that the true class label is $\tilde{c}$ conditioned on the actual output c of a classifier. Following [8], a weighed sum can then be computed as $p(\tilde{c}|D_{1:N_P}^*, \Theta_{1:N_P}) = \sum_p \sum_c p(c|D_p^*, \Theta_p)p(\tilde{c}|c, p)$. Thus, intuitively, a local classifier that is easily confused between several actions will distribute its vote over all those actions, while a classifier that is very confident in classifying an action will account its vote only to this action. The conditional error probabilities $p(\tilde{c}|c, p)$ are estimated from confusion matrices, i.e. by counting how often an observation is classified as c if the true class label was actually $\tilde{c}$.

Top-Level SVM Classifier. Using a hierarchical classification scheme, we can combine the outputs of all local classifiers into a single feature vector and learn a global SVM classifier on top of this representation.

As shown in Table 2, when there are no occlusions, the product rule combination and the SVM classifier perform best, closely followed by the rest. However, as shown in Table 4, occlusions degrade the performance of the product rule even below that of the sum rule. This was to be expected since we use artificial occlusions in training and real ones in the testing sets and the product rule is the most sensitive one to biases in the learned probabilities. Overall, we therefore prefer the weighted sum and SVM based methods since they result in higher or at least similar classification rates on all datasets.

4 Experiments

We experimented with the well-known Weizmann [1], KTH [20], UCF [18], and IXMAS [27] datasets. Since none of these datasets includes occluded subjects, we also acquired and processed our own video sequences involving the actions in the IXMAS dataset, but with substantial occlusions and cluttered backgrounds.

We implemented two baseline methods to compare our results on this newly acquired multiview dataset with occlusions. Both methods use the same 3DHOG features and training data as the local method that we advocate in this paper. However *global SVM*, the first baseline method, combines the HOG blocks into a single feature vector followed by global embedding along the temporal axis and a linear SVM classifier. This method hence resembles the original global HOG approach [2] combined with the temporal embedding of [26]. *BoW SVM*, the second baseline method, accumulates the HOG blocks into a histogram of 4000 visual words and classifies them using a non-linear SVM with χ^2-kernel. This approach hence resembles the approaches [11,12] and more specifically the dense 3DHOG representations in [24], except that for comparison purposes we sample features *not* at multiple scales, because the local approach and global HOG also use only a single scale, and we use information only within the same ROIs centered around the subjects as for the other methods.

To compute the ROIs around people that our approach requires, we proceed as follows. For KTH, we use the bounding boxes provided by [13]. For UCF we use the bounding boxes available in the dataset. For Weizmann and IXMAS we use the background subtracted silhouettes and fit a bounding box around them.

For our new recordings we interactively determine the bounding box in every first frame of an action, because simple background subtraction can not accurately detect the partially occluded persons. For all datasets, the ROIs are scaled and concatenated to produce $48 \times 64 \times t$ cubes, where t is the number of frames in the sequence.

Unless stated otherwise, we use $16 \times 16 \times 16$ pixel blocks subdivided in $2 \times 2 \times 2$ cells for 3DHOG, which implies an overlap of 8 pixels in all dimensions. We compute histograms using the dodecahedron based quantization [11] with 6 orientation bins. For the embedding we use a set of approximately 500 prototypes.

Also, unless stated otherwise, recognition rates are computed by the leave-one-out method: If K subjects appear in a dataset, we average over K runs, leaving a different person out of the training set each time.

The recognition speed depends on the length of a sequence and on the HOG and embedding dimensions used. With our experimental setting on the IXMAS data, computing the HOG features takes on average 75.5ms per sequence, with our Matlab implementation on a Core i7 CPU. The cost of computing the embedding is on average 34ms per sequence. The hierarchical classification is the fastest step and takes on average 1ms per sequence.

Table 1. Comparison of recognition rates (in %) on Weizmann (left), KTH (middle), and UCF (right) datasets

Method	Weizmann
Local SVM	**100.0**
Local Weighted	**100.0**
Local Product	**100.0**
Local Sum	**100.0**
Global SVM	**100.0**
BoW SVM	**100.0**
Lin [13]	**100.0**
Schindler [19]	**100.0**
Blank [1]	99.6
Jhuang [9]	98.8
Thurau [22]	94.4
Kläser [11]	84.3

Method	KTH
Local SVM	92.2
Local Weighted	92.4
Local Product	92.2
Local Sum	92.0
Global SVM	90.7
BoW SVM	89.3
Gilbert [6]	**94.5**
Lin [13]	93.4
Schindler [19]	92.7
Wang [12]	92.1
Laptev [12]	91.8
Jhuang [9]	91.7
Kläser [11]	91.4
Rodriguez [18]	88.7
Schuldt [20]	71.7

Method	UCF
Local SVM	**90.1**
Local Weighted	89.4
Local Product	87.7
Local Sum	87.7
Global SVM	85.6
BoW SVM	81.2
Wang [24]	85.6
Rodriguez [18]	69.2

4.1 Weizmann, KTH, and UCF Datasets

The Weizmann dataset consists of videos of 9 actors performing 9 actions. Recently, several approaches reported close to perfect recognition rates on this relatively easy dataset. Note that existing approaches use slightly different evaluation methodologies on the data. Some evaluate on the whole sequences, others split sequences into multiple subparts. We report here results for the full sequences, where our method yields perfect recognition rates, that is 100%. In Table 1, we summarize our recognition results and compare them against other approaches.

The KTH dataset consists of 6 actions performed by 25 actors in four different scenarios. We follow the evaluation procedure of the original paper [20] and split the data into training/validation (8+8 people) and testing (9 people) sets, and report results for learning a single model from all scenarios. Note that some of the approaches use slightly different evaluation schemes, e.g. a leave-one-out cross validation, or do not require bounding boxes, etc. Optimizing our parameters on the validation set, we found HOG blocks of size $16 \times 16 \times 2$ subdivided into $2 \times 2 \times 1$ cells, and an icosahedron based quantization to give best results. With this setting, we achieve a recognition rate of 92.4% using the weighted sum based combination, which is among the best results reported for this dataset.

In Table 1, we summarize our recognition results and compare them against other approaches.

We also evaluate our approach on the UCF dataset that consists of 10 actions. Since the publicly available part of the dataset does not contain the videos for *pole vaulting*, we report results using the 9 available ones and achieve a mean recognition rate of 90.1% using the SVM which is the best reported result for this dataset. Note that these are not directly comparable to the reported rate of 69.2% [18], nevertheless, they demonstrate that our approach generalizes well to broadcast action videos.

4.2 IXMAS Dataset

The IXMAS dataset [27] is a multiview action recognition dataset. It consists of videos of 10 actors performing each 3 times 11 actions. Each action was recorded with 5 cameras observing the subjects from very different perspectives and as shown in Fig. 1, the actors freely choose their orientation for each sequence.

We learn single action models from all camera views. Average recognition rates for the different combination strategies are shown in Table 2 evaluated on all cameras. In Table 3 we show individual rates per camera when learning from all views or individual views, and also compare against other methods that used the same evaluation methodology on the full IXMAS dataset. For each camera, we improve upon previously published results.

In summary, we observed that combining training data from multiple viewpoints and using a non-invariant dense representation yields comparable recognition rates than invariant representations. However, performance is adversely affected by local changes in feature statistics. Our local classification step mitigates this problem. As a result, our local approach performs better than competing ones.

Table 2. Average recognition rates (in %) on IXMAS dataset for different combination strategies for our local method compared against the global SVM and BoW SVM baselines

Method	Local SVM	Local Product	Local Sum	Local Weighted	Global SVM	BoW SVM
Rec. Rate	83.4	**83.5**	82.8	82.4	80.6	71.9

4.3 IXMAS Actions with Occlusions

To demonstrate the generalization power of our approach, we recorded our own dataset composed of the IXMAS actions, but performed by different actors, who could be partially occluded. The actions were performed on average 3 times by 6 actors and recorded with 5 cameras. As shown in Fig. 1, actors chose their orientation freely and the occluding objects were rearranged between each take.

We split the data into two subsets: 395 sequences were recorded without occlusions, and 698 sequences contain objects partially occluding the actors. We then evaluate on the two sets by learning from all sequences of the original IXMAS dataset and by testing on either one of these subsets.

Table 3. Recognition rates (in %) on IXMAS dataset for individual cameras. The left half of the table shows the results when all cameras are used for training. The other half shows the results for training using a single camera.

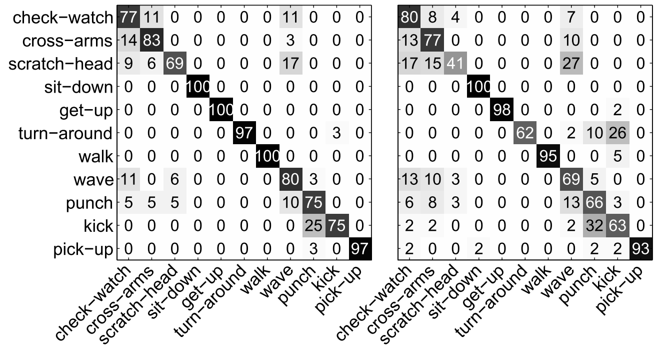
Method	Training with All Cameras						Training with Single Camera				
	all	cam1	cam2	cam3	cam4	cam5	cam1	cam2	cam3	cam4	cam5
Local SVM	83.4	86.7	**89.9**	**86.4**	**87.6**	66.4	84.7	85.8	87.9	**88.5**	72.6
Local Product	**83.5**	**87.0**	88.3	85.6	87.0	**69.7**	**85.8**	**86.4**	**88.0**	88.2	**74.7**
Tran [23]	80.2	—	—	—	—	—	—	—	—	—	—
Liu [14]	—	76.7	73.3	72.0	73.0	—	—	—	—	—	—
Junejo [10]	72.7	74.8	74.5	74.8	70.6	61.2	76.4	77.6	73.6	68.8	66.1
Reddy [17]	72.6	69.6	69.2	62.0	65.1	—	—	—	—	—	—
Yan [28]	—	—	—	—	—	—	72.0	53.0	68.0	63.0	—
Farhadi [5]	58.1	—	—	—	—	—	—	—	—	—	—
Weinland [27]	57.9	65.4	70.0	54.3	66.0	33.6	55.2	63.5	—	60.0	—

Table 4. Average recognition rates (in %) when learning from IXMAS dataset and testing on new *clean* and *occluded* recordings. Results are shown for learning models with (*oc*) and without (*no oc*) the additional *occlusion class*. In all cases our local combination strategy outperforms the baselines.

Method	clean		occluded	
	no oc	oc	no oc	oc
Local SVM	83.5	**86.3**	61.9	**76.7**
Local Weighted	83.3	85.1	61.6	**76.7**
Local Sum	79.0	82.5	54.0	72.8
Local Product	77.7	81.5	44.6	68.9
Global SVM	74.4	76.0	46.1	58.3
BoW SVM	47.1	52.9	18.1	27.8

Results are shown in Fig. 5 and Table 4. Columns *clean* in Table 4 show results on the occlusion free sequences. This is relevant because it still requires that our approach generalizes to new viewpoints and actors not included in the training data. Because the sequences contain no occlusions, also the performance of the global HOG classifier generalizes well to this sequences (74.4%). Interestingly, when introducing the additional occlusion classifier, performance on the dataset improves (86.3% for SVM combination), even though it contains no occlusions. This is because the occlusion classifier also responds to background clutter, reducing its effect on classification. Note, that for columns *oc* the baseline classifiers were trained using all clean as well as all artificially occluded sequences as a single training set.

When evaluating on the sequences with occlusions the effect of the additional occlusion classifier becomes even more evident. We observe the best performance with 76.7% recognition rate for the SVM based combination and also for the weighted sum.



	check−watch	cross−arms	scratch−head	sit−down	get−up	turn−around	walk	wave	punch	kick	pick−up
check−watch	77	11	0	0	0	0	0	11	0	0	0
cross−arms	14	83	0	0	0	0	0	3	0	0	0
scratch−head	9	6	69	0	0	0	0	17	0	0	0
sit−down	0	0	0	100	0	0	0	0	0	0	0
get−up	0	0	0	0	100	0	0	0	0	0	0
turn−around	0	0	0	0	0	97	0	0	0	3	0
walk	0	0	0	0	0	0	100	0	0	0	0
wave	11	0	6	0	0	0	0	80	3	0	0
punch	5	5	5	0	0	0	0	10	75	0	0
kick	0	0	0	0	0	0	0	0	25	75	0
pick−up	0	0	0	0	0	0	0	0	3	0	97

	check−watch	cross−arms	scratch−head	sit−down	get−up	turn−around	walk	wave	punch	kick	pick−up
check−watch	80	8	4	0	0	0	0	7	0	0	0
cross−arms	13	77	0	0	0	0	0	10	0	0	0
scratch−head	17	15	41	0	0	0	0	27	0	0	0
sit−down	0	0	0	100	0	0	0	0	0	0	0
get−up	0	0	0	0	98	0	0	0	0	2	0
turn−around	0	0	0	0	0	62	0	2	10	26	0
walk	0	0	0	0	0	0	95	0	0	5	0
wave	13	10	3	0	0	0	0	69	5	0	0
punch	6	8	3	0	0	0	0	13	66	3	0
kick	2	2	0	0	0	0	0	2	32	63	0
pick−up	2	0	0	2	0	0	0	0	2	2	93

Fig. 5. Confusion matrixes (in %) for the new IXMAS recording. **(Left)** Recordings *without* occlusions with average recognition rate 86.3%. **(Right)** Recordings *with* occlusions with average recognition rate 76.7%.

In all cases, our experiments demonstrates that using local classifiers as well as explicitly introducing occlusions into the training set leads to strong performance improvements for recognition of partially occluded actions.

5 Conclusion

In this paper, we proposed a new approach based on a local 3D HOG descriptor. Our approach is simple, efficient, and combines the benefits of the HOG based dense representation with that of local approaches to achieve occlusion robust action recognition. We demonstrated that our descriptor, when trained from multiple views, can perform action recognition from multiple viewpoints, with highest recognition rates on the difficult IXMAS dataset. Moreover, we showed that these results carry over to new situations, with different backgrounds, subjects, viewpoints, and partial occlusions.

References

1. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: ICCV, pp. 1395–1402 (2005)
2. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: CVPR (2005)
3. Dollar, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: VS-PETS. pp. 65–72 (2005)
4. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. JMLR 9, 1871–1874 (2008)
5. Farhadi, A., Tabrizi, M.K.: Learning to recognize activities from the wrong view point. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 154–166. Springer, Heidelberg (2008)

6. Gilbert, A., Illingworth, J., Bowden, R.: Fast realistic multi-action recognition using mined dense spatio-temporal features. In: ICCV (2009)
7. Ikizler, N., Forsyth, D.: Searching video for complex activities with finite state models. In: Forsyth, D. (ed.) CVPR, pp. 1–8 (2007)
8. Ivanov, Y., Heisele, B., Serre, T.: Using component features for face recognition. In: FG (2004)
9. Jhuang, H., Serre, T., Wolf, L., Poggio, T.: A biologically inspired system for action. In: ICCV (2007)
10. Junejo, I., Dexter, E., Laptev, I., Prez, P.: Cross-view action recognition from temporal self-similarities. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 293–306. Springer, Heidelberg (2008)
11. Kläser, A., Marszałek, M., Schmid, C.: A spatio-temporal descriptor based on 3d-gradients. In: BMVC (2008)
12. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR (2008)
13. Lin, Z., Jiang, Z., Davis, L.S.: Recognizing actions by shape-motion prototype trees. In: ICCV (2009)
14. Liu, J., Shah, M.: Learning human actions via information maximization. In: CVPR (2008)
15. Parameswaran, V., Chellappa, R.: View invariants for human action recognition. In: CVPR, vol. 2, pp. II–613–II–19 (2003)
16. Rao, C., Yilmaz, A., Shah, M.: View-invariant representation and recognition of actions. In: IJCV, vol. 50(2), pp. 203–226 (2002)
17. Reddy, K.K., Liu, J., Shah, M.: Incremental action recognition using feature-tree. In: ICCV (2009)
18. Rodriguez, M.D., Ahmed, J., Shah, M.: Action mach a spatio-temporal maximum average correlation height filter for action recognition. In: CVPR, pp. 1–8 (2008)
19. Schindler, K., van Gool, L.: Action snippets: How many frames does human action recognition require? In: CVPR, pp. 1–8 (2008)
20. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: A local svm approach. In: ICPR, pp. 32–36 (2004)
21. Serre, T., Wolf, L., Poggio, T.: Object recognition with features inspired by visual cortex. In: CVPR (2005)
22. Thurau, C., Hlavac, V.: Pose primitive based human action recognition in videos or still images. In: CVPR, pp. 1–8 (2008)
23. Tran, D., Sorokin, A.: Human activity recognition with metric learning. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 548–561. Springer, Heidelberg (2008)
24. Wang, H., Ullah, M.M., Kläser, A., Laptev, I., Schmid, C.: Evaluation of local spatio-temporal features for action recognition. In: BMVC (2009)
25. Wang, X., Han, T.X., Yan, S.: An hog-lbp human detector with partial occlusion handling. In: ICCV (2009)
26. Weinland, D., Boyer, E.: Action recognition using exemplar-based embedding. In: CVPR (2008)
27. Weinland, D., Boyer, E., Ronfard, R.: Action recognition from arbitrary views using 3d exemplars. In: ICCV (2007)
28. Yan, P., Khan, S.M., Shah, M.: Learning 4d action feature models for arbitrary view action recognition. In: CVPR (2008)