Learning realistic human actions from movies

Ivan Laptev
INRIA Rennes, IRISA
ivan.laptev@inria.fr

Marcin Marszałek
INRIA Grenoble, LEAR - LJK
marcin.marszalek@inria.fr

Cordelia Schmid
INRIA Grenoble, LEAR - LJK
cordelia.schmid@inria.fr

Benjamin Rozenfeld
Bar-Ilan University
grurgrur@gmail.com

Abstract

The aim of this paper is to address recognition of natural human actions in diverse and realistic video settings. This challenging but important subject has mostly been ignored in the past due to several problems one of which is the lack of realistic and annotated video datasets. Our first contribution is to address this limitation and to investigate the use of movie scripts for automatic annotation of human actions in videos. We evaluate alternative methods for action retrieval from scripts and show benefits of a text-based classifier. Using the retrieved action samples for visual learning, we next turn to the problem of action classification in video. We present a new method for video classification that builds upon and extends several recent ideas including local space-time features, space-time pyramids and multi-channel non-linear SVMs. The method is shown to improve state-of-the-art results on the standard KTH action dataset by achieving 91.8% accuracy. Given the inherent problem of noisy labels in automatic annotation, we particularly investigate and show high tolerance of our method to annotation errors in the training set. We finally apply the method to learning and classifying challenging action classes in movies and show promising results.

1. Introduction

In the last decade the field of visual recognition had an outstanding evolution from classifying instances of toy objects towards recognizing the classes of objects and scenes in natural images. Much of this progress has been sparked by the creation of realistic image datasets as well as by the new, robust methods for image description and classification. We take inspiration from this progress and aim to transfer previous experience to the domain of video recognition and the recognition of human actions in particular.

Existing datasets for human action recognition (e.g. [15], see figure 8) provide samples for only a few action classes recorded in controlled and simplified settings. This stands in sharp contrast with the demands of real applications focused on natural video with human actions subjected to individual variations of people in expression, posture, motion and clothing; perspective effects and camera motions; illumination variations; occlusions and variation in scene surroundings. In this paper we address limitations of current datasets and collect realistic video samples with human actions as illustrated in figure 1. In particular, we consider the difficulty of manual video annotation and present a method for automatic annotation of human actions in movies based on script alignment and text classification (see section 2).

Figure 1. Realistic samples for three classes of human actions: kissing; answering a phone; getting out of a car. All samples have been automatically retrieved from script-aligned movies.

Action recognition from video shares common problems with object recognition in static images. Both tasks have to deal with significant intra-class variations, background clutter and occlusions. In the context of object recognition in static images, these problems are surprisingly well handled by a bag-of-features representation [17] combined with state-of-the-art machine learning techniques like Support Vector Machines. It remains, however, an open question whether and how these results generalize to the recognition of realistic human actions, e.g., in feature films or personal videos.

Building on the recent experience with image classification, we employ spatio-temporal features and generalize spatial pyramids to spatio-temporal domain. This allows us to extend the spatio-temporal bag-of-features representation with weak geometry, and to apply kernel-based learning techniques (cf. section 3). We validate our approach on a standard benchmark [15] and show that it outperforms the state-of-the-art. We next turn to the problem of action classification in realistic videos and show promising results for eight very challenging action classes in movies. Finally, we present and evaluate a fully automatic setup with action learning and classification obtained for an automatically labeled training set.

1.1. Related work

Our script-based annotation of human actions is similar in spirit to several recent papers using textual information for automatic image collection from the web [10, 14] and automatic naming of characters in images [1] and videos [4]. Differently to this work we use more sophisticated text classification tools to overcome action variability in text. Similar to ours, several recent methods explore bag-of-features representations for action recognition [3, 6, 13, 15, 19], but only address human actions in controlled and simplified settings. Recognition and localization of actions in movies has been recently addressed in [8] for a limited dataset, i.e., manual annotation of two action classes. Here we present a framework that scales to automatic annotation for tens or more visual action classes. Our approach to video classification borrows inspiration from image recognition methods [2, 9, 12, 20] and extends spatial pyramids [9] to space-time pyramids.

2. Automatic annotation of human actions

This section describes an automatic procedure for collecting annotated video data for human actions from movies. Movies contain a rich variety and a large number of realistic human actions. Common action classes such as kissing, answering a phone and getting out of a car (see figure 1), however, often appear only a few times per movie. To obtain a sufficient number of action samples from movies for visual training, it is necessary to annotate tens or hundreds of hours of video which is a hard task to perform manually.

To avoid the difficulty of manual annotation, we make use of *movie scripts* (or simply "scripts"). Scripts are publicly available for hundreds of popular movies[1] and provide text description of the movie content in terms of scenes, characters, transcribed dialogs and human actions. Scripts as a mean for video annotation have been previously used for the automatic naming of characters in videos by Everingham et al. [4]. Here we extend this idea and apply text-based script search to automatically collect video samples for human actions.

Automatic annotation of human actions from scripts, however, is associated with several problems. Firstly, scripts usually come without time information and have to be aligned with the video. Secondly, actions described in scripts do not always correspond with the actions in movies. Finally, action retrieval has to cope with the substantial variability of action expressions in text. In this section we address these problems in subsections 2.1 and 2.2 and use the proposed solution to automatically collect annotated video samples with human actions, see subsection 2.3. The resulting dataset is used to train and to evaluate a visual action classifier later in section 4.

2.1. Alignment of actions in scripts and video

Movie scripts are typically available in plain text format and share similar structure. We use line indentation as a simple feature to parse scripts into monologues, character names and scene descriptions (see figure 2, right). To align scripts with the video we follow [4] and use time information available in movie subtitles that we separately download from the Web. Similar to [4] we first align speech sections in scripts and subtitles using word matching and dynamic programming. We then transfer time information from subtitles to scripts and infer time intervals for scene descriptions as illustrated in figure 2. Video clips used for action training and classification in this paper are defined by time intervals of scene descriptions and, hence, may contain multiple actions and non-action episodes. To indicate a possible misalignment due to mismatches between scripts and subtitles, we associate each scene description with the alignment score a. The a-score is computed by the ratio of matched words in the near-by monologues as $a = (\#matched\ \ words)/(\#all\ \ words)$.

Temporal misalignment may result from the discrepancy between subtitles and scripts. Perfect subtitle alignment ($a = 1$), however, does not yet guarantee the correct action annotation in video due to the possible discrepancy between

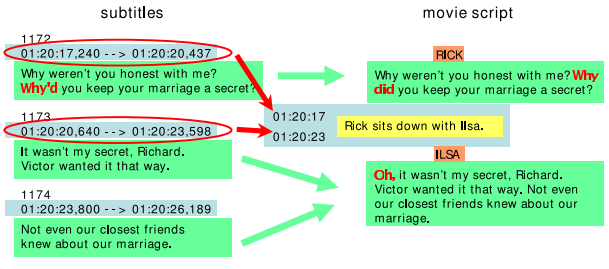

Figure 2. Example of matching speech sections (green) in subtitles and scripts. Time information (blue) from adjacent speech sections is used to estimate time intervals of scene descriptions (yellow).

[1] We obtained hundreds of movie scripts from www.dailyscript.com, www.movie-page.com and www.weeklyscript.com.

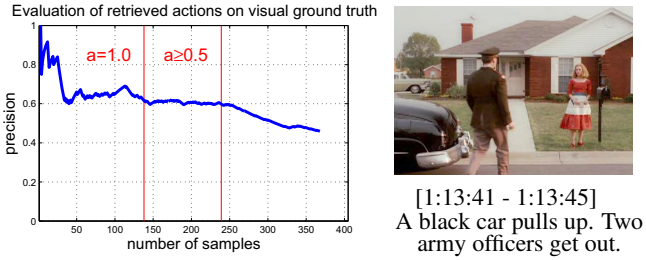

Figure 3. Evaluation of script-based action annotation. Left: Precision of action annotation evaluated on visual ground truth. Right: Example of a visual false positive for "get out of a car".

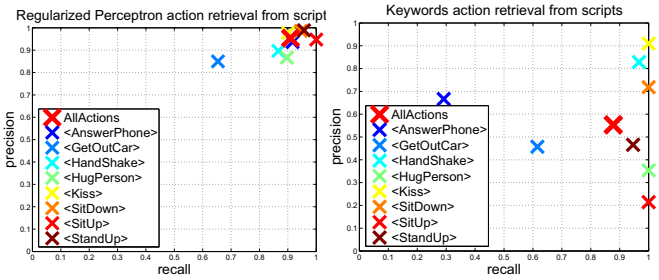

Figure 4. Results of retrieving eight classes of human actions from scripts using regularized perceptron classifier (left) and regular expression matching (right).

scripts and movies. To investigate this issue, we manually annotated several hundreds of actions in 12 movie scripts and verified these on the visual ground truth. From 147 actions with correct text alignment (a=1) only 70% did match with the video. The rest of samples either were misaligned in time (10%), were outside the field of view (10%) or were completely missing in the video (10%). Misalignment of subtitles ($a < 1$) further decreases the visual precision as illustrated in figure 3 (left). Figure 3 (right) shows a typical example of a "visual false positive" for the action "get out of a car" occurring outside the field of view of the camera.

2.2. Text retrieval of human actions

Expressions for human actions in text may have a considerable within-class variability. The following examples illustrate variations in expressions for the "GetOutCar" action: *"Will gets out of the Chevrolet."*, *"A black car pulls up. Two army officers get out."*, *"Erin exits her new truck."*. Furthermore, false positives might be difficult to distinguish from positives, see examples for the "SitDown" action: *"About to sit down, he freezes."*, *"Smiling, he turns to sit down. But the smile dies on his face when he finds his place occupied by Ellie."*. Text-based action retrieval, hence, is a non-trivial task that might be difficult to solve by a simple keyword search such as commonly used for retrieving images of objects, e.g. in [14].

To cope with the variability of text describing human actions, we adopt a machine learning based text classification approach [16]. A classifier labels each scene description in scripts as containing the target action or not. The implemented approach relies on the bag-of-features model, where each scene description is represented as a sparse vector in a high-dimensional feature space. As features we use words, adjacent pairs of words, and non-adjacent pairs of words occurring within a small window of N words where N varies between 2 and 8. Features supported by less than three training documents are removed. For the classification we use a regularized perceptron [21], which is equivalent to a support vector machine. The classifier is trained on a manually labeled set of scene descriptions, and the parameters (regularization constant, window size N, and the acceptance threshold) are tuned using a validation set.

We evaluate text-based action retrieval on our eight classes of movie actions that we use throughout this paper: *AnswerPhone, GetOutCar, HandShake, HugPerson, Kiss, SitDown, SitUp, StandUp*. The text test set contains 397 action samples and over 17K non-action samples from 12 manually annotated movie scripts. The text training set was sampled from a large set of scripts different from the test set. We compare results obtained by the regularized perceptron classifier and by matching regular expressions which were manually tuned to expressions of human actions in text. The results in figure 4 very clearly confirm the benefits of the text classifier. The average precision-recall values for all actions are [prec. 0.95 / rec. 0.91] for the text classifier versus [prec. 0.55 / rec. 0.88] for regular expression matching.

2.3. Video datasets for human actions

We construct two video training sets, a manual and an automatic one, as well as a video test set. They contain video clips for our eight classes of movie actions (see top row of figure 10 for illustration). In all cases we first apply automatic script alignment as described in section 2.1. For the *clean, manual* dataset as well as the test set we manually select visually correct samples from the set of manually text-annotated actions in scripts. The *automatic* dataset contains training samples that have been retrieved automatically from scripts by the text classifier described in section 2.2. We limit the automatic training set to actions with an alignment score $a > 0.5$ and a video length of less than 1000 frames. Our manual and automatic training sets contain action video sequences from 12 movies [2] and the test set actions from 20 different movies [3]. Our datasets, i.e., the

[2] "American Beauty", "Being John Malkovich", "Big Fish", "Casablanca", "The Crying Game", "Double Indemnity", "Forrest Gump", "The Godfather", "I Am Sam", "Independence Day", "Pulp Fiction" and "Raising Arizona".

[3] "As Good As It Gets", "Big Lebowski", "Bringing Out The Dead", "The Butterfly Effect", "Dead Poets Society", "Erin Brockovich", "Fargo", "Gandhi", "The Graduate", "Indiana Jones And The Last Crusade", "Its A Wonderful Life", "Kids", "LA Confidential", "The Lord of the Rings: Fellowship of the Ring", "Lost Highway", "The Lost Weekend", "Mission To Mars", "Naked City", "The Pianist" and "Reservoir Dogs".

	<AnswerPhone>	<GetOutCar>	<HandShake>	<HugPerson>	<Kiss>	<SitDown>	<SitUp>	<StandUp>	Total labels	Total samples
False	5	6	9	7	10	21	5	33	96	
Correct	15	6	14	8	34	30	7	29	143	
All	20	12	23	15	44	51	12	62	239	233
automatically labeled training set										
	22	13	20	22	49	47	11	47	231	219
manually labeled training set										
	23	13	19	22	51	30	10	49	217	211
test set										

Table 1. The number of action labels in automatic training set (top), clean/manual training set (middle) and test set (bottom).

video clips and the corresponding annotations, are available at `http://www.irisa.fr/vista/actions`.

The objective of having two training sets is to evaluate recognition of actions both in a supervised setting and with automatically generated training samples. Note that no manual annotation is performed neither for scripts nor for videos used in the automatic training set. The distribution of action labels for the different subsets and action classes is given in table 1. We can observe that the number of correctly labeled videos in the automatic set is 60%. Most of the wrong labels result from the script-video misalignment and a few additional errors come from the text classifier. The problem of classification in the presence of wrong training labels will be addressed in section 4.3.

3. Video classification for action recognition

This section presents our approach for action classification. It builds on existing bag-of-features approaches for video description [3, 13, 15] and extends recent advances in static image classification to videos [2, 9, 12]. Lazebnik et al. [9] showed that a spatial pyramid, i.e., a coarse description of the spatial layout of the scene, improves recognition. Successful extensions of this idea include the optimization of weights for the individual pyramid levels [2] and the use of more general spatial grids [12]. Here we build on these ideas and go a step further by building space-time grids. The details of our approach are described in the following.

3.1. Space-time features

Sparse space-time features have recently shown good performance for action recognition [3, 6, 13, 15]. They provide a compact video representation and tolerance to background clutter, occlusions and scale changes. Here we follow [7] and detect interest points using a space-time extension of the Harris operator. However, instead of performing scale selection as in [7], we use a multi-scale approach and extract features at multiple levels of spatio-temporal scales (σ_i^2, τ_j^2) with $\sigma_i = 2^{(1+i)/2}, i = 1, ..., 6$ and $\tau_j = 2^{j/2}, j = 1, 2$. This choice is motivated by the reduced computational complexity, the independence from scale selection artifacts and the recent evidence of good recognition performance using dense scale sampling. We also eliminate detections due to artifacts at shot boundaries [11]. Interest points detected for two frames with human actions are illustrated in figure 5.

Figure 5. Space-time interest points detected for two video frames with human actions hand shake (left) and get out car (right).

To characterize motion and appearance of local features, we compute histogram descriptors of space-time volumes in the neighborhood of detected points. The size of each volume $(\Delta_x, \Delta_y, \Delta_t)$ is related to the detection scales by $\Delta_x, \Delta_y = 2k\sigma, \Delta_t = 2k\tau$. Each volume is subdivided into a (n_x, n_y, n_t) grid of cuboids; for each cuboid we compute coarse histograms of oriented gradient (*HoG*) and optic flow (*HoF*). Normalized histograms are concatenated into HoG and HoF descriptor vectors and are similar in spirit to the well known SIFT descriptor. We use parameter values $k = 9$, $n_x, n_y = 3$, $n_t = 2$.

3.2. Spatio-temporal bag-of-features

Given a set of spatio-temporal features, we build a spatio-temporal bag-of-features (BoF). This requires the construction of a visual vocabulary. In our experiments we cluster a subset of $100k$ features sampled from the training videos with the k-means algorithm. The number of clusters is set to $k = 4000$, which has shown empirically to give good results and is consistent with the values used for static image classification. The BoF representation then assigns each feature to the closest (we use Euclidean distance) vocabulary word and computes the histogram of visual word occurrences over a space-time volume corresponding either to the entire video sequence or subsequences defined by a spatio-temporal grid. If there are several subsequences the different histograms are concatenated into one vector and then normalized.

In the spatial dimensions we use a *1x1* grid—corresponding to the standard BoF representation—, a *2x2* grid—shown to give excellent results in [9]—, a horizontal *h3x1* grid [12] as well as a vertical *v1x3* one. Moreover, we implemented a denser *3x3* grid and a center-focused *o2x2* grid where neighboring cells overlap by 50% of their width and height. For the temporal dimension we subdivide the video sequence into 1 to 3 non-overlapping temporal bins,

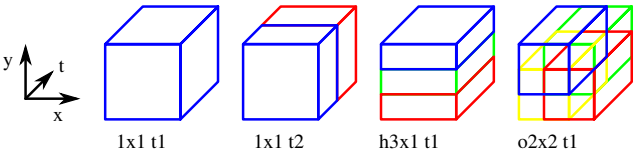

Figure 6. Examples of a few spatio-temporal grids.

resulting in *t1*, *t2* and *t3* binnings. Note that *t1* represents the standard BoF approach. We also implemented a center-focused *ot2* binning. Note that for the overlapping grids the features in the center obtain more weight.

The combination of six spatial grids with four temporal binnings results in 24 possible spatio-temporal grids. Figure 6 illustrates some of the grids which have shown to be useful for action recognition. Each combination of a spatio-temporal grid with a descriptor, either HoG or HoF, is in the following called a channel.

3.3. Non-linear Support Vector Machines

For classification, we use a non-linear support vector machine with a multi-channel χ^2 kernel that robustly combines channels [20]. We use the multi-channel Gaussian kernel defined by:

$$K(H_i, H_j) = \exp\big(-\sum_{c \in \mathcal{C}} \frac{1}{A_c} D_c(H_i, H_j)\big) \qquad (1)$$

where $H_i = \{h_{in}\}$ and $H_j = \{h_{jn}\}$ are the histograms for channel c and $D_c(H_i, H_j)$ is the χ^2 distance defined as

$$D_c(H_i, H_j) = \frac{1}{2} \sum_{n=1}^{V} \frac{(h_{in} - h_{jn})^2}{h_{in} + h_{jn}} \qquad (2)$$

with V the vocabulary size. The parameter A_c is the mean value of the distances between all training samples for a channel c [20]. The best set of channels $\mathcal{C}$ for a given training set is found based on a greedy approach. Starting with an empty set of channels all possible additions and removals of channels are evaluated until a maximum is reached. In the case of multi-class classification we use the one-against-all approach.

4. Experimental results

In the following we first evaluate the performance of the different spatio-temporal grids in section 4.1. We then compare our approach to the state-of-the-art in section 4.2 and evaluate the influence of noisy, i.e., incorrect, labels in section 4.3. We conclude with experimental results for our movie datasets in section 4.4

4.1. Evaluation of spatio-temporal grids

In this section we evaluate if spatio-temporal grids improve the classification accuracy and which grids perform best in our context. Previous results for static image classification have shown that the best combination depends on the class as well as the dataset [9, 12]. The approach we take here is to select the overall most successful channels and then to choose the most successful combination for each class individually.

As some grids may not perform well by themselves, but contribute within a combination [20], we search for the most successful combination of channels (descriptor & spatio-temporal grid) for each action class with a greedy approach. To avoid tuning to a particular dataset, we find the best spatio-temporal channels for both the KTH action dataset and our manually labeled movie dataset. The experimental setup and evaluation criteria for these two datasets are presented in sections 4.2 and 4.4. We refer the reader to these sections for details.

Figure 7 shows the number of occurrences for each of our channel components in the optimized channel combinations for KTH and movie actions. We can see that HoG descriptors are chosen more frequently than HoFs, but both are used in many channels. Among the spatial grids the horizontal *3x1* partitioning turns out to be most successful. The traditional *1x1* grid and the center-focused *o2x2* perform also very well. The *2x2*, *3x3* and *v1x3* grids occur less often and are dropped in the following. They are either redundant (*2x2*), too dense (*3x3*), or do not fit the geometry of natural scenes (*v1x3*). For temporal binning no temporal subdivision of the sequence *t1* shows the best results, but *t3* and *t2* also perform very well and complement *t1*. The *ot2* binning turns out to be rarely used in practice—it often duplicates *t2*—and we drop it from further experiments.

Table 2 presents for each dataset/action the performance of the standard bag-of-features with HoG and HoF descriptors, of the best channel as well as of the best combination of channels found with our greedy search. We can observe that the spatio-temporal grids give a significant gain over the standard BoF methods. Moreover, combining two to three

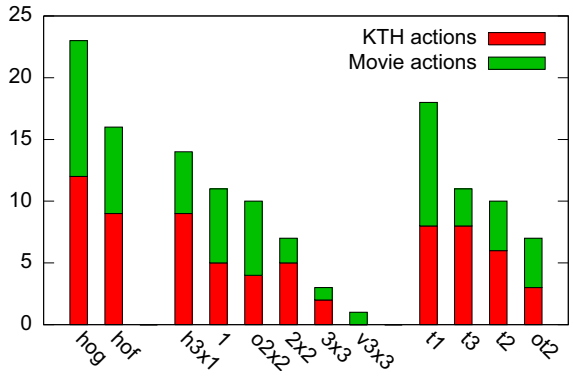

Figure 7. Number of occurrences for each channel component within the optimized channel combinations for the KTH action dataset and our manually labeled movie dataset.

Task	HoG BoF	HoF BoF	Best channel	Best combination
KTH multi-class	81.6%	89.7%	91.1% (hof h3x1 t3)	91.8% (hof 1 t2, hog 1 t3)
Action AnswerPhone	13.4%	24.6%	26.7% (hof h3x1 t3)	32.1% (hof o2x2 t1, hof h3x1 t3)
Action GetOutCar	21.9%	14.9%	22.5% (hof o2x2 1)	41.5% (hof o2x2 t1, hog h3x1 t1)
Action HandShake	18.6%	12.1%	23.7% (hog h3x1 1)	32.3% (hog h3x1 t1, hog o2x2 t3)
Action HugPerson	29.1%	17.4%	34.9% (hog h3x1 t2)	40.6% (hog 1 t2, hog o2x2 t2, hog h3x1 t2)
Action Kiss	52.0%	36.5%	52.0% (hog 1 1)	53.3% (hog 1 t1, hof 1 t1, hof o2x2 t1)
Action SitDown	29.1%	20.7%	37.8% (hog 1 t2)	38.6% (hog 1 t2, hog 1 t3)
Action SitUp	6.5%	5.7%	15.2% (hog h3x1 t2)	18.2% (hog o2x2 t1, hog o2x2 t2, hog h3x1 t2)
Action StandUp	45.4%	40.0%	45.4% (hog 1 1)	50.5% (hog 1 t1, hof 1 t2)

Table 2. Classification performance of different channels and their combinations. For the KTH dataset the average class accuracy is reported, whereas for our manually cleaned movie dataset the per-class average precision (AP) is given.

channels further improves the accuracy.

Interestingly, HoGs perform better than HoFs for all real-world actions except for answering the phone. The inverse holds for KTH actions. This shows that the context and the image content play a large role in realistic settings, while simple actions can be very well characterized by their motion only. Furthermore, HoG features also capture motion information up to some extent through their local temporal binning.

In more detail, the optimized combinations for sitting down and standing up do not make use of spatial grids, which can be explained by the fact that these actions can occur anywhere in the scene. On the other hand, temporal binning does not help in the case of kissing, for which a high variability with respect to the temporal extent can be observed. For getting out of a car, handshaking and hugging a combination of a *h3x1* and a *o2x2* spatial grid is successful. This could be due to the fact that those actions are usually pictured either in a wide setting (where a scene-aligned grid should work) or as a closeup (where a uniform grid should perform well).

The optimized combinations determined in this section, cf. table 2, are used in the remainder of the experimental section.

4.2. Comparison to the state-of-the-art

We compare our work to the state-of-the-art on the KTH actions dataset [15], see figure 8. It contains six types of human actions, namely walking, jogging, running, boxing, hand waving and hand clapping, performed several times by 25 subjects. The sequences were taken for four different scenarios: outdoors, outdoors with scale variation, outdoors with different clothes and indoors. Note that in all cases the background is homogeneous. The dataset contains a total of 2391 sequences. We follow the experimental setup of Schuldt et al. [15] with sequences divided into the training/validation set (8+8 people) and the test set (9 people). The best performing channel combination, reported in the previous section, was determined by 10-fold cross-validation on the combined training+validation set. Results are reported for this combination on the test set.

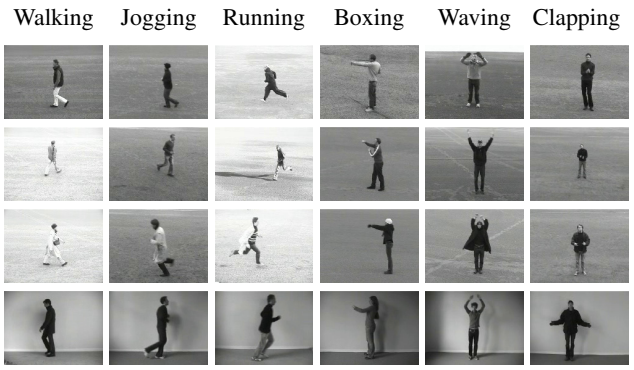

Figure 8. Sample frames from the KTH actions sequences. All six classes (columns) and scenarios (rows) are presented.

Table 3 compares the average class accuracy of our method with results reported by other researchers. Compared to the existing approaches, our method shows significantly better performance, outperforming the state-of-the-art in the same setup. The confusion matrix for our method is given in table 4. Interestingly, the major confusion occurs between jogging and running.

Method	Schuldt et al. [15]	Niebles et al. [13]	Wong et al. [18]	ours
Accuracy	71.7%	81.5%	86.7%	**91.8%**

Table 3. Average class accuracy on the KTH actions dataset.

	Walking	Jogging	Running	Boxing	Waving	Clapping
Walking	.99	.01	.00	.00	.00	.00
Jogging	.04	.89	.07	.00	.00	.00
Running	.01	.19	.80	.00	.00	.00
Boxing	.00	.00	.00	.97	.00	.03
Waving	.00	.00	.00	.00	.91	.09
Clapping	.00	.00	.00	.05	.00	.95

Table 4. Confusion matrix for the KTH actions.

Note that results obtained by Jhuang et al. [6] and Wong et al. [19] are not comparable to ours, as they are based on non-standard experimental setups: they either use more training data or the problem is decomposed into simpler tasks.

4.3. Robustness to noise in the training data

Training with automatically retrieved samples avoids the high cost of manual data annotation. Yet, this goes in hand with the problem of wrong labels in the training set. In this section we evaluate the robustness of our action classification approach to labeling errors in the training set.

Figure 9 shows the recognition accuracy as a function of the probability p of a label being wrong. Training for $p = 0$ is performed with the original labels, whereas with $p = 1$ all training labels are wrong. The experimental results are obtained for the KTH dataset and the same setup as described in subsection 4.2. Different wrong labelings are generated and evaluated 20 times for each p; the average accuracy and its variance are reported.

The experiment shows that the performance of our method degrades gracefully in the presence of labeling errors. Up to $p = 0.2$ the performance decreases insignificantly, i.e., by less than two percent. At $p = 0.4$ the performance decreases by around 10%. We can, therefore, predict a very good performance for the proposed automatic training scenario, where the observed amount of wrong labels is around 40%.

Note that we have observed a comparable level of resistance to labeling errors when evaluating an image classification method on the natural-scene images of the PASCAL VOC'07 challenge dataset.

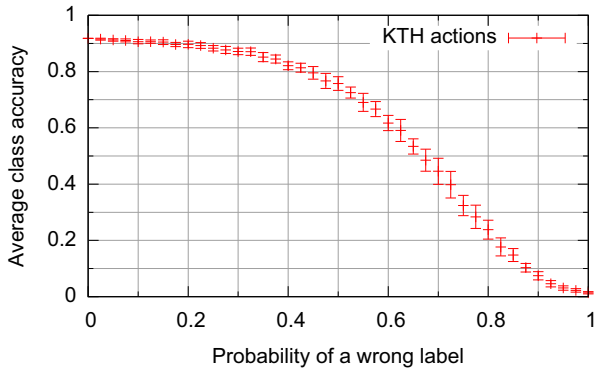

Figure 9. Performance of our video classification approach in the presence of wrong labels. Results are report for the KTH dataset.

4.4. Action recognition in real-world videos

In this section we report action classification results for real-word videos, i.e., for our test set with 217 videos. Training is performed with a clean, manual dataset as well as an automatically annotated one, see section 2.3 for details. We train a classifier for each action as being present or not following the evaluation procedure of [5]. The performance is evaluated with the average precision (AP) of the precision/recall curve. We use the optimized combination of spatio-temporal grids from section 4.1. Table 5 presents the AP values for the two training sets and for a random classifier referred to as chance AP.

	Clean	Automatic	Chance
AnswerPhone	32.1%	16.4%	10.6%
GetOutCar	41.5%	16.4%	6.0%
HandShake	32.3%	9.9%	8.8%
HugPerson	40.6%	26.8%	10.1%
Kiss	53.3%	45.1%	23.5%
SitDown	38.6%	24.8%	13.8%
SitUp	18.2%	10.4%	4.6%
StandUp	50.5%	33.6%	22.6%

Table 5. Average precision (AP) for each action class of our test set. We compare results for clean (annotated) and automatic training data. We also show results for a random classifier (chance).

The classification results are good for the manual training set and lower for the automatic one. However, for all classes except "HandShake" the automatic training obtains results significantly above chance level. This shows that an automatically trained system can successfully recognize human actions in real-world videos. For kissing, the performance loss between automatic and manual annotations is minor. This suggests that the main difficulty with our automatic approach is the low number of correctly labeled examples and not the percentage of wrong labels. This problem could easily be avoided by using a large database of movies which we plan to address in the future.

Figure 10 shows some example results obtained by our approach trained with automatically annotated data. We display key frames of test videos for which classification obtained the highest confidence values. The two top rows show true positives and true negatives. Note that despite the fact that samples were highly scored by our method, they are far from trivial: the videos show a large variability of scale, viewpoint and background. The two bottom rows show wrongly classified videos. Among the false positives many display features not unusual for the classified action, for example the rapid getting up is typical for "GetOutCar" or the stretched hands are typical for "HugPerson". Most of the false negatives are very difficult to recognize, see for example the occluded handshake or the hardly visible person getting out of the car.

5. Conclusion

This paper has presented an approach for automatically collecting training data for human actions and has shown that this data can be used to train a classifier for action recognition. Our approach for automatic annotation

Figure 10. Example results for action classification trained on the automatically annotated data. We show the key frames for test movies with the highest confidence values for true/false positives/negatives.

achieves 60% precision and scales easily to a large number of action classes. It also provides a convenient semi-automatic tool for generating action samples with manual annotation. Our method for action classification extends recent successful image recognition methods to the spatio-temporal domain and achieves best up to date recognition performance on a standard benchmark [15]. Furthermore, it demonstrates high tolerance to noisy labels in the training set and, therefore, is appropriate for action learning in automatic settings. We demonstrate promising recognition results for eight difficult and realistic action classes in movies.

Future work includes improving the script-to-video alignment and extending the video collection to a much larger dataset. We also plan to improve the robustness of our classifier to noisy training labels based on an iterative learning approach. Furthermore, we plan to experiment with a larger variety of space-time low-level features. In the long term we plan to move away from bag-of-features based representations by introducing detector style action classifiers.

Acknowledgments. M. Marszałek is supported by the European Community under the Marie-Curie project VISITOR. This work was supported by the European research project CLASS. We would like to thank J. Ponce and A. Zisserman for discussions.

References

[1] T. Berg, A. Berg, J. Edwards, M. Maire, R. White, Y. Teh, E. Learned Miller, and D. Forsyth. Names and faces in the news. In *CVPR*, 2004.

[2] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *CIVR*, 2007.

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

[4] M. Everingham, J. Sivic, and A. Zisserman. Hello! my name is... Buffy – automatic naming of characters in TV video. In *BMVC*, 2006.

[5] M. Everingham, L. van Gool, C. Williams, J. Winn, and A. Zisserman. Overview and results of classification challenge, 2007. The PASCAL VOC'07 Challenge Workshop, in conj. with ICCV.

[6] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *ICCV*, 2007.

[7] I. Laptev. On space-time interest points. *IJCV*, 64(2/3):107–123, 2005.

[8] I. Laptev and P. Pérez. Retrieving actions in movies. In *ICCV*, 2007.

[9] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[10] L. Li, G. Wang, and L. Fei Fei. Optimol: automatic online picture collection via incremental model learning. In *CVPR*, 2007.

[11] R. Lienhart. Reliable transition detection in videos: A survey and practitioner's guide. *IJIG*, 1(3):469–486, 2001.

[12] M. Marszałek, C. Schmid, H. Harzallah, and J. van de Weijer. Learning object representations for visual object class recognition, 2007. The PASCAL VOC'07 Challenge Workshop, in conj. with ICCV.

[13] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. In *BMVC*, 2006.

[14] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. In *ICCV*, 2007.

[15] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *ICPR*, 2004.

[16] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[17] J. Willamowski, D. Arregui, G. Csurka, C. R. Dance, and L. Fan. Categorizing nine visual classes using local appearance descriptors. In *IWLAVS*, 2004.

[18] S.-F. Wong and R. Cipolla. Extracting spatiotemporal interest points using global information. In *ICCV*, 2007.

[19] S.-F. Wong, T. K. Kim, and R. Cipolla. Learning motion categories using both semantic and structural information. In *CVPR*, 2007.

[20] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *IJCV*, 73(2):213–238, 2007.

[21] T. Zhang. Large margin winnow methods for text categorization. In *KDD-2000 Workshop on Text Mining*, 2000.