Modeling and visual recognition of human actions and interactions

Ivan Laptev

Habilitation à diriger des recherches

École Normale Suprieure

Spécialité: Informatique

Ivan Laptev

Modeling and visual recognition of human actions and interactions

Defended on July 3, 2013

Rapporteurs:

M. David **FORSYTH**	University of Illinois at Urbana Champaign
Mme. Michal **IRANI**	The Weizmann Institute of Science
M. Luc **VAN GOOL**	Eidgenössische Technische Hochschule Zürich

Membres du jury:

M. Francis **BACH**	INRIA
M. Alexei A. **EFROS**	Carnegie Mellon University
M. Patrick **PÉREZ**	Technicolor
M. Jean **PONCE**	ENS
Mme. Cordelia **SCHMID**	INRIA

Contents

Chapter 1

Introduction

This thesis addresses the automatic interpretation of dynamic scenes from visual observations. Its particular focus is on recognition of human actions, objects and human-object interactions. Before detailed description of the work in Chapters 2-5, this chapter motivates our work, outlines the challenges, summarizes contributions, and reviews related work in Sections 1.1-1.4.

1.1 Motivation

Data. The amount of video has increased dramatically over recent years and continues to grow. Striking examples include 4331 years of video uploaded on YouTube in 2011[1] and millions of CCTV cameras installed only in the UK. National TV archives also own large amounts of video, such as British BBC with 600K hours (68 years) of video[2] and French INA[3] with 900K hours (102 years) of video and audio content. The introduction of new devices such as wearable cameras and Google Glass will likely accelerate the growth of video even further. According to Cisco[4], video is expected to dominate Internet traffic by 91% in 2014.

While the growth of video continues, its access becomes critically dependent on efficient search engines. Current systems partly solve video search with text-based indexing using, e.g., tags for YouTube videos or program schedules for TV broadcasts. Most of the information in video, however, remains "hidden", and its extraction requires new methods for automatic video analysis.

Content. People are the most frequent and, arguably, most important objects in images and video. A simple test[5] reveals that about 35% of screen pixels in movies, TV programs and YouTube videos belong to people. Photographs on Flickr.com contain about 25% of person pixels. These numbers indicate that the visual data we tend to produce, share and consume is strongly biased towards people. Notably, the percentage of "person pixels" in the field of view of our daily life is only about 4%[6]. This further indicates that the strong person bias in consumer videos and images

[1] www.youtube.com/t/press_statistics

[2] www.bbc.co.uk/archive/tv_archive.shtml

[3] www.ina-sup.com/collections/linatheque-de-france-1

[4] http://newsroom.cisco.com/dlls/2010/prod_060210.html

[5] The test has been done by the author and has involved manual segmentation of people in 100 randomly selected frames from movies, popular YouTube videos and one month of TV stream.

[6] The statistics for daily life images was obtained by hand-segmenting people in random frames from the first-person view wearable camera dataset [43]. Body parts of the person wearing the camera were excluded.

is not natural but is caused by the content production.

The strong bias of video towards people clearly motivates the need of automatic methods that interpret person pixels. Meanwhile, it is also interesting to identify the origin of this bias and to understand what information in person pixels is particularly important. Why do people choose to watch other people? Why do people watch video at all? An answer to a related question "Why do people read books?" might be that *books teach us new patterns of behaviour.*[7] Indeed, the experience of our lives is limited by time, geographic location, and surrounding people. Books, as well as movies, TV series and other types of narrations can be regarded as a shared collective experience describing possible actions and reactions of people in new situations as well as consequences thereof. Learning how to act and react in new situations by observing other people, hence, might be a plausible answer to the above questions. Such an answer explains well the strong person bias in video, and it also implies that human actions are likely to be among the most important elements of the video content.

Applications. Analyzing people and human actions in video is of high interest for many applications within entertainment, education, health care, security and other areas. Human motion capture is widely used to animate human characters in movies and video games. Biometric identification involves the recognition of people by facial characteristics, finger-prints, iris scans or gate patterns. Typical video surveillance tasks include detection of intrusion, violence and other abnormal events. Off-line video analytics and recognition of human actions, facial expressions and gaze patterns can be used to analyze person behaviour in retail applications and sociological studies. Motion analysis of large groups of people is useful to prevent crowd stampedes, to count participants of a demonstration, or to enhance urban planning. Automatic annotation of people and human actions in video would greatly improve the search capabilities of public and personal video archives. Recognition of daily activities is becoming important for assisted living and elderly care applications.

Looking further into the future, I believe computers will be able to analyze and learn from the immense amount of collective human behaviour. The models of behaviour learned from the collective experience will stretch far beyond capabilities of individual people. Using these models, computers will be able to increase our efficiency and enrich our daily lives by, for example, assisting us in preparing a new meal we have never tasted before, helping us to pose the right questions and analyze reactions of people at job interviews, instantly instructing us what to do in a car accident, teaching us how to manipulate a kite-surf, or telling us where we have left the keys yesterday night when we are about to rush to work in the morning. Before this can happen, however, many challenges must be addressed.

1.2 Challenges

Representation. Visual data has a huge *variability*. The appearance of objects and people in still images and in video is greatly influenced by changes of lighting, camera viewpoints, object shape and backgrounds, by occlusions, non-rigid object deformations, within-class object variations, aging, seasonal changes, and many other factors. Specific to video, dynamic appearance is for example influenced by camera motion, relative object displacements, and non-rigid motions. Specific to human actions, video content varies greatly due to many existing types of actions

[7] The answer suggested in the interview with Olga Slavnikova, a contemporary Russian author.

and interactions, due to different action styles, people clothing, and types of manipulated objects (Figure 1.1). Modeling of some of these variations (e.g., the shape of an object, the action of a person) independently of the others (e.g., camera view-points and occlusions) is a major challenge of computer vision.

Figure 1.1: Human actions in movies. (a): Many different activities are depicted in films. (b): Within-class variations of actions due to different body motions, manipulated objects and clothes.

Learning. Computer vision aims to solve the inverse and ill-posed problem of determining the cause of visual observations. The relation between visual observations and their cause can sometimes be modeled analytically as, for example, in motion estimation under the assumption of brightness constancy [40]. In most cases, however, the observation-cause relations are too complex for analytical modeling. Statistical modeling and machine learning methods can be used instead in such cases. The relations between visual observations x and underlying properties y can be modeled in the form of joint distributions $p(x, y)$ (generative models) or conditional distributions $p(y|x)$ (discriminative models) [55]. Standard methods to estimate (or learn) such distributions, however, typically require a large number of training samples (x_i, y_i) where x_i, for example, could be a video depicting some action, and y_i the corresponding class label. Given the large variety in both x and y, manual collection and annotation of sufficiently many training samples (x_i, y_i) is prohibitive in general. A major challenge is, hence, to design new learning methods able to derive $p(y|x)$ or $p(y, x)$ from large amounts of *readily-available but incomplete and imprecise supervision* such as the scripts accompanying movies or TV series, images with corresponding image tags and other types of meta-data.

Action vocabulary. Visual recognition is often regarded as a labeling problem, i.e., assigning objects and actions in images or video with their class labels such as "airplane", "cat", "walking", "shaking hands", etc. While the object vocabulary is well defined for thousands of object categories [3], a vocabulary of activities for action recognition is much less developed. Scaling up current action recognition datasets beyond tens of action classes may not be just a practical issue: On the one hand, action verbs are often quite ambiguous as illustrated by different examples of "open" action in Figure 1.2. On the other hand, the same type of actions may be expressed in many different ways, e.g., "he gets out of the Chevrolet", "she exits her new truck" for "getting out of car" action.

Figure 1.2: Examples of the "open" action illustrate ambiguity of the verb open.

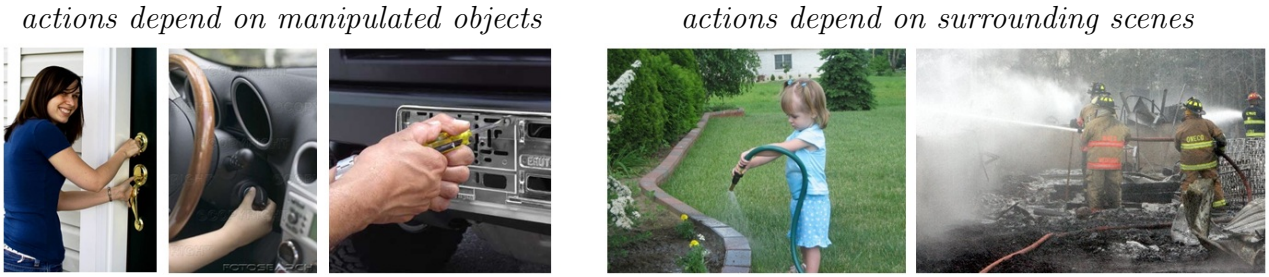


Figure 1.3: Left: Similar body motions (e.g., a twist of a hand) may imply very different actions depending on the manipulated objects. Right: Similar person-object interactions may imply different actions depending on the global scene context.

Figure 1.4: Examples of unusual (left) and common (right) person-scene interactions.

The difficulty of defining an action vocabulary compared to an object vocabulary may have several reasons. First, many actions have compositional structure and exhibit a large variability depending on manipulated objects, scene structure, etc., as illustrated in Figure 1.3. Second, actions typically exist over a limited period of time and never repeat in exactly the same way. Third, unlike spatial object boundaries, the temporal boundaries of actions may often lack a precise definition. The challenge of defining an action vocabulary has also been recognized in natural language processing where recent studies explore e.g., the problem of *grounding* natural language to robot control commands [53].

Establishing a proper link between actions and their linguistic descriptions is an important task since language is frequently used to describe and search images and video. Describing actions by words, however, may not always be necessary and may make solutions to some visual tasks unnecessary complicated. For example, the detection of unusual events (Figure 1.4) could potentially be solved by analysing human poses, relative positions and orientations of objects and people, recognizing object function, etc. Finding prototypical linguistic expressions for unusual events, on the other hand, may not be easy or necessary for this task. Non-linguistic description of actions by the properties of person-object and person-scene interaction is, hence, another interesting and challenging topic in action recognition.

1.3 Contributions

This work addresses several of the challenges outlined above. Its main focus is on recognizing actions and interactions in realistic video settings such as movies and consumer videos. The first contribution of this thesis (Chapters 2 and 4) is concerned with new *video representations for action recognition*. We introduce local space-time descriptors and demonstrate their potential to classify and localize actions in complex settings while circumventing the difficult intermediate steps of person detection, tracking and human pose estimation. The material on bag-of-features action recognition in Chapter 2 is based on publications [L14, L22, L23][8] and is related to other work by the author [L6, L7, L8, L11, L12, L13, L16, L21]. The work on object and action localization in Chapter 4 is based on [L9, L10, L13, L15] and relates to [L1, L17, L19, L20].

The second contribution of this thesis is concerned with *weakly-supervised action learning*. Chapter 3 introduces methods for automatic annotation of action samples in video using readily-available video scripts. It addresses the ambiguity of action expressions in text and the uncertainty of temporal action localization provided by scripts. The material presented in Chapter 3 is based on publications [L4, L14, L18]. Finally Chapter 5 addresses interactions of people with objects and concerns *modeling and recognition of object function*. We exploit relations between objects and co-occurring human poses and demonstrate object recognition improvements using automatic pose estimation in challenging videos from YouTube. This part of the thesis is based on the publication [L2] and relates to other work by the author [L3, L5].

1.4 Related work

Video representations for action recognition. The models of video content proposed in this thesis build on local space-time descriptors for video [L8, L12] extending local descriptors for still images [51, 54]. Related work [5, 70, 71] has explored the matching of actions based on the local space-time structure of the video. In particular, Boiman at al. [5] has developed a compositional model of actions in space-time for detecting unusual events. Our bag-of-features action model in Chapter 2 is simpler, but enables recognition of diverse actions in realistic settings.

Bag-of-features representations have been successfully applied to image recognition tasks [12, 49, 73, 90]. In video, early statistical action representations have been proposed in [9, 88]. Since the work presented in [L21], several other methods have explored bag-of-features representations for action recognition [16, 42, 56]. The work presented in Chapters 2 and 4 was among the first to address action recognition in realistic settings outside controlled laboratory setups. Since then, several follow-up studies have been presented. Regarding action classification, [28, 57] have addressed more elaborated modeling of temporal structure of actions. Action representations in terms of pre-trained features such as action attributes and "action banks" have been explored in [50, 66]. New local motion descriptors such as short point trajectories and motion boundary histograms [80] have been shown to provide high recognition results in a number of recent benchmarks. Concerning action localization, [47] has addressed joint person detection and action localization in movies.

Weakly-supervised learning of actions. Weakly-supervised learning has been addressed in the context of automatic annotation of still images with keywords [1, 32, 82] or labeling faces with names in news [2]. Berg *et al.* [2] label detected faces in news photographs with names of people obtained from text captions. Recent work has looked at learning spatial relations (such as "on

[8] References indicated by [L#] point to the work of the author. References [#] point to the work by others.

top of") from prepositions [33] or generating entire sentence-level captions for images [21, 60]. A generative model of faces and poses (such as "hit backhand") is learnt in [52] from names and verbs in manually provided captions for news photographs. While the goal of this work is related to ours, we focus on learning from video with sparse, noisy and imprecise annotations extracted from a script. To deal with the imprecise temporal localization of actions, Chapter 3 develops a new discriminative weakly-supervised method for temporal clustering of actions. Action clustering has been addressed with generative methods, e.g., in [89], but for much simpler setups than ours. Our weakly supervised clustering is also related to the work on learning object models from weakly annotated images [10, 48]. The temporal localization of training samples in videos, addressed in this work, is similar in spirit to the weakly supervised learning of object part locations for object detection [24] and temporal localization of action parts [28, 57, 76]. In video, manually provided text descriptions have been used to learn a causal model of human actions in the constrained domain of sports events [34]. Others have looked at learning character classifiers from videos with readily-available text [11, 18, 72], our work described in Chapter 3 was among the first to consider weakly-supervised learning of actions in video. In parallel to our work, Buehler et al. [6] have proposed learning sign language from weak TV video annotations using multiple instance learning.

Recognition of object function. The interplay between people and objects has recently attracted significant attention. Interactions between people and objects have been studied in still images with the focus on improving action recognition [34],[L3], object localization [15, 34, 75] and discovery [64] as well as pose estimation [86, 87]. In video, constraints between human actions and objects (e.g., drinking from a coffee cup) have been investigated in restricted laboratory setups [29, 34, 45] or ego-centric scenarios [22]. In both still images and video the focus has been typically on small objects manipulated by hands (e.g., coffee cups, footballs, tennis rackets) rather than larger objects such as chairs, sofas or tables, which exhibit large intra-class variability. In addition, manual annotation of action categories [34, 45] or human poses [86] in the training data is often required.

Functional scene descriptions have been developed for surveillance setups, e.g., [63, 77, 81], but the models are usually designed for specific scene instances and use only coarse-level observations of object/person tracks [77, 81], or approximate person segments obtained from background subtraction [63]. In contrast, the method proposed in Chapter 5 generalizes to new challenging scenes, and uses finer grain descriptors of estimated body configuration enabling discrimination between object classes such as sofas and chairs.

Recent related work [31, 35] has inferred functions or affordances, i.e., types of actions supported by objects or scenes (chair: sitting) [30] from automatically obtained noisy 3D reconstructions of indoor scenes. These methods infer affordance based on the geometry and physical properties of the space. For example, they find places where a person *can* sit by fitting a 3D human skeleton in a particular pose at a particular location in the scene. While people can sit at many places, they tend to sit in sofas more often than on tables. Moreover, they may sit on sofas in a different way than on a floor or on a chair. In our work we aim to leverage these observations and focus on *statistical affordances* by learning typical human poses associated with each object.

In a setup similar to ours, Fouhey *et al.* [L5] have looked at people's actions as a cue for a coarse 3D box-like geometry of indoor scenes. Here we investigate the interplay between object function and object classes, rather than scene geometry. While in [L5] the geometric person-scene relations are designed manually, we here learn semantic person-object interactions from data.

Chapter 2

Video representations for action recognition

As discussed in the Introduction, the appearance of actions in video is very variable (Figure 1.1). Addressing this variability explicitly through detailed understanding of scene structure (e.g., locations and properties of objects, pose and motion of people, scene geometry, etc.) would have clear benefits. Unfortunately, the reliable interpretation of scene details from visual data is still a hard challenge for automatic methods. This chapter addresses the problem from another direction and attempts to model actions without explicit assumptions on the scene structure. While such an approach is expected to have some disadvantages, it benefits from being conceptually simple and direct, for example, by eliminating error accumulation of intermediate steps (person and object detection, tracking, pose estimations, etc.). The proposed approach currently provides state-of-the-art results for action recognition in challenging and realistic settings. As shown at the end of this chapter, our approach permits the gradual introduction of scene structure leading to further improvements. This chapter is based on the work done in collaboration with students Marcin Marszałek, Muneeb Ullah and Heng Wang and was published in [L14, L22, L23].

2.1 Bag-of-features video classification

We represent videos by collections of local space-time patches. The intuition behind the approach is illustrated in Figure 2.1, where videos of three simple actions are represented with space-time surfaces obtained by thresholding gray-value. One can observe that each sequence has local space-time neighborhoods with action-characteristic structure. Building action representations based on local space-time neighborhoods, hence, has the promise of being both discriminative and resistant to global variations of actions and the background. The method is also closely related to image representations in terms of local image patches that have been shown successful for image classification tasks [62, 90].

2.1.1 Space-time features

To select characteristic space-time neighborhoods, we follow our previous work [L8] and detect interest points using a space-time extension of the Harris operator. Rather than performing scale selection as in [L8], we use a multi-scale approach and extract features at multiple levels of spatio-temporal scales (σ_i^2, τ_j^2). The detected interest points are illustrated in Figure 2.1.

To characterize the 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 $\Delta_x, \Delta_y = 2\alpha\sigma$, $\Delta_t = 2\alpha\tau$. 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) [13, 14]. 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 $\alpha = 9$, $n_x, n_y = 3$, $n_t = 2$.

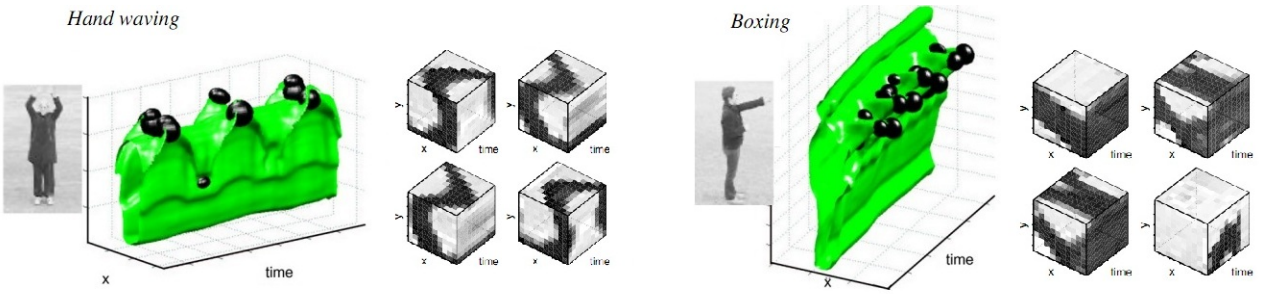


Figure 2.1: Examples of space-time interest points (STIP) [L8] extracted for video sequences with two human actions. One image for each video is shown together with a level-surface of image brightness illustrating the evolution of actions in space-time. Example space-time patches extracted at STIP locations are shown to the right of each action.

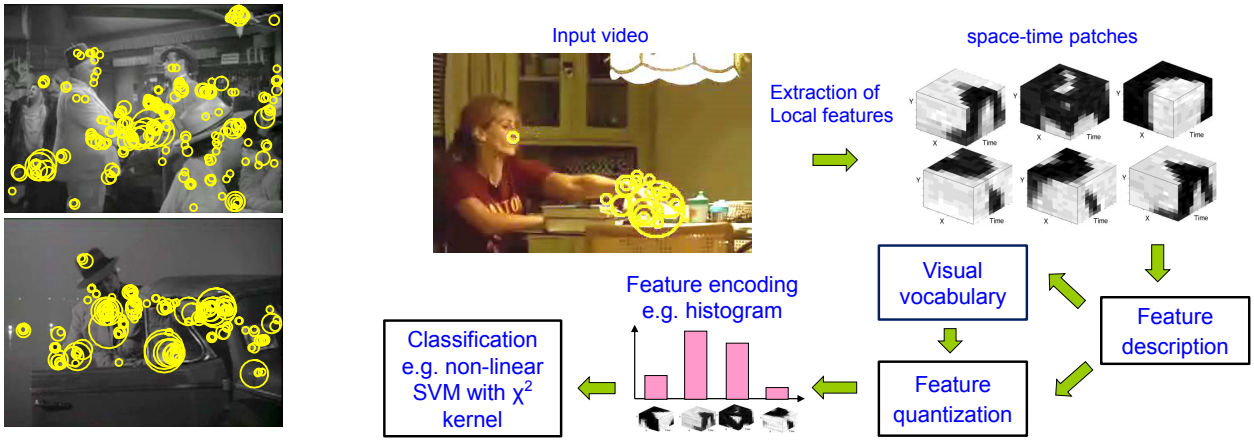


Figure 2.2: **Left:** Space-time interest points detected for two movie samples with "hand shake" and "get out car" actions. **Right:** Overview of the bag-of-features action classification pipeline.

2.1.2 Spatio-temporal bag-of-features

Given a set of spatio-temporal features, we build a spatio-temporal bag-of-features (BOF) [12]. 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.

To capture the coarse location of local features, we extend the spatial pyramid of [49] and subdivide videos using spatio-temporal grids. In the spatial dimensions we use a 1×1, 2×2, 3×3, horizontal $h3\times1$ as well as vertical $v1\times3$ grids. We have also implemented a center-focused $o2\times2$ grid where neighboring cells overlap by 50%. For the temporal dimension we subdivide the video sequence into 1 to 3 non-overlapping temporal bins, resulting in $t1$, $t2$ and $t3$ binnings (Figure 2.3). We have also implemented a center-focused $ot2$ binning. The combination of six spatial grids with four temporal binnings results in 24 possible spatio-temporal grids. Each combination of a spatio-temporal grid with a descriptor, either HOG or HOF, is in the following called a channel.

2.1.3 Action classification

For classification, we use a non-linear support vector machine with a multi-channel χ^2 kernel that robustly combines channels [90]. We use the multi-channel Gaussian kernel defined by:

$$K(H_i, H_j) = \exp\Big(-\sum_{c\in\mathcal{C}} \frac{1}{A_c} D_c(H_i, H_j)\Big) \tag{2.1}$$

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. The parameter A_c is the mean value of the distances between all training samples for a channel c [90]. 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.

Results. We test our methods on the task of action classification in the Hollywood-2 dataset collected from action samples in movies (Section 3.2.1). The dataset contains 12 action classes with large and natural variability of action samples. The results are illustrated in Table 2.1 in terms of per-class average precision and overall mean average precision values. While the performance of individual grid channels is similar to each other, their combination improves the BOF channel by 3.4%. The performance is significantly higher than chance for all action classes.

	Comb	h3x1 t2	2x2 t3	1 t2	2x2 t2	1 t3	h3x1 t3	v3x1 t3	h3x1 1	v3x1 t2	2x2 1	1 1 (bof)	3x3 t3	v3x1 1	3x3 t2	o2x2 t3	3x3 1	o2x2 t2	Chance
mAP	**50.7**	**48.8**	48.7	48.6	48.6	48.4	48.3	48.0	48.0	47.5	47.4	47.3	47.1	47.1	46.9	46.3	46.3	45.1	9.2
AnswerPhone	**20.9**	22.1	23.0	20.7	16.6	**23.4**	22.5	21.1	12.1	19.4	12.4	15.7	19.8	13.8	15.6	23.3	12.0	19.2	7.2
DriveCar	**84.6**	83.9	83.2	84.5	84.1	84.2	83.4	83.1	86.3	83.6	86.1	**86.6**	81.8	85.5	82.1	83.5	84.7	83.5	11.5
Eat	**67.0**	63.5	63.9	62.6	61.9	62.4	**65.1**	62.9	63.4	64.0	63.8	59.5	62.9	62.3	62.8	58.7	64.7	57.7	3.7
FightPerson	**69.8**	69.0	66.3	68.8	66.2	68.6	67.5	66.3	**71.7**	67.3	69.7	71.1	64.3	69.8	65.7	67.2	68.7	68.0	7.9
GetOutCar	**45.7**	38.1	44.8	42.2	40.2	44.4	41.8	**45.3**	34.6	39.0	35.1	29.3	42.2	35.0	37.2	43.7	30.0	37.0	6.4
HandShake	**27.8**	24.5	23.7	21.3	25.0	20.4	21.4	24.9	21.2	25.6	24.0	21.2	25.2	25.1	25.3	22.1	26.3	20.9	5.1
HugPerson	**43.2**	38.4	39.5	36.3	40.4	36.0	35.3	37.8	40.4	**40.8**	37.7	35.8	36.0	38.5	37.0	34.5	37.6	33.3	7.5
Kiss	**52.5**	50.7	49.0	52.3	52.4	48.7	48.8	46.1	**54.4**	49.6	52.6	51.5	47.6	50.3	51.4	44.9	52.6	44.3	11.7
Run	**67.8**	65.7	60.5	67.5	63.2	64.3	62.8	60.8	68.3	63.8	66.4	**69.1**	57.8	67.3	61.1	63.6	65.5	66.0	16.0
SitDown	**57.6**	52.8	50.3	54.0	52.6	53.5	53.1	51.5	53.9	51.3	53.4	**58.2**	53.0	53.1	52.9	50.5	51.6	51.0	12.2
SitUp	**17.2**	21.5	25.6	17.2	**27.2**	19.4	20.5	23.1	19.7	14.4	20.7	17.5	21.1	14.6	20.2	10.6	15.4	8.3	4.2
StandUp	**54.3**	55.6	54.9	56.0	53.1	55.3	**57.2**	53.1	49.8	51.2	47.6	51.7	53.6	49.3	51.8	53.3	46.6	51.8	16.5

Table 2.1: Results of action classification on the Hollywood-2 dataset (clean training subset). Results are reported in terms of per-class and per-channel average precision values (%). Mean average precision (mAP) is shown in the top row. 17 best single channels are shown sorted in the decreasing order of mAP. Results for the 24-channel combination are shown in the first column. The chance performance is given in the last column

Figure 2.3: (Left): Examples of spatio-temporal grids. (Right): Sample frames from video sequences of KTH (top), UCF Sports (middle), and Hollywood-2 (bottom) human action datasets.

2.2 Evaluation of local features for action recognition

Since our work [L8, L14, L21], several variants of local video features have been proposed. This section evaluates the performance of space-time feature detectors and descriptors presented prior to the publication of [L23]. We evaluate three spare detectors: Harris3D [L8], Cuboids [16] and Hessian [83] as well as the "dense detector" corresponding to the uniform sampling of space-time patches. We also evaluate several local descriptors: HOG, HOF and their concatenation HOGHOF (see previous section), HOG3D [46], Cuboids [16] and ESURF [83]. The evaluation is performed on three action datasets: KTH actions [L21], UCF sports [65] and Hollywood-2 [L18] illustrated in Figure 2.3(Right). We have used BOF histogram feature encoding (no space-time grids) and SVM with χ^2 kernel for classification as described in the previous section.

Results of the evaluation are summarized in Table 2.2. Among the main conclusions we notice that the dense detector outperforms sparse detectors on the realistic UCF and Hollywood-2 datasets, probably due to the action-characteristic information in the background. Among descriptors, we notice the relatively low performance of the HOG descriptor indicating the importance of motion information captured by other descriptors.

KTH actions dataset

	HOG3D	HOGHOF	HOG	HOF	Cuboids	ESURF
Harris3D	89.0%	91.8%	80.9%	**92.1%**	–	–
Cuboids	90.0%	88.7%	82.3%	88.2%	89.1%	–
Hessian	84.6%	88.7%	77.67%	88.6%	–	81.4%
Dense	85.3%	86.1%	79.0%	88.0%	–	–

UCF sports dataset

	HOG3D	HOGHOF	HOG	HOF	Cuboids	ESURF
Harris3D	79.7%	78.1%	71.4%	75.4%	–	–
Cuboids	82.9%	77.7%	72.7%	76.7%	76.6%	–
Hessian	79.0%	79.3%	66.0%	75.3%	–	77.3%
Dense	**85.6%**	81.6%	77.4%	82.6%	–	–

Hollywood-2 dataset

	HOG3D	HOGHOF	HOG	HOF	Cuboids	ESURF
Harris3D	43.7%	45.2%	32.8%	43.3%	–	–
Cuboids	45.7%	46.2%	39.4%	42.9%	45.0%	–
Hessian	41.3%	46.0%	36.2%	43.0%	–	38.2%
Dense	45.3%	**47.4%**	39.4%	45.5%	–	–

Table 2.2: Results of action classification for various detector/descriptor combinations. Average accuracy is reported for KTH and UCF datasets. Mean average precision (mAP) is reported for Hollywood-2 dataset.

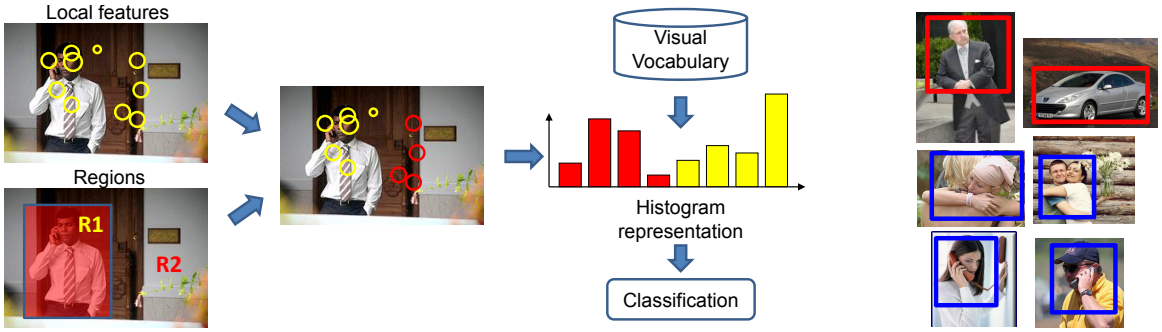


Figure 2.4: (Left): Schematic illustration of bag-of-features models with non-local cues. (Right): Examples of object and action regions used in this work.

2.3 Bag-of-features action recognition with non-local cues

Bag-of-features methods represent videos by order-less collections of local patches and, hence, lack information about the global structure of actions and scenes. The space-time grids introduced in Section 2.1.2 address this issue and encode rough feature positions in the global coordinate frame of a video. Here we refine this idea and introduce "semantic coordinate frame" with the aim of encoding feature positions with respect to particular objects or regions of the scene. We rely on a set of methods for segmenting the video into regions and compute bag-of-features representations (channels) separately for each type of such regions. A schematic illustration of our method is presented in Figure 2.4(Left). We consider four methods of region segmentation as described next.

Foreground/background motion segmentation. Foreground regions are more likely to belong to the action. We segment the video into foreground and background regions using motion-based video segmentation using [58]. We generate four pairs of fg/bg masks by thresholding the estimated foreground likelihood maps using four threshold values. The resulting regions define eight feature channels referred to here by *Motion8*.

Action detection The global appearance of actions in a single frame may provide complementary cues to local action descriptors. We thus train static detectors for action classes of the Hollywood-2 dataset using object detector [24]. For each class we collect positive training images from Internet image search (Figure 2.5). Detection bounding boxes obtained on video frames define action channels referred to here as *Action12*.

Person and object detection Similar to action detection, we use still image detectors for people and objects to define person and object regions in the video. We use the Calvin upper-body detector [7] to find person regions in the video. We also use object detectors [24] trained on the Pascal VOC2008 dataset. We apply multiple detection thresholds to each detector type and obtain person and object regions with the corresponding bag-of-features channels different thresholds to each detector to obtain bag-of-feature channels denoted by *Person12* and *Objects12* respectively.

Experimental results. The results of our method are summarized in Table 2.3. We evaluate action recognition on the Hollywood-2 dataset and present results for different combination of channels defined above. The first two columns in Table 2.3 correspond to the bag-of-features (BOF) and space-time grid (STGrid24) baselines presented in Section 2.1. Each type of new channel obtains slightly better performance compared to STGrid24. Channel combinations using multi-channel kernel (2.1), on the other hand, result in significant improvements. In particular we note that

Figure 2.5: Sample images collected from the Internet and used to train action detectors.

Channels	BOF	STGrid24	Action12	STGrid24 + Action12	STGrid24 + Motion8 + Action12 + Person12 + Objects12
mean AP	0.486	0.518	0.528	**0.557**	0.553
AnswerPhone	0.157	0.259	0.208	**0.263**	0.248
DriveCar	0.876	0.859	0.869	0.865	**0.881**
Eat	0.548	0.564	0.574	0.592	**0.614**
FightPerson	0.739	0.749	0.757	0.762	**0.765**
GetOutCar	0.334	0.440	0.383	0.457	**0.474**
HandShake	0.200	0.297	0.457	**0.497**	0.384
HugPerson	0.378	**0.461**	0.408	0.454	0.446
Kiss	0.521	0.550	0.560	0.590	**0.615**
Run	0.711	0.694	0.732	0.720	**0.743**
SitDown	0.590	0.589	0.596	**0.624**	0.613
SitUp	0.239	0.184	0.241	**0.275**	0.255
StandUp	0.533	0.574	0.549	0.588	**0.604**

Table 2.3: Per-class AP performance by different channels and channel combinations.

the classification of "hand shake" class improves from 20% (BOF) to 49.7% (STGrid24+Action8). This improvement correlates with the good performance of the static hand shake detector in our experiments.

2.4 Summary

This chapter has described an approach to action recognition based on a statistical video representation in terms of local features. The method does not rely on the intermediate steps (e.g., person detection and tracking), and achieves encouraging results for very challenging video data. Our evaluation of alternative local features on the final task of action recognition emphasizes the advantage of densely sampled motion descriptors. While some structural information has been introduced to bag-of-features representations, the method is still lacking the ability of explicitly reasoning about the spatial structure and the causality of dynamic scenes. Future extensions in this direction will likely improve performance of action recognition and will provide interpretations of people, their motion trajectories, scene layouts, objects, etc.

Chapter 3

Weakly-supervised learning of actions

Learning methods for human action recognition described in the previous chapter require many video samples of human actions with corresponding action labels. The collection of action samples by recording staged actions has been attempted in the past, e.g., to compose the KTH [L21] and Weizmann [4] datasets. Such an approach, however, typically fails to represent the rich variety of actions given the difficulty of sampling many realistic environments and involving many skilled actors. Existing video resources such as movies, TV series and YouTube, on the other hand, already contain a large number of realistic action samples which could be used for training. One way to collect action samples from these resources would be through manual annotation. Unfortunately, manual video annotation is difficult to scale beyond a few action classes given the rare occurrence of the majority of the classes. For example, the actions "Hand Shake" and "Answer Phone" appear only 2-3 times per movie on average[1].

This chapter explores the use of *video scripts* as a source of weak and automatic video supervision. First, Section 3.1 presents a method for automatic annotation of actions in movies using scripts. Section 3.2 extends this method to the automatic annotation of scenes and shows the benefit of joint action and scene recognition. Finally, Section 3.3 addresses the problem of temporal uncertainty in script-based video annotation and describes a discriminative clustering method for learning human actions. This chapter is based on the work done in collaboration with students Marcin Marszałek and Olivier Duchenne and was published in [L4, L14, L18].

3.1 Learning actions from movie scripts

Video scripts are publicly available for hundreds of popular movies[2] 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. [18]. Here we extend this idea and apply text-based script search to automatically collect video samples for human actions.

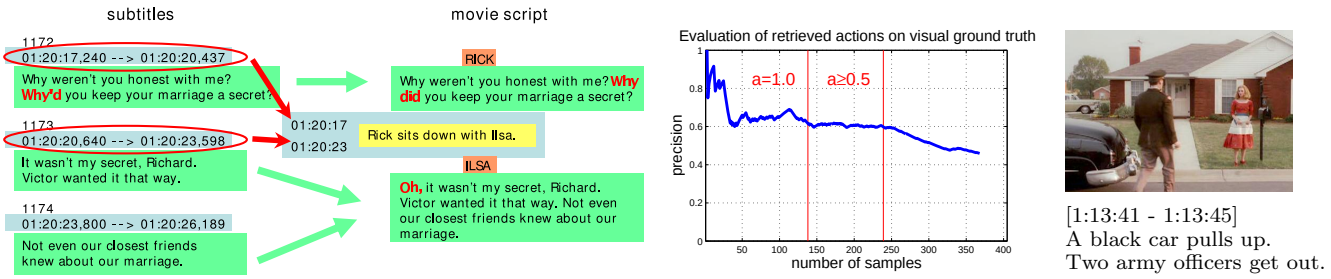
The automatic annotation of human actions from scripts, however, is not straightforward. First, scripts usually come without time information and have to be aligned with the video. Second, actions described in scripts do not always correspond with the actions in movies. Third, action retrieval has to cope with the substantial variability of action expressions in text. We address these problems in Sections 3.1.1 and 3.1.2 and show action classification results on the automatically collected video dataset in Section 3.1.3.

[1] The statistics was obtained by manually annotating "Hand Shake" and "Answer Phone" actions in 69 movies.

[2] Resources of movie scripts include `www.dailyscript.com`, `www.movie-page.com` and `www.weeklyscript.com`.

3.1.1 Alignment of actions in scripts and video

Movie scripts are typically available in plain text format and share a similar structure. We use line indentation as a simple feature to parse scripts into monologues, character names and scene descriptions (Figure 3.1(a),right). To align scripts with the video we follow [18] and use time information available in movie subtitles that we separately download from the Web. Similar to [18] 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 3.1(a). Video clips used for action training and classification in this work 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. This "a-score" is computed as the ratio of matched words in the near-by monologues as, $a = (\#matched\ \ words)/(\#all\ \ words)$.



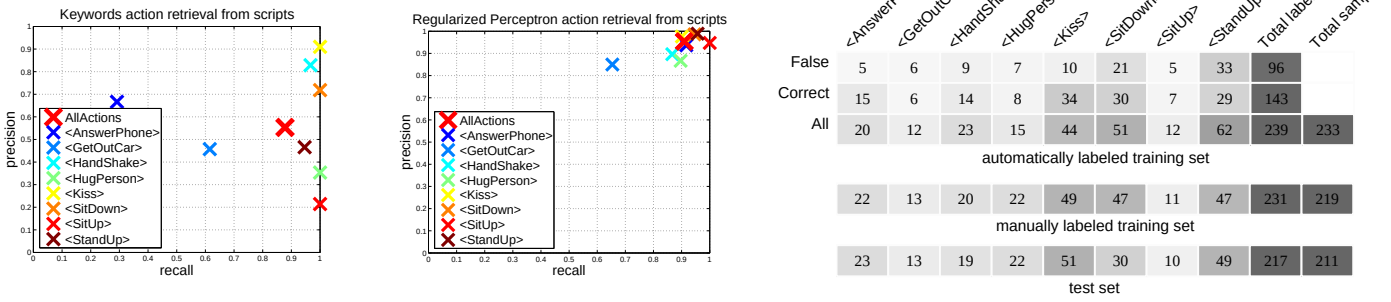
(a) (b) (c)

Figure 3.1: (a): 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). (b): Precision of script-based action annotation in video evaluated on visual ground truth. (c): Example of a "get out of car" action which is mentioned in the script but happens outside of the view of the camera.

Temporal misalignment may result from the discrepancy between subtitles and scripts. Perfect subtitle alignment ($a = 1$), however, does not guarantee the correct action annotation in video due to possible discrepancies between scripts and movies. To investigate this issue, we have manually annotated several hundreds of actions in 12 movie scripts and compared these to the visual ground truth obtained by manual video annotation. From 147 actions with correct text alignment (a=1) only ∼70% did match with the video. The rest of the samples either were misaligned in time (∼ 10%), were outside the field of view (∼10%) or were completely missing in the video (∼10%). The misalignment of subtitles ($a < 1$) further decreases visual precision as illustrated in Figure 3.1(b). Figure 3.1(c) shows a typical example of a "visual false positive" for the action "GetOutCar" occurring outside the field of view of the camera.

3.1.2 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 real ones, 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) (b) (c)

Figure 3.2: Text-based retrieval of action samples from scripts. (a),(b): Precision-Recall for action retrieval using regular expression matching and regularized perceptron classification. (c): Statistics of action samples in the automatic training, clean training and test sets of the Hollywood human action dataset.

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 [67].

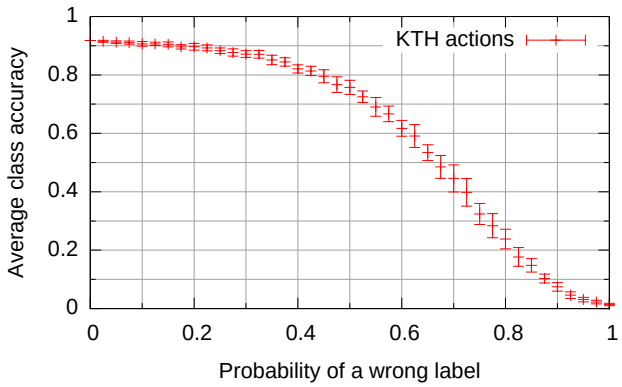
To cope with the variability of natural language descriptions of actions, we adopt a text classification approach [68]. A classifier labels each scene description in scripts as containing the target action or not. The implemented approach relies on the bag-of-features model, where each scene description 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 fewer than three training documents are removed. For the classification we use a regularized perceptron [91]. 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 eight classes of movie actions. 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 Figures 3.2(a)-(b) clearly indicate the benefit of the text classifier. The average precision-recall values over all classes are [prec. 0.95 / rec. 0.91] for the text classifier versus [prec. 0.55 / rec. 0.88] for regular expression matching.

3.1.3 Action classification

Hollywood human actions dataset. Using the video-to-script alignment and text classification described above, we have constructed Hollywood human actions dataset[3] with two training and one test subsets containing eight classes of actions (see Figure 3.2(c)). Action samples in training and test subsets come from two non-overlaping sets of 12 and 20 Hollywood movies respectively. For all subsets we first apply automatic script alignment and script-based action annotation. For samples in the *clean* training and test subsets we manually verify and correct action labels. The *automatic* training subset contains samples that have been retrieved automatically from scripts by the text classifier. We limit the automatic training subset to actions with an alignment score $a > 0.5$ and a video length of less than 1000 frames. As shown in Figure 3.2(c), the label noise of the automatic training set is ~40%.

[3]The dataset is available from http://www.irisa.fr/vista/actions.



(a)

	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%
All (mAP)	38.9%	22.9%	12.5%

(b)

Figure 3.3: (a): Accuracy of action classification for the KTH dataset using varying fractions of wrong labels at training time. (b): Average precision (AP) of action classification for the Hollywood dataset using clean and automatic training subsets.

Robustness to noise in the training data. Given the large amount of label noise in the automatic training subset, we first test the action classification method (see Section 2.1) on the simpler KTH dataset under varying amount of label noise. Figure 3.3(a) shows the recognition accuracy as a function of the probability p of a training sample label being wrong. 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 action classification degrades gracefully in the presence of labeling errors. At $p = 0.4$ the performance decreases by around 10%. A comparable level of resistance to label noise have been observed by the authors of [L14] when evaluating image classification on the PASCAL VOC'07 dataset.

Action recognition in real-world videos. We report action classification results for eight action classes in 217 test videos of the Hollywood dataset. To measure the effect of noisy labels in the automatic training set, we train binary action classifiers for the clean and automatic training subsets. The performance for both classifiers is compared to chance and is reported using per-class average precision measure in Figure 3.3(b).

The results obtained by the trained classifiers can be regarded as good given the difficulty of the problem and the relatively small number of training samples. The results obtained for the automatic training subset are lower compared to the clean training subset, however, for all classes except "HandShake" the automatic training obtains significantly higher performance than chance. This shows that script-based action annotation can be used successfully to automatically train action classifiers and, hence, to scale action recognition to many new action classes. Moreover, as will be shown in the next section, a further increase in performance can be obtained by increasing the number of training samples when using additional movies and scripts for training. Explicitly resolving the noise issue in the automatic training set is another direction for improvement that will be explored in Section 3.3.

3.2 Joint recognition of actions and scenes

Human actions are frequently constrained by the purpose and the physical properties of scenes and demonstrate high correlation with particular scene classes. For example, eating often happens in

Figure 3.4: Video samples from our dataset with high co-occurrences of actions and scenes and automatically assigned annotations.

a kitchen while running is more common outdoors (see examples in Figure 3.4). In this section we aim to exploit relations between actions and scenes. Using movie scripts as a means of weak supervision we (a) automatically discover relevant scene classes and their correlation with human actions, (b) show how to learn selected scene classes from video without manual supervision, and (c) develop a joint framework for action and scene recognition and demonstrate improved recognition of both in natural video.

3.2.1 Action and scene retrieval

Actions. To automatically collect video samples for human actions, we follow the approach of the previous section and automatically annotate twelve frequent action classes in 69 movies using the corresponding scripts. The training subsets and the test subset are obtained from two non-overlaping sets of movies. The resulting dataset (Hollywood-2[4]) extends the Hollywood dataset in Section 3.1 with more action classes and samples as illustrated in Figure 3.5(a).

Scenes. We aim to automatically (i) identify action-related scene types and (ii) estimate co-occurrence relations between actions and scenes. We use movie scripts and explore *scene captions*, i.e., short descriptions of the scene setup, which are consistently present in most movie scripts and usually provide information on the locations and day times:

INT. TRENDY RESTAURANT - NIGHT
INT. M. WALLACE'S DINING ROOM - MORNING
EXT. STREETS BY DORA'S HOUSE - DAY.

To discover relevant scene concepts, we collect unique words and consecutive word pairs from the captions. We use WordNet [23] to select expressions corresponding to instances of "physical entity". We also use the WordNet hierarchy to generalize concepts to their hyponyms, such as *taxi*→*car*, *cafe*→*restaurant*, but preserve concepts which share hyponyms, such as *kitchen*, *living room* and *bedroom*, that share the *room* hyponym. We also explicitly recognize INT (interior) and EXT (exterior) tags. From the resulting 2000 contextual concepts we select 30 that maximize co-occurrence with action classes in the training scripts. To select both frequent and informative concepts, we sort them by the entropy computed for distributions of action labels. This results in an ordered list from which we take the top ten scene concepts (see Figure 3.5(b)).

Figure 3.5(c) illustrates co-occurrences between automatically selected scene classes and actions. The size of the circles corresponds to the estimated probabilities of scenes for given action classes and coincides well with intuitive expectations. For example, *running* mostly occurs in outdoor scenes while *eating* mostly does in kitchen and restaurant scenes. Figure 3.5(c) also shows the consistency of action-scene relations estimated from text and from the visual ground truth.

[4]The dataset is available from http://www.irisa.fr/vista/actions/hollywood2.

Actions

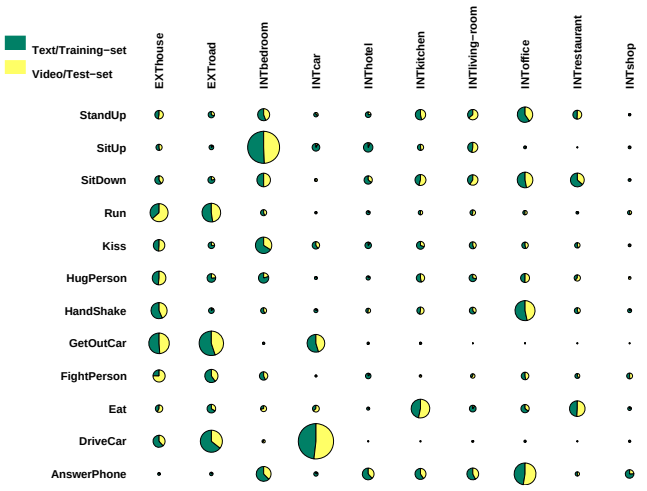
	Clean Train	Auto Train	Clean Test
AnswerPhone	66	59	64
DriveCar	85	90	102
Eat	40	44	33
FightPerson	54	33	70
GetOutCar	51	40	57
HandShake	32	38	45
HugPerson	64	27	66
Kiss	114	125	103
Run	135	187	141
SitDown	104	87	108
SitUp	24	26	37
StandUp	132	133	146
All Samples	823	810	884

(a)

Scenes

	Auto Train	Clean Test
EXT-House	81	140
EXT-Road	81	114
INT-Bedroom	67	69
INT-Car	44	68
INT-Hotel	59	37
INT-Kitchen	38	24
INT-LivingRoom	30	51
INT-Office	114	110
INT-Restaurant	44	36
INT-Shop	47	28
All Samples	570	582

(b)



(c)

Figure 3.5: (a)-(b): Distribution of video samples over action and scene classes as well as different subsets of the Hollywood-2 dataset. (c): Conditional probabilities $p(Scene|Action)$ estimated from scripts (green) and ground truth visual annotation (yellow). Note the consistency of high probability values (large circles) with intuitively expected correlations between actions and scenes. Observe also the consistency of probability values automatically estimated from text and manually estimated from visual video data.

3.2.2 Classification with context

The goal in this section is to improve action classification based on scene context. We assume we are given action and scene classifiers g_a, g_s obtained from automatically generated training samples as described in the previous section. For a given test sample x, we integrate scene context by updating the action classification score $g_a(x)$ for an action $a \in \mathcal{A}$. To the original classification score we add a linear combination of the scores $g_s(x)$ for contextual concepts (scenes) $s \in \mathcal{S}$:

$$g'_a(x) = g_a(x) + \tau \sum_{s \in \mathcal{S}} w_{as} g_s(x) \tag{3.1}$$

where τ is a global context weight and w_{as} are the weights linking concepts a and s. The parameter τ allows to control the influence of the context. We have observed that the results are not very sensitive to this parameter and set $\tau = 2$ for all our experiments.

Mining contextual links from scripts. One way to obtain the weights w_{as} is by using action-scene correlations estimated from scripts. As shown in Figure 3.5(c), such correlations correspond well to our intuition and to the actual action-scene correlations in the video. Let us rewrite the context model in (3.1) as

$$\bar{g}'_a(x) = \bar{g}_a(x) + \tau W \bar{g}_s(x) \tag{3.2}$$

where $\bar{g}'_a(x)$ is the vector of new scores and $\bar{g}_a(x)$ is a vector with the original scores for all basic (action) classes, $\bar{g}_s(x)$ is a vector with scores for all context (scene) classes, and W is a weight matrix. We determine W from scripts of the training set by defining it in terms of conditional probabilities $W_{ij} = p(a_i|s_j)$. $W\bar{g}_s(x)$ can be interpreted as a vector proportional to $(p(a_1), ..., p(a_n))^\top$ if we consider the classification scores $\bar{g}_s(x)$ as likelihoods of scene labels.

	SIFT	HOG HOF	SIFT HOG HOF
Action average	*0.200*	*0.324*	***0.326***
Scene average	*0.319*	*0.296*	***0.351***
Total average	*0.259*	*0.310*	***0.339***

(a)

Actions

Context	*mAP*
text context	***0.352***
no context	*0.326*
context only	*0.212*
chance	*0.125*

(b)

Table 3.1: (a): Average precision for the classification of actions and scenes in the Hollywood-2 dataset using automatic subsets for training and different sets of features. Both actions and scenes classes benefit from combining static and dynamic features. (b): mean average precision for action classification with and without scene context.

3.2.3 Experimental results

We represent actions and scenes by histograms of quantized local feature descriptors computed over videos. In particular, we use local STIP feature detectors in combination with HOG and HOF descriptors to describe dynamic video content as described Chapter 2. We also use SIFT descriptors extracted on every 25th frame of a video to represent scene-specific static information. We use SVM with χ^2 kernel to obtain separate action and scene classifiers g_a, g_s trained for different subsets of features. Table 3.1(a) presents results for independent action and scene classification using dynamic and static features as well as their combination.

The results of action classification using scene context are presented in Table 3.1(b). As can be seen, scene context improves action classification by 2.5% on average. The largest gain of 10% was observed for action classes SitUp and DriveCar, which have strong correlation with scene classes Bedroom and Car respectively (see Figure 3.5(c)). Interestingly, using context *only*, i.e., classifying actions with only contextual scene classifiers, is still significantly better than chance. This clearly demonstrates the importance of context for action recognition in natural scenes.

3.3 Weakly-supervised temporal action localization

This section addresses the problem of automatic temporal localization of human actions in video. We consider two associated problems: (a) weakly-supervised learning of action models from readily available annotations, and (b) temporal localization of human actions in test videos. As in the previous sections of this chapter, we use movie scripts as a means of weak supervision. Scripts, however, provide only implicit, noisy, and imprecise information about the type and location of actions in video. We address this problem with a kernel-based discriminative clustering algorithm that locates actions in the weakly-labeled training data. Using the obtained action samples, we train temporal action detectors and apply them to locate actions in the raw video data.

3.3.1 Automatic supervision for action recognition

Script-based retrieval of action samples. We follow Section 3.1 and automatically collect training samples with human actions using video scripts. While Section 3.1 uses supervised text classification to localize human actions in scripts, here we avoid manual text annotation altogether. We use the OpenNLP toolbox [59] for natural language processing and apply part of speech (POS) tagging to identify instances of nouns, verbs and particles. We also use named entity recognition

(NER) to identify people's names. Given results of POS and NER we search for patterns corresponding to particular classes of human actions such as (*/PERSON .* opens/VERB .* door/NOUN). This procedure automatically locates instances of human actions in text such as "... **Jane** jumps up and **opens** the **door** ...", "... **Carolyn opens** the front **door** ...", "... **Jane opens** her bedroom **door** ...".

Temporal localization of actions in video. Automatic script alignment described in Section 3.1.1 only provides coarse temporal localization of human actions, especially for episodes with rare dialogues. In addition, incorrect ordering of actions and speech in scripts and errors of script-to-subtitle alignment often result in unreliable temporal offsets. To overcome temporal misalignment, we increase the estimated time boundaries of scene descriptions and obtain corresponding videos denoted here as *video clips*. Table 3.2 illustrates the accuracy of automatic script-based annotation in video clips with increasing temporal extents.

Training an accurate action classifier requires video clips with both accurate labels and precise temporal boundaries. The high labelling accuracy in Table 3.2, however, is bound to the imprecise temporal localization of action samples. This trade-off between accuracies in labels and temporal localization of action samples comes from the aforementioned problems with imprecise script alignment. In this section we target this problem and address visual learning of human actions in a *weakly supervised setting* given imprecise temporal localization of training samples. Next, we present a weakly supervised clustering algorithm to automatically localize actions in training samples.

3.3.2 Human action clustering in video

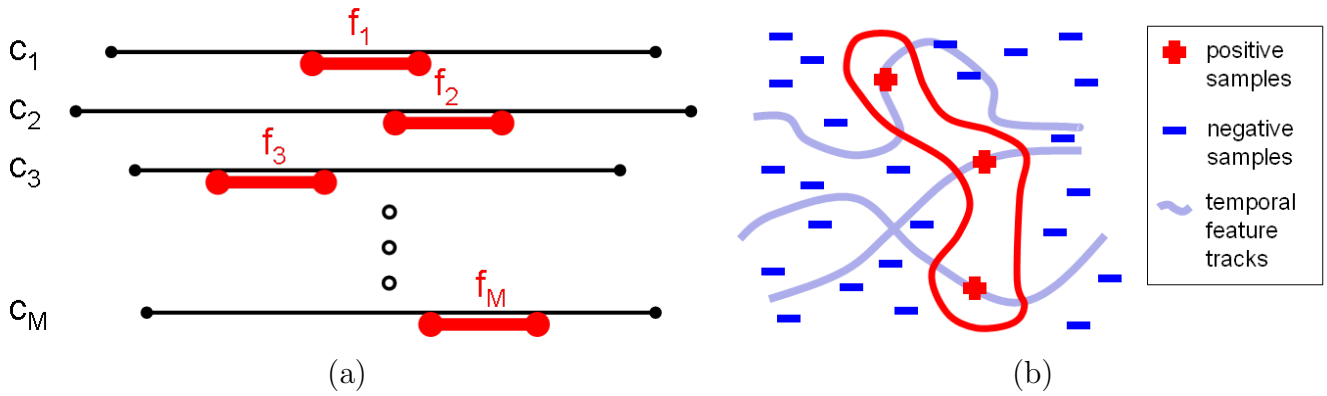
To train accurate action models we aim at localizing human actions inside video clips provided by automatic video annotation. We assume most of the clips contain at least one instance of a target action and exploit this redundancy by clustering clip segments with consistent motion and shape. We next formalize the clustering problem and describe a discriminative clustering procedure. Section 3.3.3 evaluates the effect of the method for weakly-supervised action localization.

Discriminative clustering for video clips. Our goal is to jointly segment video clips containing a particular action—that is, we aim at separating what is common within the video clips (i.e., the particular action) from what is different among these (i.e, the background frames). Our setting is however simpler than general co-segmentation in the image domain since we only perform *temporal* segmentation. That is, we look for segments that are composed of contiguous frames.

For simplicity, we further reduce the problem to separating one segment per video clip (the action segment) from a set of *background video segments*, taken from the same movie or other movies, and which are unlikely to contain the specific action. We thus have the following learning problem: We are given M video clips $c_1, \ldots, c_M$ containing the action of interest but at unknown

Clip length (frames)	100	200	400	800	1600
Label accuracy	19%	43%	60%	75%	83%
Localization accuracy	74%	37%	18%	9%	5%

Table 3.2: Accuracy of automatic script-based action annotation in video clips. Label accuracy indicates the proportion of clips containing labeled actions. Localization accuracy indicates the proportion of frames corresponding to labeled actions. The evaluation is based on the annotation of three actions classes: StandUp, SitDown and OpenDoor in fifteen movies selected based on their availability.



(a) (b)

Figure 3.6: (a): Illustration of the temporal action clustering problem. Given a set of M video clips $c_1, \ldots, c_M$ containing the action of interest at unknown position, the goal is to temporally localize a video segment in each clip containing the action. (b): In feature space, positive samples are constrained to be located on temporal feature tracks corresponding to consequent temporal windows in video clips. Background (non-action) samples provide further constrains on the clustering.

position within the clip as illustrated in Figure 3.6(a). Each clip c_i is represented by n_i temporally overlapping segments centered at frames $1, \ldots, n_i$ represented by histograms $h_i[1], \ldots, h_i[n_i]$ in $\mathbb{R}^N$. Each histogram captures the ℓ_1-normalized frequency counts of quantized space-time interest points, as described in Chapter 2, i.e. it is a positive vector in $\mathbb{R}^N$ whose components sum to 1. We are also given P background video segments represented by histograms $h_1^b, \ldots, h_P^b \in \mathbb{R}^N$. Our goal is to find in each of the M clips i one specific video segment centered at frame $f_i \in \{1, \ldots, n_i\}$ so that the set of M histograms $h_i[f_i]$, $i = 1, \ldots, M$ form one cluster while the P background histograms form another cluster as illustrated in figure 3.6(b).

Problem formulation. We formulate the above clustering problem as the minimization of a discriminative cost function [84]. First, let us assume that correct segment locations f_i, $i \in \{1, \ldots, M\}$, are known (i.e., we have identified the locations of the actions in video). We can now consider a support vector machine (SVM) [69] classifier aiming at separating the identified action video segments from the given background video segments, which leads to the following cost function

$$J(f, w, b) = C_+ \sum_{i=1}^{M} \max\{0, 1 - w^\top \Phi(h_i[f_i]) - b\} + C_- \sum_{i=1}^{P} \max\{0, 1 + w^\top \Phi(h_i^b) + b\} + \|w\|^2, \tag{3.3}$$

where $w \in \mathcal{F}$ and $b \in \mathbb{R}$ are parameters of the classifier and Φ is the implicit feature map from $\mathbb{R}^N$ to feature space $\mathcal{F}$, corresponding to the intersection kernel between histograms, defined as [38]

$$k(x, x') = \sum_{j=1}^{N} \min(x_j, x'_j). \tag{3.4}$$

Note that the first two terms in cost function (3.3) represent the hinge loss on positive and negative training data weighted by factors C_+ and C_- respectively, and the last term is the regularizer of the classifier. Note that training the SVM with locations f_i known and fixed corresponds to minimizing $J(f, w, b)$ with respect to classifier parameters w, b.

However, in the clustering setup considered in this work, where the locations f_i of action video segments within clips are unknown, the goal is to minimize the cost function (3.3) with respect to both the locations f_i and the classifier parameters w, b, so as to separate positive action segments from (fixed) negative background video segments. Denoting by $H(f) = \min_{w \in \mathcal{F}, b \in \mathbb{R}} J(f, w, b)$ the associated optimal values of $J(f, w, b)$, the cost function $H(f)$ now characterizes the separability of a particular selection of action video segments f from the (fixed) background videos. Following [41, 84], we can now optimize $H(f)$ with respect to the assignment f.

We use a coordinate descent algorithm, where we iteratively optimize $H(f)$ with respect to position f_i of the action segment in each clip, while leaving all other components (positions of other positive video segments) fixed. In our implementation, which uses the LibSVM [8] software, in order to save computing time, we re-train the SVM (updating w and b) only once after an optimal f_i is found in each clip. The Initial histogram h_i^0 for each video clip c_i is set to the average of all segment histograms $h_i[f_i]$ within the clip.

The evaluation of the above clustering procedure provides satisfactory results and details of this evaluation can be found in [L4].

3.3.3 Action localization experiments

We test our full framework for automatic learning of action detectors including (i) automatic retrieval of training action clips by means of script mining (Section 3.3.1), (ii) temporal localization of actions inside clips using discriminative clustering (Section 3.3.2) and (iii) supervised temporal detection of actions in test videos using a temporal sliding-window classifier. To train an action classifier we use fifteen movies aligned with scripts, and choose two test action classes OpenDoor and SitDown based on their high frequency in our data. The only manual supervision provided to the system consists of text patterns for the actions defined as (*/PERSON .* opens/VERB .* door/NOUN) and (*/PERSON .* sits/VERB .* down/PARTICLE). Matching these text patterns with scripts results in 31 and 44 clips with OpenDoor and SitDown actions respectively. We use these clips as input to the discriminative clustering algorithm and obtain segments with temporally localized action boundaries. The segments are passed as positive training samples to train an SVM action classifier. We compare the performance of our method with two action classifiers trained using positive training samples corresponding to (a) entire clips and (b) ground truth action intervals.

To test detection performance we manually annotated all 93 OpenDoor and 86 SitDown actions in three movies: Living in oblivion, The crying game and The graduate. Detection results for the three different methods and two action classes are illustrated in terms of precision-recall curves in Figure 3.7. The comparison of detectors trained on clips and on action segments provided by the clustering clearly indicates the improvement achieved by the discriminative clustering algorithm for both actions. Moreover, the performance of automatically trained action detectors is comparable to the detectors trained on the ground truth data. We emphasize the large amount (450.000 frames in total) and high complexity of our test data illustrated with a few detected action samples in Figure 3.8.

3.4 Summary

This chapter has addressed weakly-supervised learning of human actions in video. Our methods use readily-available resources such as movie scripts to automatically annotate actions at the training

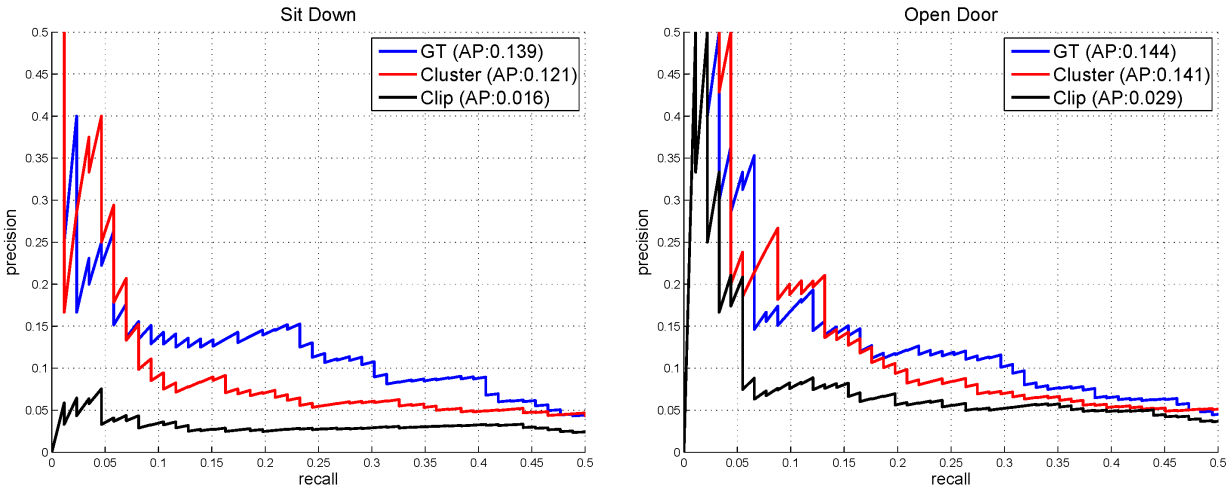


Figure 3.7: Precision-recall curves corresponding to detection results for two action classes in three movies. The three compared methods correspond to detectors trained on ground truth intervals (GT), clustering output (Cluster) and clips obtained from script mining (Clip). Note that the axes are scaled between 0 and 0.5 for better clarity.

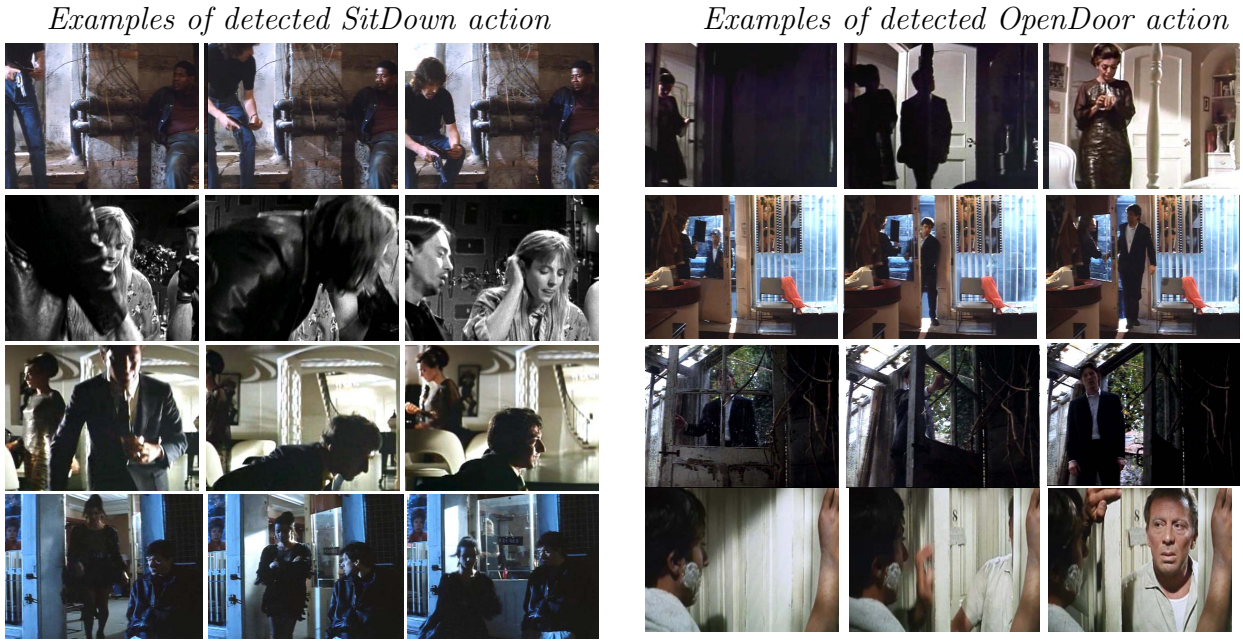


Figure 3.8: Examples of action samples detected with the automatically trained action detector in three test movies.

time and, hence, enable scalability of action recognition to many classes. While we have so far explored the automatic annotation of action and scene types, video scripts contain much of the additional and detailed information about objects, person-object interactions, person identities, etc. Using such information will likely support richer modeling of dynamic scenes in the future. Such modeling, however, will also require more structured representations of both video and text compared to the bag-of-features methods used here. To overcome uncertainty of script-to-video alignment and to handle the large variability of corresponding elements in video and text, new weakly-supervised learning methods are required.

Chapter 4

Object and action localization

The two previous chapters have addressed the problem of classification and the temporal localization of events in the video without explicitly reasoning about their *spatial location*. While classification tasks enable the use of conceptually simple and efficient bag-of-features representations, modeling spatial location of objects and events is important to reason about the structure of dynamic scenes, e.g., to identify the people and objects involved in actions and interactions.

This chapter addresses the spatial localization of objects in images and spatio-temporal localization of actions in the video. In contrast to methods described in the two previous chapters, here we explicitly model the structure of objects and dynamic events in terms of *parts*. We first present a method for object detection and then generalize it to the detection of simple actions (e.g., "Drinking") by considering actions as space-time objects in the video. This chapter is based on the work published in [L9, L10, L15].

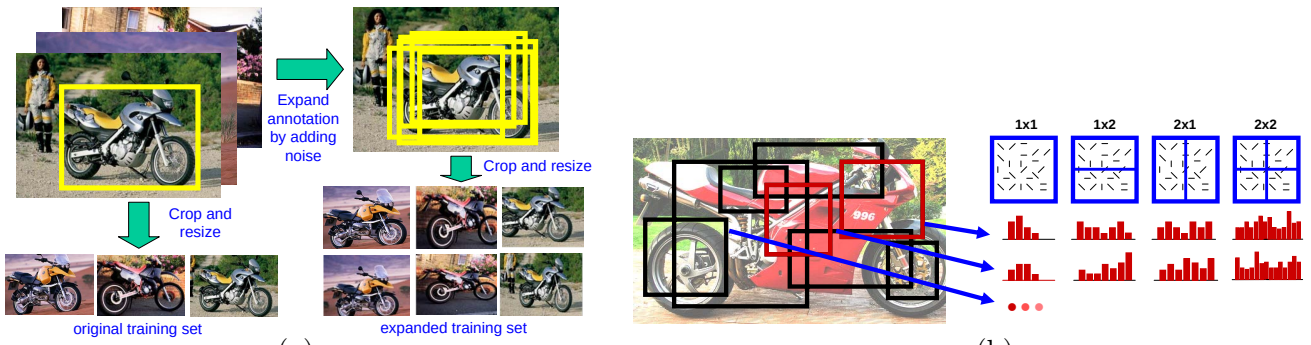
4.1 Object detection

We address the problem of category-level object localization in still images. We assume a number of object images are provided for training along with the corresponding bounding box annotations. To account for slight misalignments of objects within bounding boxes and to increase the size of the training set, we "jitter" annotations by adding a small amount of noise to the corners of the original annotations and generate multiple training samples I for each original object image as illustrated in Figure 4.1(a).

We represent each cropped object sample by the collection of its subregions, or parts. To avoid a heuristic selection of parts, we consider an exhaustive set of rectangular sub-windows r (see Figure 4.1(b)(left) and select a small subset of parts at training time. Following the work on SIFT and HOG image descriptors [13, 51], we represent each part by the l_1-normalized histograms of gradient orientations (HOG). We construct HOG features by discretizing image gradient orientations into $m = 4$ bins. To preserve rough location of image measurements within a part, we sub-divide parts into cells using four alternative grids k and compute four features $f_{k,r}(I), k = 1...4$ for each part r as illustrated in Figure 4.1(b)(right).

4.1.1 AdaBoost learning

AdaBoost [26] is a popular machine learning method combining properties of an efficient classifier and feature selection. The discrete version of AdaBoost defines a strong binary classifier C using



(a) (b)

Figure 4.1: (a): (Left): Positive training samples are obtained using crop-and-resize procedure applied to training images with rectangular objects annotations. (Right): The same procedure is applied to training images using the large number o f automatically generated noisy annotations. Note how annotation noise adds simulated affine deformations to the novel training samples. (b): Histogram features. (Left): Sample regions from an exhaustive set of regions defined by different spatial extents and positions within the object window. (Right): histogram features computed for each region according to four types of spatial grids.

a weighted combination of T weak learners h_t with weights α_t as

$$C(I) = \text{sgn}(\sum_{t=1}^{T} \alpha_t h_t(I)), \quad h(I) = \begin{cases} 1 & \text{if } \; g(f(I)) > threshold \\ -1 & \text{otherwise} \end{cases}. \tag{4.1}$$

At each new training round t, AdaBoost selects a weak learner h_t associated with a feature f_t that best classifies training samples with high classification error in the previous rounds. In this work we define weak learners for features $f_{k,r}(I)$ in terms of Fisher Linear Discriminant (FLD) [17]

$$g = w^\top f \quad \text{with} \quad w = (S^{(1)} + S^{(2)})^{-1}(\mu^{(1)} - \mu^{(2)}), \tag{4.2}$$

where $\mu^{(1|2)}$ and $S^{(1|2)}$ are means and covariance matrices of $f_{k,r}(I)$ for positive$^{(1)}$ and negative$^{(2)}$ training examples. As required by AdaBoost, h minimizes the weighted error using the weighted means and covariance matrices defined for sample weights d_i as :

$$\mu = \frac{1}{n \sum d_i} \sum_{i}^{n} d_i f(I_i), \quad S = \frac{1}{(n-1) \sum d_i^2} \sum_{i}^{n} d_i^2 (f(I_i) - \mu)(f(I_i) - \mu)^\top. \tag{4.3}$$

The optimal threshold for h in (4.1) is obtained by the exhaustive search as in [79].

4.1.2 Evaluation

We evaluate the our method on the PASCAL VOC 2005 and VOC 2006 datasets [19, 20] on the task of detecting object classes “motorbike”, “bicycle”, people, “car”, “horse” and “cow”. For the detection we use the standard window scanning technique and apply the classifier to a large number of image sub-windows with densely sampled positions and sizes. We apply non-maximum suppression by clustering detections with similar positions and scales, the number of detections within a cluster is used as a detection confidence. Results are reported by precision-recall curves and average precision values.

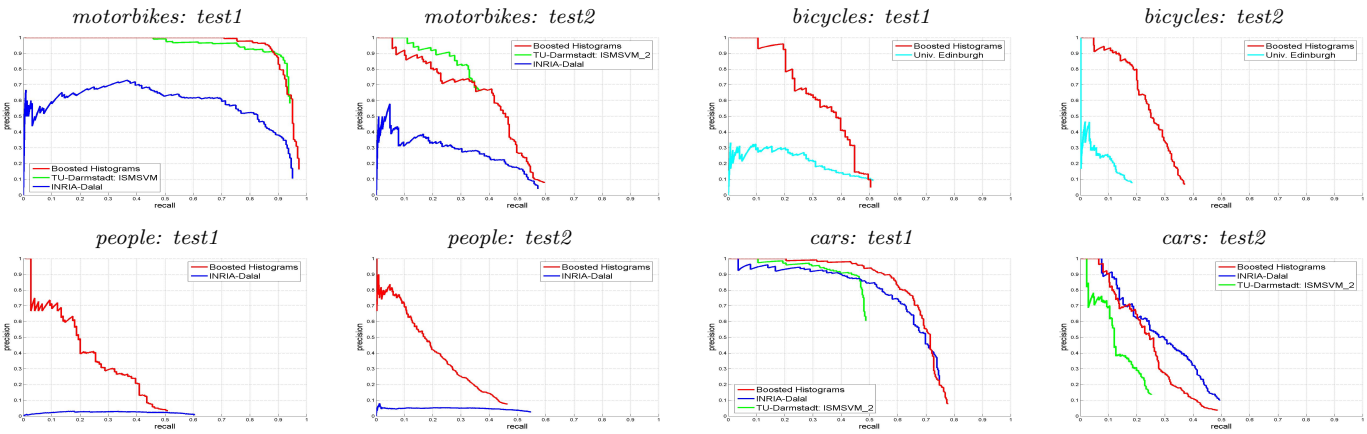


Figure 4.2: PR-curves for eight object detection tasks in PASCAL VOC 2005 Challenge. The proposed method (Boosted Histograms) is compared to the best performing methods reported in Pascal VOC 2005 Challenge.

Method	bicycle	cow	horse	motorbike	person
INRIA_Douze	0.414	0.212	–	**0.390**	**0.164**
INRIA_Laptev	**0.440**	0.224	**0.140**	0.318	0.114
TKK	0.303	**0.252**	0.137	0.265	0.039

Table 4.1: Average precision for the three best methods participating in the object detection `task 3` of the PASCAL VOC 2006 `http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2006/prelimres/index.html`.

In Figure 4.2 our method (*boosted histograms*) demonstrates best results in seven out of eight detection tasks of VOC 2005. Boosted histograms significantly outperforms results reported in [20] for people and bicycles. For motorbikes and cars our method has comparable performance to the best results reported by *INRIA-Dalal* [13] and *TU-Darmstadt* [27]. Our method has entered the object detection competition of PASCAL VOC 2006 for five object classes: *bicycle*, *cow*, *horse*, *motorbike* and *person*. As shown in Table 4.1, the method (INRIA_Laptev) has obtained best results for the classes *bicycle* and *horse* and second-best results the three other classes. Object detection has been improved since VOC 2006 by more recent methods, e.g., [24].

We examine the parameters of our method in Figure 4.3. The method benefits from the increased number of positive training samples generated by "jittering" original training images as illustrated in Figure 4.3(left). HOG image features are compared to alternative descriptors for object parts in terms of histograms of quantized responses of (a) second order derivatives (2Jet), (b)multi-scale Laplacian filters (MS-Lap) and (c) color values (Color). Figure 4.3(right) illustrates the advantage of HOG compared to other single descriptors. Enabling the classifier to choose among all four descriptors (Comb) results in the best performance.

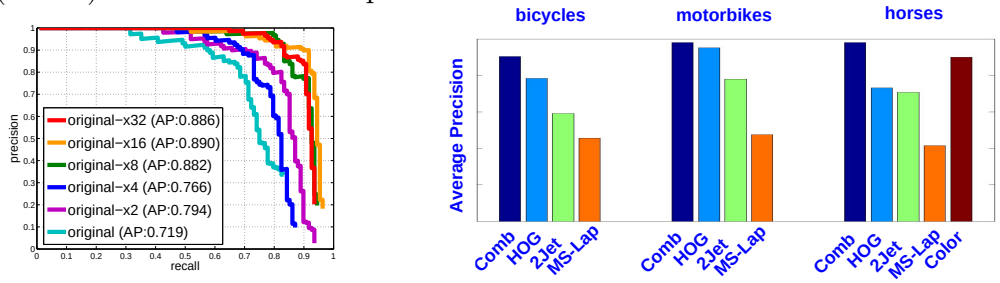


Figure 4.3: Evaluation of parameters. (Left): Results of motorbike detection in VOC 2005 validation set with varying number of "jittered" training samples. (Right): Evaluation of alternative local descriptors for the detection of three object classes in VOC 2006 dataset.

4.2 Action detection in video

We next address the problem of recognizing and localizing human actions in realistic videos. We consider "atomic" actions characterized by consistent structure in space-time. Figure 4.4(top) illustrates examples of such actions with similar space-time patterns in the video. We build upon the intuition that atomic actions in video can be treated similarly to objects in images, see [78] for a related discussion in psychology. We also build upon recent advances in object recognition and extend object detection in still images to action detection in video. We adapt the boosted histogram object detector described in the previous section and model actions by constellations of space-time parts as illustrated in Figure 4.5(top). Our features f encode the appearance and motion of each action part as described next. We define an action classifier in terms of action features f by replacing static image features of the AdaBoost object classifier in (4.1).

4.2.1 Action modeling and classification

We wish to exploit the appearance and motion of actions for action recognition. Following the previous section, we use 4-bin histograms of gradients orientations (Grad4), to represent the shape of action parts. To represent motion, we use 5-bin histograms of optical flow [14] (OF5) with four bins corresponding to quantized motion directions and the last bin corresponding to no motion. The Grad4 and OF5 histograms are accumulated in space-time blocks of the normalized action cuboids as illustrated in Figure 4.5(bottom). We assume a rough alignment of action samples and their features in space-time. Each feature f_θ, $\theta = (x, y, t, \delta x, \delta y, \delta t, \beta, \psi)$ is defined by the

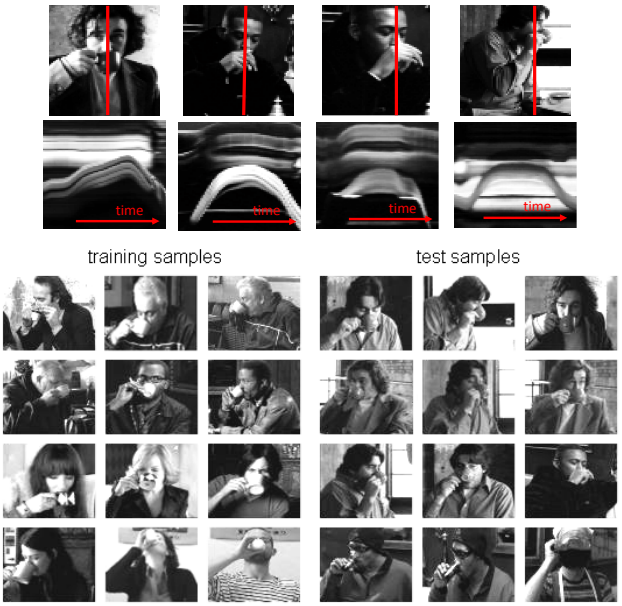


Figure 4.4: Action examples from the movie "Coffee and Cigarettes". (Top): Examples of drinking actions and the corresponding space-time slices. Observe similar structure of actions in time despite variations in viewpoints and actors. (Bottom): Example frames of annotated drinking actions in the training and test subsets.

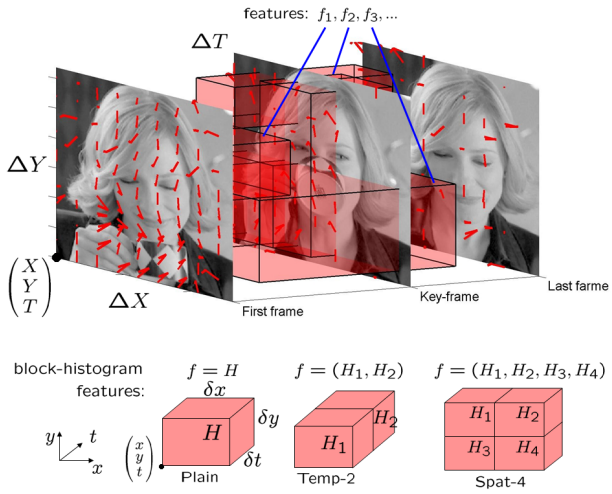


Figure 4.5: (Top): action volume in space-time represented by three frames of a drinking action. Arrows on the frames correspond to the computed optic flow vectors. Transparent blocks in red demonstrate some of the space-time features of a boosted space-time classifier. (Bottom): Three types of features with different arrangement of histogram blocks. Histograms for composed blocks (Temp-2, Spat-4) are concatenated into a single feature vector.

space-time location (x, y, t) and the space-time extents $(\delta x, \delta y, \delta t)$ of the histogram block, by the type of block β in $\{\text{Plain}, \text{Temp-2}, \text{Spat-4}\}$ (Figure 4.5(bottom)) and by the type of histogram ψ in $\{\text{OF5}, \text{Grad4}\}$.

Space-time classifier. To investigate the influence of shape information on action recognition, we learn two AdaBoost classifiers (4.1), one with optic flow features only (*OF5* classifier) and another one with shape and motion features (*OFGrad9* classifier). OF5 classifier is closely related to the method by Ke et al. [44]. To efficiently compute features, we use integral video histograms and space-time video pyramids. When training OF5 and OFGrad9 classifiers we randomly subsample the initially very large set of space-time features.

Keyframe classifier. We also investigate the possibility of recognizing actions from a single frame. For this purpose we adopt the boosted histogram object classifier from Section 4.1 and train it using action *keyframes* as positive samples while using non-action video frames for the pool of negative examples (see Figure 4.5 and Section 4.2.2 for the definition of action keyframes).

STIP-NN classifier. We also test the performance of a bag-of-features classifier using space-time interest points (see Chapter 2). We accumulate histograms of STIP features from space-time volumes of action samples and use Nearest Neighbour (NN) for classification. In contrast to OF5 and OFGrad9 classifiers above, the resulting STIP-NN classifier does not encode the global space-time structure of actions.

4.2.2 Dataset and annotation

To train and test action detection in realistic dynamic scenes, we use the movies "Coffee and Cigarettes" (2003) and "Sea of Love" (1989), providing an excellent pool of natural samples for action classes "drinking" (127 samples) and "smoking" (149 samples). These actions appear in different scenes, and they are performed by different people while being recorded from different view points. We split action samples into training and test subsets that share no common actors nor scenes (see [L15] for details). Figure 4.4 illustrates the large within-class variability of drinking samples as well as the difference between the training and the test subsets. A few examples of scenes in Figure 4.7 illustrate the large variability of scales, locations and view-points as well as the typical "background clutter" with surrounding people acting in various ways.

We use manual annotations of drinking actions both for the alignment of training samples and for the evaluation of detection performance on the test set. Each drinking action is associated with a manually-annotated space-time cuboid $R = (p, \Delta p)$ with location $p = (X, Y, T)^\top$ and a spatio-temporal extent $\Delta p = (\Delta X, \Delta Y, \Delta T)^\top$ as illustrated in Figure 4.5(top). We also annotate a *keyframe* for each drinking action by the time when the hand reaches the mouth.

4.2.3 Classification results

We study the relative performance of the action classification methods introduced in Section 4.2.1 on the tasks of discriminating drinking actions from (i) smoking actions and (ii) random motion patterns in the video. In these preliminary classification experiments, all test samples are cropped and resized to the common rectangular cuboid shapes in space-time. All classifiers are trained using the same set of positive training samples. The negative training samples for boosted classifiers correspond to "hard negatives" obtained from training video episodes. For the STIP-NN classifier, negative training samples correspond to smoking actions.

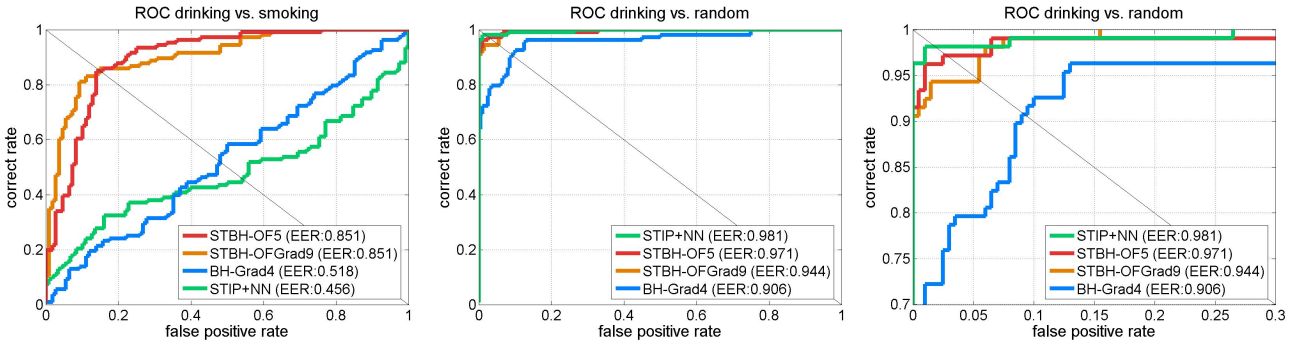


Figure 4.6: Classification of drinking actions vs. (a): smoking actions and (b): random motion patterns. (c) is a magnified part of (b). ROC curves are obtained by thresholding on the confidence values of test samples. For boosted classifiers the confidence is defined by the number of passed cascade stages. For STIP-NN classifier the confidence is the normalized distance to the closest negative training sample.

Classification results for the first experiment are illustrated in Figure 4.6(a) and show the best performance for the boosted OF5 and OFGrad9 space-time classifiers. The STIP-NN and the keyframe classifiers have close-to-chance performance in this experiment. The classification performance in the second and simpler experiment is improved for all methods as illustrated in Figures 4.6(b)-(c). We note that the extension of the OF5 classifier by the shape features in the OFGrad9 classifier does not improve classification performance in both experiments. On the other hand, the relatively high performance of all methods in the second experiment suggests that their combination could be used for further improvements.

4.2.4 Keyframe priming

We wish our action detector to exploit the shape and motion of people. In this work we combine shape and motion information using *keyframe priming* as follows. The keyframe detector is first trained on static keyframes of drinking actions and is then applied to all frames of the training videos in a low accuracy and high recall mode to ensure detection of all true positive samples. The output of the keyframe detector is used to bootstrap (or prime) the space-time action classifier by generating space-time volumes of action hypotheses around each detected keyframe. Keyframe priming is used to collect action hypotheses for the space-time classifier both at the training and test times as illustrated in Figure 4.7.

Since the evaluation of the keyframe classifier is less computationally expensive than that of the space-time classifier, keyframe priming improves the speed of action detection at test time. The fast rejection of many false action hypotheses with keyframe priming at training time also results in an improved training error of roughly 10^{-7} in our experiments. Obtaining a similar training error with a space-time classifier alone would be difficult due to this high computational cost.

4.2.5 Detection results

We test action detection on two short vignettes from the movie "Coffee and Cigarettes" with 36,000 frames in total. We evaluate the performance of action detection using space-time classifiers with and without keyframe priming. For OF5 and OFGrad9 classifiers without keyframe priming we

Figure 4.7: Keyframe priming. (Left): Examples of the training scenes from the movie "Sea of Love". False positive detections of the keyframe classifier (red) are used as negative examples for training space-time action classifiers. Keyframe detections are not shown for positive training samples (cyan). (Right): Examples of test scenes from the movie "Coffee and Cigarettes". The keyframe-primed action classifier correctly classifies drinking hypotheses (cyan) among all detected keyframes (red). Results are shown before non-maximum suppression step.

generate and evaluate a very large set of space-time windows with different positions and space-time extents. For action classifiers using keyframe priming, we bootstrap the detection by applying the keyframe classifier independently on every frame and then using space-time classifiers to evaluate action hypotheses of different temporal extents and centered on the detected keyframes. For each method, we cluster multiple action detections with similar positions and sizes in space-time and use the size of the cluster as detection confidence.

The quantitative performance for the four tested detection methods is illustrated in Figure 4.8(left) in terms of precision-recall curves and average precision values. The OFGrad9 detector using shape and motion information outperforms the OF5 detector using motion features only. The largest improvement is obtained by the keyframe priming. This gain could be explained by the better training error of keyframe-primed action classifiers as discussed in the previous section. This improvement indicates the importance of shape information for action classification and detection.

Qualitative results are illustrated in Figure 4.8(right) by the twelve strongest detections of drinking actions in the test set obtained with the keyframe primed OF5 classifier. Most of the detections correspond to correct actions despite substantial variations in the appearance and actions of people, scene clutter and view points in the video. A video with detection results and the dataset used in this work is available on-line[1].

[1] http://www.di.ens.fr/~laptev/actiondetection.html

Figure 4.8: Results of drinking action detection. (Left): Precision-recall curves illustrating the performance of drinking action detection achieved by the four tested methods. (Right): Detections of drinking actions (yellow: true positives, red: false positives) sorted in the decreasing confidence order and obtained with the OF5 Keyframe primed event detector.

4.3 Summary

This chapter has presented methods for spatial object detection and spatio-temporal action localization. With the example of drinking actions, we have shown that simple actions can be modeled as space-time objects and detected using a generalization of object detection methods to video. We have investigated combinations of shape and motion information for action modeling and have shown particular benefit of combining static keyframe classifiers with space-time action classifiers.

This work was among the first to address action recognition and localization in realistic settings of movies. While the obtained results are encouraging, the method leaves much room for improvement. Our models of actions currently use low level features and may benefit from explicit modeling of human body parts, body kinematics and person-object interactions. Some of these directions are explored in the next chapter. While actions in this work are modeled in isolation, the relations of actions with surrounding objects and scenes as well as the temporal relations between actions performed in a sequence could be used as constraints for action recognition. Another direction for future research concerns scalability. While the locations and temporal extents of actions in this work have been annotated manually, scaling to many action classes will most likely require new weakly-supervised learning methods overcoming the need of the costly manual annotation.

Chapter 5

Modeling interactions

Human activities often involve object manipulations (opening a door, sitting down on a chair). Inversely, objects and scenes can often be described by associated actions (phone: pick up, put down, talk to; fridge: open, close, put objects inside,...). As mentioned in the Introduction, modeling and recognizing person-object interactions could be useful, for example, to detect unusual events (see Figure 1.4), to predict likely human actions or to search objects by their function. Motivated by these problems, this chapter studies person-object interactions with the aim of learning and recognizing objects by their functional properties, i.e., what people do with an object and how they do it.

Object function can be derived from known associations between object categories and human actions (the *mediated perception of function* approach [61]), for example *chair→sittable, window→openable*. Actions such as sitting, however, can be realized in many different ways which can be characteristic for some objects but not for others, as illustrated in Figure 5.1. Moreover, some objects may not support the common function associated with their category: for example, windows in airplanes are usually not openable. These and many other examples suggest that the category-level association between objects and their functions is not likely to scale well to the very rich variety of person-object interactions. Instead, we argue that functional descriptions of objects should be learned directly from observations of visual data.

In this work, we design object descriptions by learning associations between objects and spatially co-occurring human poses. To capture the rich variety of person-object interactions, we automatically detect people and estimate body poses in long-term observations of realistic indoor scenes using the state-of-the-art method of [85]. While reliable pose estimation is still a challenging problem, we circumvent the noise involved in this process by observing *many* person interactions

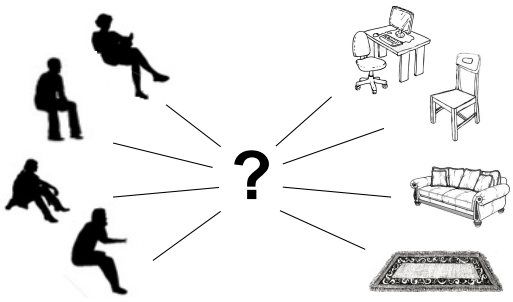


Figure 5.1: *Different ways of using objects.* While all people depicted on the left are sitting, their sitting poses can be rather unambiguously associated with the objects on the right. In this paper we build on this observation and learn object descriptions in terms of characteristic body poses.

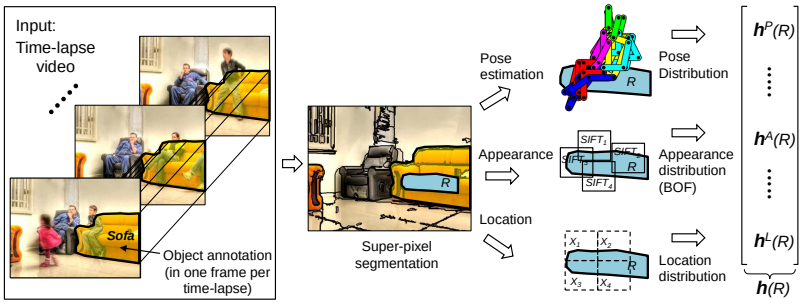


Figure 5.2: *Overview of the proposed person-based object description.* Input scenes are over-segmented into super-pixels; each super-pixel (denoted R here) is described by the distribution of co-occurring human poses over time as well as by the appearance and location of the super-pixel in the image.

with the *same instances* of objects. For this purpose we use videos from long events (parties, house cleaning), recorded with a static camera and summarized into time-lapses[1]. Static objects in time-lapses (e.g., sofas) can be readily associated with hundreds of co-occurring human poses spanning the typical interactions of people with these objects (see Figures 5.2-5.4). Equipped with this data, we construct statistical object descriptors which combine the signatures of object-specific body poses with the object's appearance. The model is learned discriminatively from many time-lapse videos of a variety of scenes. The work in this chapter was done in collaboration with students Vincent Delaitre and David Fouhey, and was published in [L2].

5.1 Method overview

Our functional object description is limited to larger objects (sofas, tables, chairs) yet, it generalizes across realistic and challenging scenes and provides significant improvements in object recognition. To simplify the learning task, we assume that input videos contain static objects with fixed locations in each frame of the video. Annotating such objects in the whole video can be simply done by outlining object boundaries in one video frame as illustrated in Figure 5.2. Moreover, person interactions with static objects can be automatically recorded by detecting people in the spatial proximity of annotated objects.

We start by over-segmenting input scenes into super-pixels, which will form the candidate object regions. For each object region R we construct a descriptor vector $\vec{h}(R)$ to be used for subsequent learning and recognition. A particular novelty of our method is a new descriptor representing an object region by the temporal statistics $\vec{h}^P(R)$ of co-occurring people. This descriptor contains a distribution of human body poses and their relative location with respect to the object region. We also represent each object region by appearance features, denoted $\vec{h}^A(R)$, and the absolute location in the frame, denoted $\vec{h}^L(R)$.

Given these descriptor vectors, one for each object region, containing statistics of characteristic poses, appearance and image locations, a linear support vector machine (SVM) classifier is learnt for each object class from the labelled training data. At test time, the same functional and appearance representation is extracted from candidate object regions of the testing video. Individual candidate object regions are then classified as belonging to one of the semantic object classes.

[1]Time-lapse http://en.wikipedia.org/wiki/Time-lapse_photography is a common media type used to summarize recordings of long events into short video clips by temporal sub-sampling. We use time-lapses widely available on public video sharing web-sites such as YouTube, which are typically sampled at one frame per 1-60 seconds.

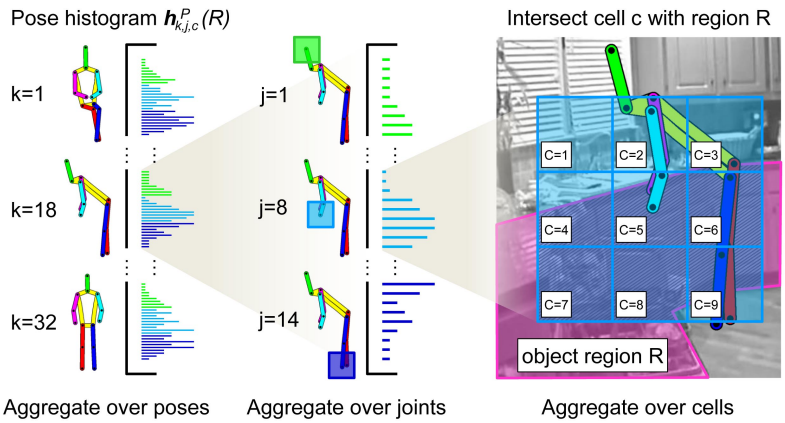


Figure 5.3: *Capturing person-object interactions.* An object region R is described by a distribution (histogram) over poses k (left), joints j (middle) and cells c (right). The 3×3 grid of cells c is placed around each joint to capture the relative position of an object region R with respect to joint j. The pixel overlap between the grid cell c and the object region R weights the contribution of the j^{th} joint and the k^{th} pose cluster.

5.2 Modeling long-term person-object interactions

Describing objects by a distribution of poses. We wish to characterize objects by the typical locations and poses of surrounding people. While 3D reasoning about people and scenes [35] has some advantages, reliable estimation of scene geometry and human poses in 3D is still an open problem. Moreover, deriving rich person-object co-occurrences from a single image is difficult due to the typically limited number of people in the scene and the noise of automatic human pose estimation. To circumvent these problems, we take advantage of the spatial co-occurrence of objects and people in the image plane. Moreover, we accumulate many human poses by observing scenes over an extended period of time.

In our setup we assume a static camera and consider larger objects such as sofas and tables which are less likely to change locations over time. We describe object region R in the image by the temporal statistics $\vec{h}^P$ of co-occurring human poses. Each person detection d is represented by the locations of $J(=14)$ body joints, indexed by j, and the assignment q^d_k of d's pose to a vocabulary of K^P discrete pose clusters (see Figure 5.3). To measure the co-occurrence of people and objects, we define a spatial grid of 9 cells c around each body joint j. We measure the overlap between the object region R and the grid cell $B^d_{j,c}$ by the normalized area of their intersection $\mathcal{I}(B_{j,c}, R) = \frac{|B_{j,c} \cap R|}{|B_{j,c}|}$. We then accumulate overlaps from all person detections $\mathcal{D}$ in a given video and compute one entry $h^P_{k,j,c}(R)$ of the histogram descriptor $\vec{h}^P(R)$ for region R as

$$h^P_{k,j,c}(R) = \sum_{d \in \mathcal{D}} \frac{\mathcal{I}(B^d_{j,c}, R)}{1 + \exp(-3s_d)} \, q^d_k, \tag{5.1}$$

where k, j, and c index pose clusters, body joints and grid cells, respectively. The contribution of each person detection in (5.1) is weighted by the detection score s_d. The values of q^d_k indicate the similarity of the person detection d with a pose cluster k. In the case of the hard assignment of d to the pose cluster $\tilde{k}$, $q^d_k = 1$ for $k = \tilde{k}$ and $q^d_k = 0$ otherwise. In our experiments we found that better results can be obtained using soft pose assignment as described in the next section.

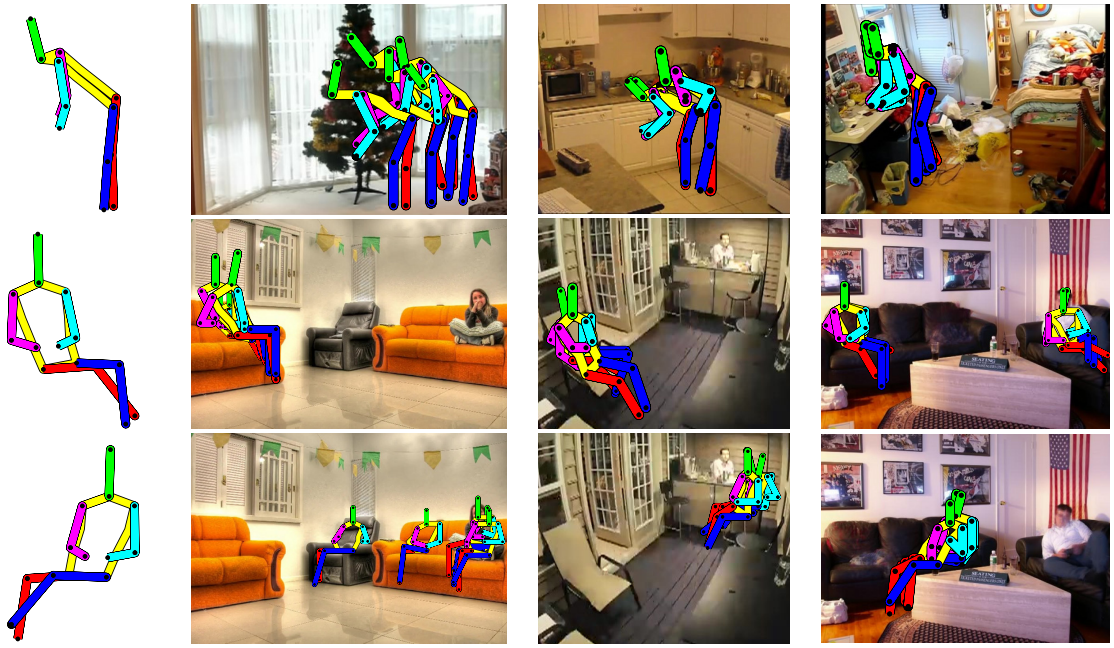


Figure 5.4: *Pose cluster and detection examples.* Left: example cluster means from our pose vocabulary. Right: person detections in multiple frames of time-lapse videos assigned to the pose clusters on the left.

Building a vocabulary of poses. We represent object-specific human actions by a distribution of *quantized* human poses. To compute pose quantization, we build a vocabulary of poses from person detections in the training set by unsupervised clustering.

In order to build the pose vocabulary, we first convert each detection d in the training video into a $2J$-dimensional pose vector $\vec{x}^d$ by concatenating mid-point coordinates of all detected body joints. We center and normalize all pose vectors in the training videos and cluster them by fitting a Gaussian Mixture Model (GMM) with K^P components via expectation maximization (EM). The components are initialized by the result of a K-means clustering, and during fitting we constrain the covariances to be diagonal. The resulting mean vectors $\vec{\mu}_k$, diagonal covariance matrices $\vec{\Sigma}_k$ and weights π_k for each pose cluster $k = 1, \cdots, K^P$ form our vocabulary of poses (see Figure 5.4). A pose vector $\vec{x}^d$ for a detection d can be described by a soft assignment to each of the $\vec{\mu}_k$ by computing the posterior probability vector $\vec{q}^d$, where

$$q_k^d = \frac{p(\vec{x}^d|\vec{\mu}_k, \vec{\Sigma}_k)\pi_k}{\sum_{j=1}^{K^P} p(\vec{x}^d|\vec{\mu}_j, \vec{\Sigma}_j)\pi_j}. \tag{5.2}$$

Person detection and pose estimation. We focus on detecting people in three body configurations common in indoor scenes: standing, sitting and reaching. We use the person detector from Yang and Ramanan [85], which has been shown to perform very well at both people detection and pose estimation, and train three separate models, one for each body configuration. We found that training 3 separate models improved pose estimation performance over using a single generic pose estimator. Given the static camera, we use background subtraction and pre-estimated room geometry [37] to discard some of the false person detections.

5.3 Modeling appearance and location

In addition to the distribution of poses we also model the appearance and absolute position of image regions. We build on the orderless bag-of-features representation [12] and describe the appearance of image regions by a distribution of visual words. We first densely extract SIFT descriptors [51] $f \in \mathcal{F}_k$ from image patches B^f of multiple sizes s_k for $k = 1, \cdots, S$ for all training videos and quantize them into visual words by fitting a GMM with K^A components. Each feature f is then soft-assigned to this vocabulary in the same manner as described in Eq. (5.2). This results in an assignment vector $\vec{q}^f$ for each feature. The K^A-dimensional appearance histogram $\vec{h}^A(R)$ for region R is computed as a weighted sum of assignment vectors $\vec{q}^f$

$$\vec{h}^A(R) = \sum_{k=1}^{S} \sum_{f \in \mathcal{F}_k} s_k^2 \mathcal{I}(B^f, R)\ \vec{q}^f, \tag{5.3}$$

where $s_k^2 \mathcal{I}(B^f, R)$ is the number of pixels belonging to both object region R and feature patch B^f.

Similar to [39], we also represent the absolute position of regions R within the video frame. This is achieved by spatially discretizing the frame into a grid of $m \times n$ cells, resulting in a $(m \times n)$-dimensional histogram $\vec{h}^L(R)$ for each region R. Here the i^{th} bin of $\vec{h}^L(R)$ is simply the proportion of pixels of the i^{th} cell of the grid falling into R.

5.4 Learning from long-term observations

Obtaining candidate object regions. As described in previous sections, we represent objects by accumulating statistics of human poses, image appearance and location at object regions R. Candidate object regions are obtained by over-segmenting video frames into super-pixels using the method and on-line implementation of [25]. As individual video frame may contain many people occluding the objects in the scene, we represent each video using a single "background frame" with (almost) no people obtained by temporal median filtering. Rather than relying on a single segmentation, we follow [39] and compute multiple overlapping segmentations by varying the parameters of the segmentation algorithm.

Learning object model. We train a classifier for each object class in a one-versus-all manner. The training data for each classifier is obtained by collecting all (potentially overlapping) super-pixels, R_i for $i = 1, \cdots, N$, from all training videos. For each region, we extract their corresponding pose, appearance and location histograms as described in Sections 5.2 and 5.3. The histograms are separately L_1-normalized and concatenated into a single K-dimensional feature vector $\vec{x}_i = [\tilde{\vec{h}}^P(R_i), \tilde{\vec{h}}^A(R_i), \tilde{\vec{h}}^L(R_i)]$, where $\tilde{\vec{h}}$ denotes L_1-normalized histogram $\vec{h}$. An object label y_i is then assigned to each super-pixel based on the surface overlap with the provided ground truth object segmentation in the training videos. We ensure that each super-pixel is assigned up to two ground truth object labels by setting surface overlap threshold of 34%. We train a binary support vector machine (SVM) classifier with the Hellinger kernel for each object class using the labelled super-pixels as training data. The Hellinger kernel is efficiently implemented using the explicit feature map $\Phi(\vec{x}_i) = \sqrt{(1/||\vec{x}_i||_1)\vec{x}_i}$ and a linear classifier. Finally, the outputs of individual SVM classifiers are calibrated with respect to each other by fitting a multinomial regression model from the classifiers output to the super-pixel labels [36]. The output of the learning stage is the K-dimensional weight vector $\vec{w}_y$ of the (calibrated) linear classifier for each object class y.

At test time, multiple super-pixel segmentations are extracted from the background frame of the test video and the individual classifiers are applied to each super-pixel. This leads to a confidence measure for each label and super-pixel. The confidence of a single image pixel is then the mean of the confidences of all the super-pixels it belongs to.

Time-lapse dataset. We extend the dataset of [L5] to 146 time-lapse videos containing a total of around 400,000 frames. Each video sequence shows people interacting with an indoor scene over a period of time ranging from a few minutes to several hours. The captured events include parties, working in an office, cooking or room-cleaning. The videos were downloaded from YouTube by placing queries such as "time-lapse party". Search results were manually verified to contain only videos captured with a stationary camera and showing an indoor scene. All videos are sparsely sampled in time with limited temporal continuity between consecutive frames. The dataset represents a challenging uncontrolled setup, where people perform natural non-staged interactions with objects in a variety of real indoor scenes.

We manually annotated each video with ground truth segmentation masks of eight frequently occurring semantic object classes: *'Bed'*, *'Sofa/Armchair'*, *'Coffee Table'*, *'Chair'*, *'Table'*, *'Wardrobe/ Cupboard'*, *'Christmas tree'* and *'Other'*. Similar to [37], the *'Other'* class contains various foreground room clutter such as clothes on the floor, or objects (e.g., lamps, bottles, or dishes) on tables. In addition to objects, we also annotated three room background classes: *'Wall'*, *'Ceiling'* and *'Floor'*. As the camera and majority of the objects are static, we can collect hundreds or even thousands of realistic person-object interactions throughout the whole time-lapse sequence by providing a single object annotation per video. The dataset is divided into 5 splits of around 30 videos with approximately the same proportion of labels for different objects. The dataset including the annotations is available at `http://www.di.ens.fr/willow/research/scenesemantics/`.

5.5 Experiments

Semantic labeling performance is measured by pixel-wise precision-recall curve and average precision (AP) for each object class. Table 5.1 shows the average precision for different object and room background classes for different feature combinations of our method. The performance of our method is compared to two baselines: the method of [37], trained on our data with semantic object annotations, and the deformable part model (DPM) of [24] trained over manually defined bounding boxes for each class. At test time, the DPM bounding boxes are converted to segmentation masks by assigning to each testing pixel the maximum score of any overlapping detection. Note that combining the proposed pose features with appearance (A+P) results in a significant improvement in overall performance, but further adding location features (A+L+P) brings little additional benefit, which suggests that spatial information in the scene is largely captured by the spatial relation to the human pose. The proposed method (A+L+P) also significantly outperforms both baselines. Example classification results for the proposed method are shown in Figure 5.5.

We have also evaluated our model on the task of estimating functional surfaces (walkable, sittable, reachable). For training and testing, we have provided ground truth functional surface masks for the dataset of [L5]. Our model achieves AP of 76%, 25% and 44% for *'Walkable'*, *'Sittable'* and *'Reachable'* surfaces, respectively, averaging a gain of 13% compared to [L5], which could be attributed to the discriminative nature of our model.

	DPM [24]	[37]	(A+L)	(P)	(A+P)	(A+L+P)
Wall	—	75±3.9	76±1.6	76±1.7	**82±1.2**	81±1.3
Ceiling	—	47±20	53±8.0	52±7.4	**69±6.7**	**69±6.6**
Floor	—	59±3.1	64±5.5	65±3.6	**76±3.2**	**76±2.9**
Bed	**31±20**	12±7.2	14±5.0	21±5.8	27±13	26±13
Sofa/Armchair	26±9.4	26±10	34±3.3	32±6.5	**44±5.4**	43±5.8
Coffee Table	11±5.4	11±5.2	11±4.4	12±4.3	**17±10**	**17±9.6**
Chair	9.5±3.9	6.3±2.8	8.3±2.7	5.8±1.4	11±5.4	**12±5.9**
Table	15±6.4	18±3.8	17±3.9	16±7.1	**22±6.2**	**22±6.4**
Wardrobe/Cupboard	27±10	27±8.2	28±6.4	22±1.1	**36±7.4**	**36±7.2**
Christmas tree	50±3.3	55±12	72±1.8	20±6.0	76±6.2	**77±5.5**
Other Object	12±6.4	11±1.2	7.9±1.9	13±4.2	**16±8.3**	**16±8.2**
Average	23±1.8	31±2.0	35±2.4	30±1.7	**43±4.4**	**43±4.3**

Table 5.1: Average precision (AP) for baselines of Felzenszwalb*et al.* [24] and Hedau*et al.* [37] compared to four different settings of our method: appearance and location features (A+L), person features (P), appearance and person features (A+P), appearance, location and person features combined (A+L+P).

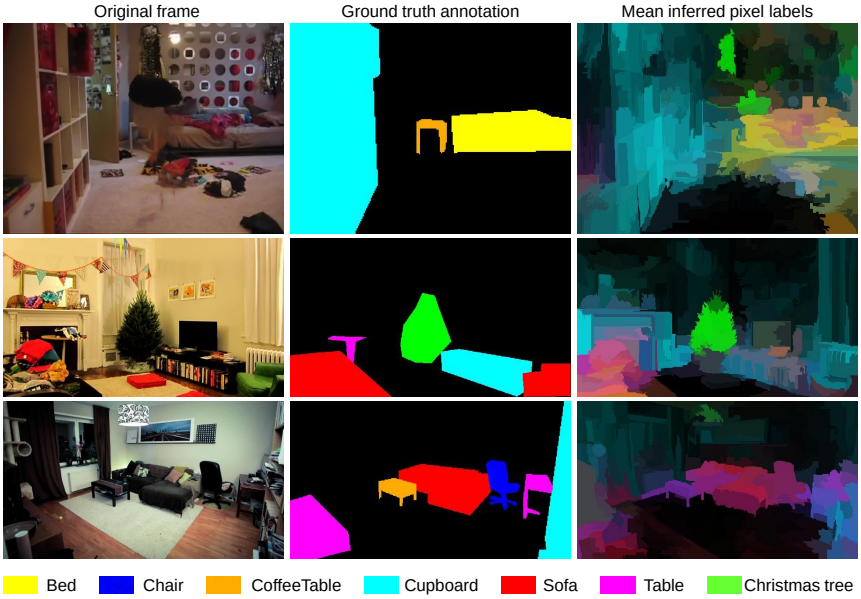


Figure 5.5: *Object soft segmentation.* Scene background with no people (left). Object ground truth (middle). Mean probability map for inferred objects (right).

5.6 Summary

This chapter has presented a first attempt at recognizing objects from their use in real unstaged environments. Despite the diversity of the data and the high level of noise in human pose estimation, our statistical pose descriptor contains significant signal enabling improved object recognition. This evidence is encouraging and suggest many directions for future work. First, we expect future improvements in pose estimation to directly improve our current method. As we learn correlations between objects and characteristic human poses, we also expect to improve pose estimation using knowledge about objects in the scene. While this work has looked at static person-object interactions, dynamic interactions should provide additional information to improve dynamic scene understanding in the future.

Chapter 6

Conclusions and outlook

This thesis has touched upon several problems in action recognition and dynamic scene understanding. The unifying theme of the work is to address the complexity of action recognition in realistic settings and to break away from controlled laboratory setups. The research area is new and the present work should be seen as a set of first steps rather than a final solution. The contributions of the thesis and future research opportunities are shortly summarized below.

Action classification and localization. Video representations introduced in Chapters 2 and 4 enable action recognition in realistic videos from movies, YouTube and television. At the time of publication [L14, L15] this work was among the first to consider such complex settings and the approach has been followed by others. Despite several notable improvements, e.g., [80], the accuracy of action recognition remains too low for practical applications, especially in the localization regime (see Section 3.3). Extending current methods towards structured representations in terms of objects, human poses, temporal order of actions, etc., is a promising and open research direction.

Another related and open problem is scalability. Current methods are limited by the efficiency to video datasets with tens of hours of video. Such datasets are likely to be too small for sampling and learning the full variability of actions. Modern applications also require new efficient action recognition methods that can scale to the processing of hundreds of years of video.

Weakly-supervised learning. Chapter 3 has introduced weakly-supervised learning in video from readily-available video scripts. While presented results are encouraging, much of the potential remains unexplored. Video scripts, indeed, provide rich information about people, their social relations and interactions, object properties, scene structure and the temporal structure of events. To better use this information, one could, for example, exploit the fact that objects, people and actions often co-occur. Knowing that "Rick sits down" in a video can help annotating a sitting down action if we can localize Rick and vice versa. Hence, jointly solving both action recognition and naming of characters in video could be beneficial. More generally, one could address the problem of automatic space-time script-to-video alignment. Knowing the alignment would provide automatic supervision for the learning of visual models of objects, actions and their attributes. Such models, however, are required to generate the alignment. One challenging research direction is, hence, to address both problems jointly and to explore redundancy in the large amount of video data to resolve the alignment and the learning of visual models within a single approach. Yet, another open problem related to weakly-supervised learning in video is the large variability in natural language expressions associated with similar kinds of dynamics scenes and events.

Recognition of functional properties. Computer vision is concerned with the automated interpretation of images and video streams. Today's research is (mostly) aimed at answering queries such as "Is this a picture of a dog?", "Is the person walking in this video?" (image and video categorisation) or sometimes "Find the dog in this photo" (object detection). While categorisation and detection are useful for many tasks, inferring correct class labels is not the final answer to visual recognition. The categories and locations of objects do not provide direct understanding of their *function*, i.e., how things work, what they can be used for, or how they can act and react. Neither do action categories provide direct understanding of subject's *intention*, i.e., the purpose of his/her activity. Such an understanding, however, would be highly desirable to answer currently unsolvable queries such as "Am I in danger?" or "What can happen in this scene?".

Addressing the above questions requires recognition of functional properties of objects and the purpose of human actions from visual observations. In my future work in the ERC project ACTIVIA I plan to leverage observations of people, i.e., their actions and interactions to automatically learn the use, the purpose and the function of objects and scenes from visual data. The work presented in Chapter 5 is an initial step in this direction, but many open challenges remain. In particular, modeling interactions requires new structural video representations beyond bag-of-features and other available methods. Learning interaction models from the video data also requires new weakly-supervised learning techniques, since the full manual annotation of large-scale video is prohibitive. While the functional object recognition is not a new problem and has been studied in 80's and early 90's, e.g., in [74], today's increasing amount of available video data and the recent progress in visual object and action recognition will likely enable new ways to successfully address this problem in the future.

From a practical point of view, recognition of functional object properties and intentions of people will directly support a number of important applications such as the prediction of likely future events and the alert of abnormal activities. In the future, automatic learning of human behaviour from large-scale visual observations will enable support of our every-day activities from assembling IKEA furniture to helping us with educating a child.

References

[L1] H. Azizpour and I. Laptev. Object detection using strongly-supervised deformable part models. In *Proc. ECCV*, 2012.

[L2] V. Delaitre, D. Fouhey, I. Laptev, J. Sivic, A. Gupta, and A. Efros. Scene semantics from long-term observation of people. In *Proc. ECCV*, 2012.

[L3] V. Delaitre, J. Sivic, and I. Laptev. Learning person-object interactions for action recognition in still images. In *Proc. NIPS*, 2011.

[L4] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *Proc. ICCV*, 2009.

[L5] D. Fouhey, V. Delaitre, A. Gupta, A. Efros, I. Laptev, and J. Sivic. People watching: Human actions as a cue for single view geometry. In *Proc. ECCV*, 2012.

[L6] I. Junejo, E. Dexter, I. Laptev, and P. Pérez. Cross-view action recognition from temporal self-similarities. In *Proc. ECCV*, pages II: 293–306, 2008.

[L7] I.N. Junejo, E. Dexter, I. Laptev, and P. Perez. View-independent action recognition from temporal self-similarities. *IEEE-PAMI*, 33(1):172–185, 2011.

[L8] I. Laptev. On space-time interest points. *IJCV*, 64(2/3):107–123, 2005.

[L9] I. Laptev. Improvements of object detection using boosted histograms. In *Proc. BMVC*, pages III:949–958, 2006.

[L10] I. Laptev. Improving object detection with boosted histograms. *IVC*, 27(5):535–544, 2009.

[L11] I. Laptev, S. Belongie, P. Pérez, and J. Wills. Periodic motion detection and segmentation via approximate sequence alignment. In *Proc. ICCV*, pages I:816–823, 2005.

[L12] I. Laptev, B. Caputo, C. Schüldt, and T. Lindeberg. Local velocity-adapted motion events for spatio-temporal recognition. *CVIU*, 108(3):207–229, 2007.

[L13] I. Laptev and T. Lindeberg. Local descriptors for spatio-temporal recognition. In *First International Workshop on Spatial Coherence for Visual Motion Analysis*, volume 3667 of *LNCS*, pages 91–103. Springer Verlag, 2004.

[L14] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proc. CVPR*, 2008.

[L15] I. Laptev and P. Pérez. Retrieving actions in movies. In *Proc. ICCV*, 2007.

[L16] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, and F. Stentiford. Video copy detection: A comparative study. In *Proc. CIVR*, pages 371–378, 2007.

[L17] J. Lezama, K. Alahari, J. Sivic, and I. Laptev. Track to the future: Spatio-temporal video segmentation with long-range motion cues. In *Proc. CVPR*, 2011.

[L18] M. Marszałek, I. Laptev, and C. Schmid. Actions in context. In *Proc. CVPR*, 2009.

[L19] M. Rodriguez, I. Laptev, J. Sivic, and J.-Y. Audibert. Density-aware person detection and tracking in crowds. In *Proc. ICCV*, 2011.

[L20] M. Rodriguez, J. Sivic, I. Laptev, and J.-Y. Audibert. Data-driven crowd analysis in videos. In *Proc. ICCV*, 2011.

[L21] C. Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. In *Proc. ICPR*, pages III:32–36, 2004.

[L22] M.M. Ullah, S.N. Parizi, and I. Laptev. Improving bag-of-features action recognition with non-local cues. In *Proc. BMVC*, 2010.

[L23] H. Wang, M.M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *Proc. BMVC*, 2009.

[1] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. Jordan. Matching words and pictures. *J. Machine Learning Research*, February 2003.

[2] T.L. Berg, A.C. Berg, J. Edwards, M. Maire, R. White, Y.W. Teh, E.G. Learned-Miller, and D.A. Forsyth. Names and faces in the news. In *Proc. CVPR*, 2004.

[3] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological review*, 94(2):115, 1987.

[4] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *Proc. ICCV*, pages 1395–1402, 2005.

[5] O. Boiman and M. Irani. Detecting irregularities in images and in video. In *Proc. ICCV*, pages I:462–469, 2005.

[6] P. Buehler, M. Everingham, and A. Zisserman. Learning sign language by watching tv (using weakly aligned subtitles). In *Proc. CVPR*, 2009.

[7] Calvin upper-body detector. http://www.vision.ee.ethz.ch/~calvin/calvin_upperbody_detector.

[8] C.C. Chang and C.J. Lin. *LIBSVM: a library for support vector machines*, 2001. http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[9] O. Chomat and J.L. Crowley. Probabilistic recognition of activity using local appearance. In *Proc. CVPR*, pages II:104–109, 1999.

[10] O. Chum and A. Zisserman. An exemplar model for learning object classes. In *Proc. CVPR*, 2007.

[11] T. Cour, B. Sapp, C. Jordan, and B. Taskar. Learning from ambiguously labeled images. In *Proc. CVPR*, 2009.

[12] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, 2004.

[13] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, pages I:886–893, 2005.

[14] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *Proc. ECCV*, pages II: 428–441, 2006.

[15] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for static human-object interactions. In *SMiCV, CVPR*, 2010.

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

[17] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley, 2001.

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

[19] M. Everingham, L. Van Gool, C. Williams, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results.
http://www.pascal-network.org/challenges/VOC/voc2006.

[20] M. Everingham, A. Zisserman, C. Williams, L. Van Gool, M. Allan, C. Bishop, O. Chapelle, N. Dalal, T. Deselaers, G. Dorko, S. Duffner, J. Eichhorn, J. Farquhar, M. Fritz, C. Garcia, T. Griffiths, F. Jurie, D. Keysers, M. Koskela, J. Laaksonen, D. Larlus, B. Leibe, H. Meng, H. Ney, B. Schiele, C. Schmid, E. Seemann, J. Shawe-Taylor, A. Storkey, S. Szedmak, B. Triggs, I. Ulusoy, V. Viitaniemi, and Zhang J. The 2005 pascal visual object classes challenge. In *Selected Proceedings of the First PASCAL Challenges Workshop*, 2005.

[21] A. Farhadi, M. Hejrati, A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: generating sentences for images. In *Proc. ECCV*, 2010.

[22] A. Fathi, X. Ren, and J. Rehg. Learning to recognize objects in egocentric activities. In *Proc. CVPR*, 2011.

[23] C. Fellbaum, editor. *Wordnet: An Electronic Lexical Database*. Bradford Books, 1998.

[24] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE-PAMI*, 32(9):1627–1645, 2010.

[25] P.F. Felzenszwalb and D.P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, September 2004.

[26] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. of Comp. and Sys. Sc.*, 55(1):119–139, 1997.

[27] M. Fritz, B. Leibe, B. Caputo, and B. Schiele. Integrating representative and discriminative models for object category detection. In *Proc. ICCV*, pages II:1363–1370, 2005.

[28] A. Gaidon, Z. Harchaoui, and C. Schmid. Temporal Localization of Actions with Actoms. *IEEE-PAMI*, March 2013.

[29] J. Gall, A. Fossati, and L. van Gool. Functional categorization of objects using real-time markerless motion capture. In *Proc. CVPR*, 2011.

[30] J. Gibson. *The ecological approach to visual perception*. Boston: Houghton Mifflin, 1979.

[31] H. Grabner, J. Gall, and L. Van Gool. What makes a chair a chair? In *Proc. CVPR*, 2011.

[32] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *Proc. CVPR*, 2009.

[33] A. Gupta and L. Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *Proc. ECCV*, 2008.

[34] A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE-PAMI*, 2009.

[35] A. Gupta, S. Satkin, A. Efros, and M. Hebert. From 3d scene geometry to human workspace. In *Proc. CVPR*, 2011.

[36] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, 2003.

[37] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *Proc. ICCV*, 2009.

[38] M. Hein and O. Bousquet. Hilbertian metrics and positive definite kernels on probability measures. In *Proc. AISTATS*, 2005.

[39] D. Hoiem, A. Efros, and M. Hebert. Geometric context from a single image. In *Proc. ICCV*, 2005.

[40] B.K.P Horn and B.G Schunck. Determining optical flow. *Artificial intelligence*, 17(1):185–203, 1981.

[41] A. Howard and T. Jebara. Learning monotonic transformations for classification. In *Proc. NIPS*, 2007.

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

[43] N. Jojic, A. Perina, and M. Murino. Structural epitome: a way to summarize one's visual experience. In *Proc. NIPS*, pages 1027–1035, 2010.

[44] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *Proc. ICCV*, pages I:166–173, 2005.

[45] H. Kjellstrom, J. Romero, D. Martinez, and D. Kragic. Simultaneous visual recognition of manipulation actions and manipulated objects. In *Proc. ECCV*, 2008.

[46] A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3D-gradients. In *Proc. BMVC*, 2008.

[47] A. Kläser, M. Marszałek, C. Schmid, and A. Zisserman. Human Focused Action Localization in Video. In *International Workshop on Sign, Gesture, and Activity (SGA) in Conjunction with ECCV*, 2010.

[48] S. Lazebnik, C. Schmid, and J. Ponce. Semi-local affine parts for object recognition. In *Proc. BMVC*, volume 2, pages 959–968, 2004.

[49] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *Proc. CVPR*, pages II: 2169–2178, 2006.

[50] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *Proc. CVPR*, pages 3337–3344, 2011.

[51] D. Lowe. Distinctive image features form scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[52] J. Luo, B. Caputo, and V. Ferrari. Who's doing what: Joint modeling of names and verbs for simultaneous face and pose annotation. In *Proc. NIPS*, 2009.

[53] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox. Learning to parse natural language commands to a robot control system. In *Proc. of the 13th Int'l Symposium on Experimental Robotics (ISER)*, 2012.

[54] K. Mikolajczyk, B. Leibe, and B. Schiele. Local features for object class recognition. In *Proc. ICCV*, pages II:1792–1799, 2005.

[55] A. Ng and M. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Proc. NIPS*, 2002.

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

[57] J.C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *Proc. ECCV*, pages 392–405, 2010.

[58] J. M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of Visual Comm. and Image Representation*, 6(4):348–365, 1995.

[59] OpenNLP. http://opennlp.sourceforge.net.

[60] V. Ordonez, G. Kulkarni, and T.L. Berg. Im2text: Describing images using 1 million captioned photographs. In *Proc. NIPS*, 2011.

[61] S. E. Palmer. *Vision science: photons to phenomenology*. MIT Press, Cambridge, Mass., 1999.

[62] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Proc. ECCV*, pages 143–156, 2010.

[63] P. Peursum, G. West, and S. Venkatesh. Combining image regions and human activity for indirect object recognition in indoor wide-angle views. In *Proc. ICCV*, 2005.

[64] A. Prest, C. Schmid, and V. Ferrari. Weakly supervised learning of interactions between humans and objects. *IEEE-PAMI*, 2011.

[65] M. Rodriguez, J. Ahmed, and M. Shah. Action mach: A spatio-temporal maximum average correlation height filter for action recognition. In *Proc. CVPR*, 2008.

[66] S. Sadanand and J.J. Corso. Action bank: A high-level representation of activity in video. In *Proc. CVPR*, pages 1234–1241, 2012.

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

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

[69] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Camb. U. P., 2004.

[70] E. Shechtman and M. Irani. Space-time behavior based correlation. In *Proc. CVPR*, pages I:405–412, 2005.

[71] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *Proc. CVPR*, pages 1–8, 2007.

[72] J. Sivic, M. Everingham, and A. Zisserman. "who are you?" - learning person specific classifiers from video. In *Proc. CVPR*, 2009.

[73] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, pages 1470–1477, 2003.

[74] L. Stark and K.W. Bowyer. Achieving generalized object recognition through reasoning about association of function to structure. *IEEE-PAMI*, 13(10):1097–1104, 1991.

[75] M. Stark, P. Lies, M. Zillich, J. Wyatt, and B. Schiele. Functional object class detection based on learned affordance cues. In *Proc. ICVS*, 2008.

[76] Kevin Tang, Li Fei-Fei, and Daphne Koller. Learning latent temporal structure for complex event detection. In *Proc. CVPR*, pages 1250–1257, 2012.

[77] M. Turek, A. Hoogs, and R. Collins. Unsupervised learning of functional categories in video scenes. In *Proc. ECCV*, 2010.

[78] B. Tversky, J.B. Morrison, and J. Zacks. On bodies and events. In A. Meltzoff and W. Prinz, editors, *The Imitative Mind*. Cambridge University Press, Cambridge, 2002.

[79] P. Viola and M.J. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004.

[80] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *IJCV*, March 2013.

[81] X. Wang, K. Tieu, and E. Grimson. Learning semantic scene models by trajectory analysis. In *Proc. ECCV*, 2006.

[82] Y. Wang and G. Mori. A discriminative latent model of image region and object tag correspondence. In *Proc. NIPS*, 2010.

[83] G. Willems, T. Tuytelaars, and L. VanGool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *Proc. ECCV*, 2008.

[84] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans. Maximum margin clustering. *Proc. NIPS*, 17:1537–1544, 2004.

[85] Y. Yang and D. Ramanan. Articulated pose estimation using flexible mixtures of parts. In *Proc. CVPR*, 2011.

[86] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *Proc. CVPR*, 2010.

[87] B. Yao, A. Khosla, and L. Fei-Fei. Classifying actions and measuring action similarity by modeling the mutual context of objects and human poses. In *Proc. ICML*, 2011.

[88] L. Zelnik-Manor and M. Irani. Event-based analysis of video. In *Proc. CVPR*, pages II:123–130, 2001.

[89] L. Zelnik-Manor and M. Irani. Statistical analysis of dynamic actions. *IEEE-PAMI*, 28(9):1530–1535, 2006.

[90] 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, June 2007.

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