Weakly supervised methods for learning actions and objects

Alessandro Prest

▶ **To cite this version:**

Alessandro Prest. Weakly supervised methods for learning actions and objects. Computer science. Eidgenössische Technische Hochschule Zürich (ETHZ), 2012. English. NNT : . tel-00758797

HAL Id: tel-00758797

https://theses.hal.science/tel-00758797v1

Submitted on 29 Nov 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

DISS. ETH NO. 20612

Weakly supervised methods for learning actions and objects

A dissertation submitted to

ETH ZURICH

for the degree of

Doctor of Sciences (Dr. sc. ETH Zürich)

presented by

Alessandro Prest
Dipl.-Ing. Elektrotechnik und Informationstechnik
born June 20, 1983
citizen of Ponte nelle Alpi, Italy

accepted on the recommendation of

Prof. Dr. Vittorio Ferrari, examiner
Res. Dir. Cordelia Schmid, co-examiner
Prof. Dr. Thomas Brox, external referee
Prof. Dr. Marc Pollefeys, co-examiner

2012

TO DIANA AND ROBERTO.

Abstract

Modern Computer Vision systems learn visual concepts through examples (i.e. images) which have been manually annotated by humans. While this paradigm allowed the field to tremendously progress in the last decade, it has now become one of its major bottlenecks. Teaching a new visual concept requires an expensive human annotation effort, limiting systems to scale to thousands of visual concepts from the few dozens that work today. The exponential growth of visual data available on the net represents an invaluable resource for visual learning algorithms and calls for new methods able to exploit this information to learn visual concepts without the need of major human annotation effort.

As a first contribution, we introduce an approach for learning human actions as interactions between persons and objects in realistic images. By exploiting the spatial structure of human-object interactions, we are able to learn action models automatically from a set of still images annotated only with the action label (weakly-supervised). Extensive experimental evaluation demonstrates that our weakly-supervised approach achieves the same performance of popular fully-supervised methods despite using substantially less supervision.

In the second part of this thesis we extend this reasoning to human-object interactions in realistic video and feature length movies. Popular methods represent actions with low-level features such as image gradients or optical flow. In our approach instead, interactions are modeled as the trajectory of the object wrt to the person position, providing a rich and natural description of actions. Our interaction descriptor is an informative cue on its own and is complimentary to traditional low-level features.

Finally, in the third part we propose an approach for learning object detectors from real-world web videos (i.e. YouTube). As opposed to the standard paradigm of learning from still images annotated with bounding-boxes, we propose a technique to learn from videos known only to contain objects of a target class. We demonstrate that learning detectors from video alone already delivers good performance requiring much less supervision compared to training from images annotated with bounding boxes. We additionally show that training from a combination of weakly annotated videos and fully annotated still images improves over training from still images alone.

Sommario

I moderni sistemi di Computer Vision apprendono nuovi concetti visivi grazie ad esempi forniti da umani. Nonostante questo paradigma abbia permesso un notevole progresso negli ultimi dieci anni, oggi rappresenta uno dei maggiori ostacoli per la Computer Vision. Insegnare un nuovo concetto visivo infatti ha un grande costo per l'umano, il quale deve fornire al sistemi migliaia di esempi annotati manualmente. Questo paradigma, oggigiorno applicato a poche dozzine di concetti visivi differenti, è di fatto inapplicabile per l'insegnamento di migliaia di essi. In questo contesto, la crescita esponenziale di immagini e video disponibili in rete rappresenta una preziosa risorsa per gli algoritmi di apprendimento visivo. È sentito il bisogno di nuovi algoritmi in grado di sfruttare questa enorme informazione per l'insegnamento di concetti visivi, limitando il più possibile la supervisione umana.

Come primo contributo, proponiamo un sistema che apprende azioni umane, più specificatamente, interazioni tra persone e oggetti in immagini. Sfruttando la struttura geometrica delle interazioni uomo-oggetto, il nostro sistema impara automaticamente modelli di azione da un insieme di immagini annotate solo con il nome dell'azione rappresentata. Un'estensiva analisi sperimentale dimostra come il nostro approccio raggiunga le stesse prestazioni di metodi concorrenti nonostante un uso molto inferiore di supervisione umana.

Nel nostro secondo lavoro, le idee di cui sopra vengono estese ad interazione uomo-oggetto in video. Metodi tradizionali rappresentano le azioni con features di basso livello quali gradienti di immagini o optical flow. Nel nostro approccio invece, le interazioni sono modellate come traiettorie dell'oggetto rispetto alla posizione della persona, fornendo una descrizione ricca e naturale di azioni. La nostra rappresentazione, pur essendo informativa di per sé, è complementare alle features tradizionali di basso livello.

Infine, nel nostro lavoro più recente proponiamo un sistema che apprende da video di YouTube, dove l'unica informazione di supervisione viene dal nome dell'oggetto mostrato in un particolare video. Al meglio della nostra conoscenza questo è il primo lavoro ad affrontare questo particolare problema. Dimostriamo come l'apprendimento da video offra già buone prestazioni, diminuendo drasticamente la quantità di supervisione necessaria rispetto all'apprendimento da immagini fisse. Dimonstriamo inoltre come l'apprendimento

da una combinazione di video e immagini fisse offra prestazioni ancora migliori rispetto a modelli appresi solamente da immagini.

Acknowledgements

First and foremost I extend my gratitude to my supervisors Cordelia Schmid and Vittorio Ferrari for all the constructive guidance I have received during this PhD. I am also extremely grateful to my co-referee, Prof. Thomas Brox, who accepted to examine the thesis on a short notice.

Senior colleagues including Dr. Thomas Deselaers and Dr. Christian Leistner have provided invaluable advice and help during my research. My most sincere thanks goes out to them.

The PhD period is also a great occasion for making friends and I've been extremely lucky in this respect. Gabriele Fanelli, Mukta Prasad, Stephan Gammeter, Thomas Mensink, Marcin Eichner, Bogdan Alexe, Daniel Kuettel and Matthieu Guillaumin are just a few of the wonderful people I've met along the way. I'm particularly obliged to my bandmates Adrien Gaidon and Josip Krapac for all the great gigs at INRIA and endless music discussions.

My business partners Luigi and Luca Boschin played an important role in my career, bridging the gap between my expertise and the market. This collaboration resulted in the founding of two successful companies that today allow me to make a living out of artificial intelligence.

Navigating the challenging waters of research at international level requires optimism and positive energy. Fostering these qualities has been the exclusive work of my mum and dad who offered me the most playful and light-hearted childhood I could ever imagine.

Last, but by no means least, I thank my friends and family in Italy, France, Switzerland, Americas and elsewhere for their support and encouragement throughout.

Contents

List of Figures

List of Tables

1

Introduction

The past decade will be remembered as one of maturity for artificial intelligence (AI). Several successful applications such as Google Goggles, Siri, IBM Watson have positively impacted people's everyday life. These systems are able to interpret in real-time highly complex natural signals, in the form of text, audio or video data: a task thought the exclusive domain of human intelligence before the two-thousands.

More specifically, an AI system has the purpose of mapping natural signals (appropriately digitized) created by or for humans into a pre-determined output space. The output is then communicated by the AI system to humans directly or to other systems. These signals, such as a spoken command or a picture, are effortlessly interpreted by humans but represents a real challenge from the machine perspective. The development of automated methods for their interpretation is the main focus of the artificial intelligence research community.

This thesis discusses methods of computer vision, a branch of AI where the input for the system is represented by images and videos depicting visual scenes. Most computer vision tasks have the objective of recognizing visual concepts such as the presence of a particular object or the occurrence of a specific event in the input data and estimate their spatial and temporal extent in the scene.

This chapter is intended as an introduction to the computer vision areas treated in this thesis and more specifically to the problem of manual supervision in visual learning (sec. 1.2). Many of the technical terms used later are introduced here in a natural manner, preparing the non-specialist reader for the following chapters. Sec. 1.1 presents the specific research areas in which this thesis operates while sec. 1.3.1 and 1.3.2 offer a review of existing work and are concluded illustrating the motivation of our work in that particular areas. Finally, sec. 1.4 closes the introduction providing an overview on the content of the following chapters.

***Figure 1.1**: An overview of different action recognition tasks tackled in this thesis. The first two rows deal with classyfing a whole video as showing a particular action. The third row illustrates the task of localizing human-object interactions in static images. In the last row instead we show spatio-temporal localization of actions in video.*

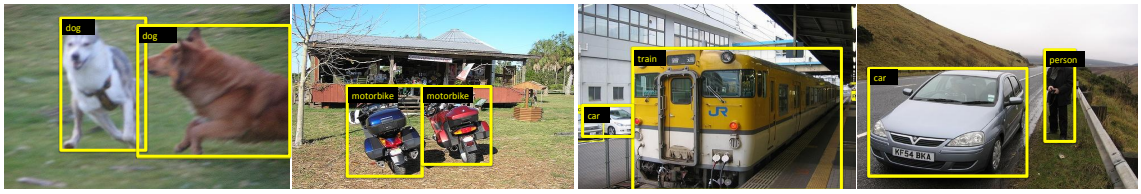

Figure 1.2: *Example images from the* PASCAL07 *dataset, which is the object detection task tackled in this thesis. Images from this dataset contain objects of different categories whose spatial extent has to be precisely determined up to a rectangular subregion of the image.*

1.1 Problem definition

This thesis operates in two major computer vision areas: human action recognition and object recognition. These areas deal with the problem of recognizing the presence of a particular object or the occurrence of a specific human action in images and videos.

Object recognition can be solved at different level of details, i.e. from classifying a whole image or video as of depicting or not a particular object, to determining the spatial extent of the object up to a rectangular region, to providing the outlines of the object (see first row of fig. 1.3). More specifically, *object detection* deals with recognizing objects in images as belonging to certain categories (i.e. cars, dogs, . . .) and determining their spatial extent up to a rectangular sub-region (bounding-box). In chapter 4 we experiment on a challenging object detection dataset named PASCAL07. Fig. 1.2 shows exemplar images from this dataset, illustrating how one image can contain multiple objects of different categories in cluttered scenes and realistic imagining conditions.

Analogously to object recognition, *human action recognition* can be solved at different levels of detail. Fig. 1.1 gives an overview on popular human action recognition tasks. The top row shows examples from [Schuldt *et al.* 2004] and [Gorelick *et al.* 2007], where the task is to classify a whole video as depicting a particular action. The second row shows example from the video datasets of [Gupta *et al.* 2009] and [Messing *et al.* 2009]. The task is the classification of actions involving humans and objects. Although the level of detail is equivalent to the previous task, the subtle variations between different actions (i.e. eating banana vs. eating chips) require more advanced motion and temporal analysis. The third row shows examples from our dataset [Prest *et al.* 2011] and that of [Gupta *et al.* 2009]. In this case the task is to localize the actor and the corresponding action in realistic static images. Finally, the bottom row shows examples from the *Coffee & Cigarettes* dataset of [Laptev and Perez 2007]. The task here is the detection in space (bounding-box on the actor location) and time (beginning and end frame of the action) of actions in feature length movies.

1.2 Manual supervision in visual learning

Recently computer vision research significantly progressed thanks to the combination of machine learning methods and large databases of images and videos. This allowed a shift from physical/geometrical models of visual concepts hand-crafted by a designer to more automated methods able to learn visual concepts from human-annotated examples. These annotations make the correspondence between visual input and expected output explicit and are used by the system to learn models of the visual concepts. These models will then be used to recognize concepts in novel images or video.

By representing visual data as an aggregation of local features such as the orientation of the image contours [Lowe 1999], machine learning methods [Dalal and Triggs 2005] are able to learn fairly complicated visual concepts provided that enough human-annotated data is available. These approaches proved to be more robust than their hand-crafted counterparts and easier to adapt to new and more complex visual concepts.

In the current visual learning paradigm, humans act as teachers by providing a large set of annotated examples called training data. Depending on the desired level of detail of the system and on its learning approach, different levels of supervision can be employed. Fig. 1.3 illustrates the most common level of supervision for tasks such as object recognition (first row) and human action recognition (second row).

The exponential growth of visual data available on the net today can be an invaluable resource for visual learning algorithms and calls for new methods able to learn thousands of different visual concepts. As of today however, state-of-the-art methods for object detection deal only with a limited number of classes and achieve modest accuracy [Everingham and others 2010]: a few tens of categories operating at a positive detection rate of about 50% while producing a false positive every four images. In the case of action recognition, state-of-the-art methods [Yao *et al.* 2011a] are able to classify humans as performing one out of nine actions with an overall accuracy of about 60%.

The main challenge laying ahead for supervised methods to approach human performance in both accuracy and number of concepts is the cost of manually annotating training data. One has to consider that with state-of-the-art approaches [Chen *et al.* 2010, Zhang *et al.* 2011, Vedaldi *et al.* 2009] learning an object category requires thousands of training images annotated with the precise location of the object. Moreover for learning human action models an even higher degree of supervision is required such as the actor silhouette [Gupta *et al.* 2009], the location of his limbs [Yao *et al.* 2011b] or the attributes of the scene [Yao *et al.* 2011a]. A popular workaround is represented by crowdsourcing tools such as Amazon Mechanical Turk [1]. These tools provide a streamlined and cost-effective way to manually annotate large amount of data by exploiting "annotation labour" in under-developed countries.

[1] http://www.mturk.com

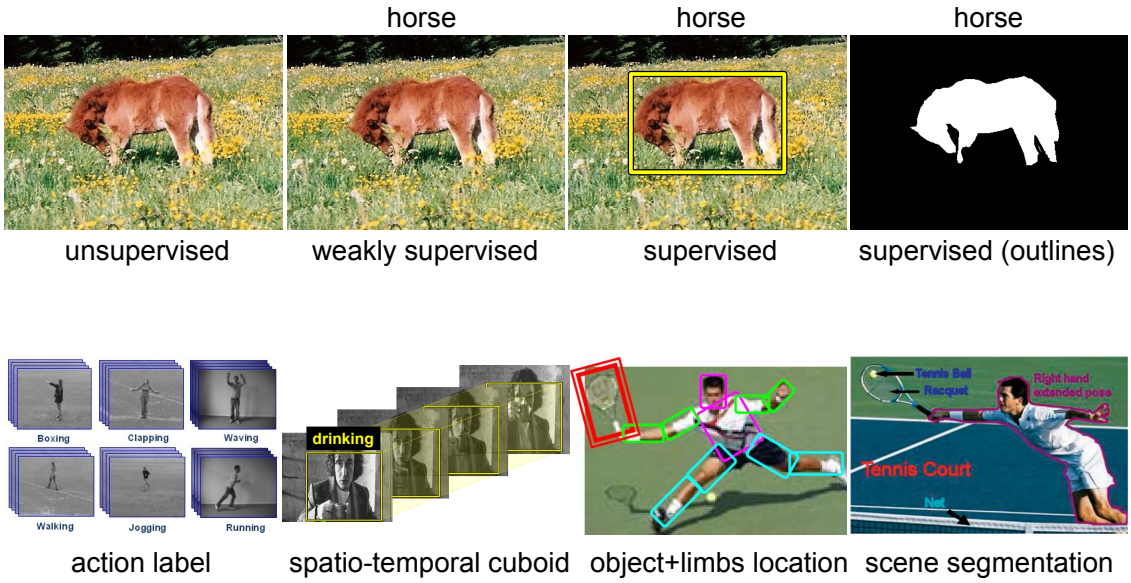

Figure 1.3: *Different levels of manual supervision ordered from less (left) to more detailed (right). Supervision levels for object recognition (first row) are fairly standardized and easy to be categorized: unsupervised data consists of the image itself without any accompanying information. Weakly-supervised data comprises a label specifying the class of the depicted object. Bounding-box annotations specify also the spatial extent of the displayed object up to a rectangular region of the image. Finally the object outline is a binary mask denoting the pixelwise extent of the object. Methods for human action recognition require instead a more diverse set of annotations in order to be trained (second row). The original method of [Schuldt* et al. *2004] operates on video clips labeled only with the action name. Spatio-temporal bounding-boxes (second illustration) represent a higher degree of supervision as they mark the location of actors in space and time as in [Laptev and Perez 2007]. Static images approaches for action recognition (third and fourth images) tend to require an higher degree of supervision. [Yao and Fei-Fei 2010b] requires precise location of the limbs as they rely on action-specific human pose models. The method of [Gupta* et al. *2009] requires an even more detailed human-provided description of the scene. This includes the actor outline, the location of the objects depicted in the scene and a label for the background.*

This thesis counters the previous trend and instead aims at reducing the amount of annotations needed for visual learning. Popular works tackled this problem, for example, in the object detection area [Fergus *et al.* 2003, Todorovic and Ahuja 2006, Deselaers *et al.* 2010] to train detectors using only the image label as opposed to bounding-boxes. Our contribution in reducing the amount of supervision for visual learning is two-fold: (i) we introduce weakly-supervised methods for *human action recognition* in static images and video, (ii) we learn object detectors from *consumer video* known only to contain a particular object class. The following section offers an overview of related work in these areas.

1.3 Related work

1.3.1 Human action recognition in images and video

Several early works in human action recognition deal with video data. They rely on simple measurements such as optical flow or spatio-temporal gradients extracted from video clips. An example are the popular bags of spatio-temporal features, initially introduced in [Dollar *et al.* 2005, Schuldt *et al.* 2004, Zelnik-Manor and Irani 2001]. These techniques extract spatio-temporal features over video clips, quantize them and use a frequency histogram to represent the clips. Recent extensions model the temporal structure of actions as a composition of smaller sub-parts [Gaidon *et al.* 2011, Laptev *et al.* 2008, Niebles *et al.* 2010]. Furthermore, they determine the temporal extent of video clips optimal for a bag-of-features representation in realistic movies [Duchenne *et al.* 2009, Satkin and Hebert 2010].

Another line of work describes the human tracks based on low-level features such as optical flow [Efros *et al.* 2003] or based on the silhouette of the humans [Bobick and Davis 2001, Yilmaz and Shah 2005, Gorelick *et al.* 2007]. Specifically, [Yilmaz and Shah 2005, Gorelick *et al.* 2007] propose human-centered approaches for action recognition based on spatio-temporal volumes (STV) obtained by accumulating silhouette information over time. They then extract information such as speed, direction and shape to characterize the STV. In [Bobick and Davis 2001] they extract silhouettes from a single view and aggregate differences between subsequent frames of an action sequence resulting in a binary motion energy image. Temporal information is included through a motion history image. The method proposed in [Efros *et al.* 2003] operates on sports footage. They compensate camera movement by tracking the person and calculate optical flow in person-centered tracks.

All of the above mentioned human-centric approaches operate either with static cameras, i.e., human can be located based on background subtraction, or with simple backgrounds from which human can be extracted easily, as for example football or ice hockey fields.

More recent human-centric approaches [Laptev and Perez 2007, Mikolajczyk and Uemura 2008, Kläser *et al.* 2010, Rodriguez *et al.* 2008] deal instead with action localization in realistic video. Laptev and Perez [Laptev and Perez 2007] aggregate local spatio-temporal features over time into a spatio-temporal grid. They use keyframe priming to refine the output of their method. In [Mikolajczyk and Uemura 2008] authors also adopt a human-centric approach where vocabularies of local motion and shape features are combined with a voting approach. Liu et al. [Liu *et al.* 2009] propose a combination of static and motion low-level features and efficient techniques for mining the most discriminative ones in realistic youtube videos. The method proposed in [Kläser *et al.* 2010] localizes actions in space and time by first extracting human tracks and then detecting specific actions within the tracks using a sliding window classifier.

Ultimately, the weakly-supervised approaches by [Ikizler and Forsyth 2008, Ikizler-Cinbis *et al.* 2009] attempt to decrease the amount of supervision necessary for training action classifiers. In [Ikizler and Forsyth 2008] instead of modeling the human pose as a single observation, they train an HMM for each body part. This reduces the combinatorial complexity as they learn a motion model for each limb separately. Training videos for learning actions are obtained inexpensively from YouTube in [Ikizler-Cinbis *et al.* 2009]. They use an incremental approach to image harvesting where the initial set largely affects the final variety of action performances.

Human-object interactions A growing number of works focus their attention on a particular class of actions involving an interaction between a human and an object (see fig. 1.4). Interestingly, these works either require a higher degree of supervision (i.e. human pose models to distinguish similar actions) or operate under simplistic conditions to facilitate the recognition (i.e. static or uniform background). In the following we review the most important works in this area and their main limitations.

Several authors tackle the problem of recognizing human-object interactions in video [Filipovych and Ribeiro 2008, Filipovych and Ribeiro 2010, Gupta *et al.* 2009, Matikainen *et al.* 2010, Messing *et al.* 2009]. [Filipovych and Ribeiro 2008, Filipovych and Ribeiro 2010] model human-object interactions based on the trajectory and appearance of spatio-temporal interest points. Their approach is demonstrated in controlled videos taken by a static camera against a static, uniform background. Importantly, the scene is seen from the actor's viewpoint.

Messing et al. [Messing *et al.* 2009] introduce a dataset of human-object interactions recorded in controlled conditions and propose a descriptor based on the velocity history of tracked point features. Matikainen et al. [Matikainen *et al.* 2010] extends this descriptor to include relations between pairs of tracked points and quantize them into vocabularies.

[Gupta *et al.* 2009] model the action object and the human-object motion for classifying interactions between humans and objects. The motion features used in their approach rely

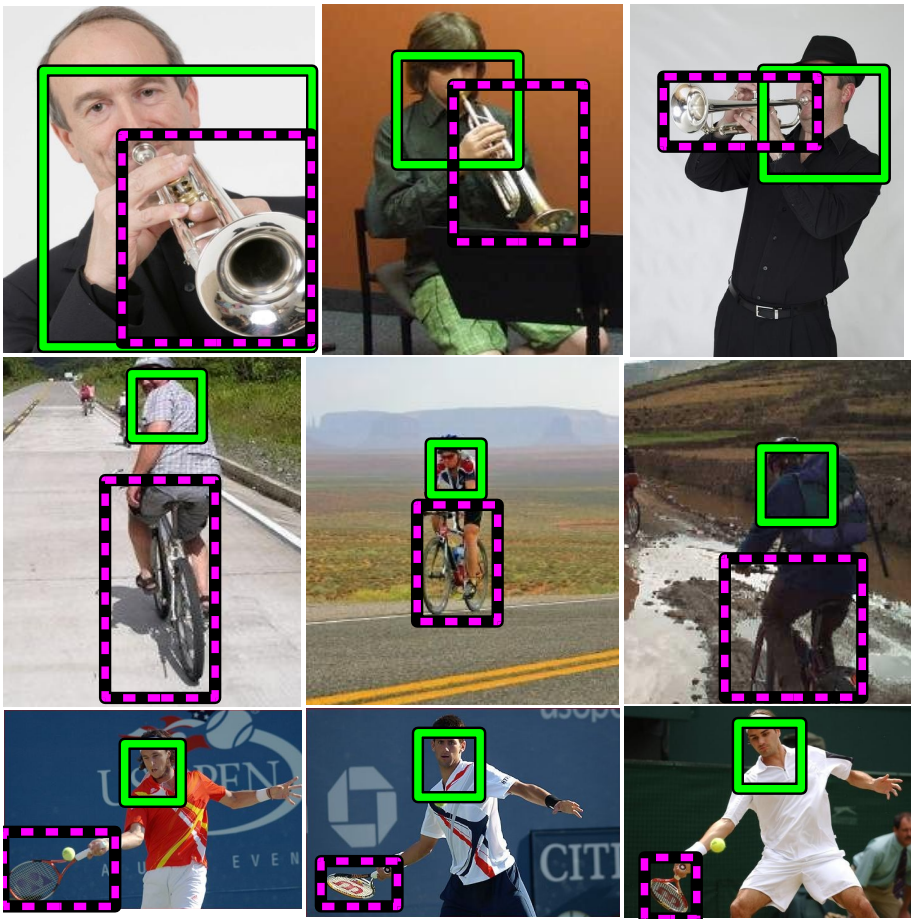

***Figure 1.4**: Examples of human (green box) and object (dashed purple) interactions.*

on hand trajectories to model how objects are reached and grasped. These features rely on motion extracted based on background subtraction, which limits its applicability to static cameras and backgrounds (as opposed to uncontrolled video such as feature films).

The works of [Yao and Fei-Fei 2010b] and [Gupta *et al.* 2009] learn to recognize human-object spatial interactions in static images. These approaches operate in a fully supervised setting, requiring training images with annotated object locations as well as human silhouettes [Gupta *et al.* 2009] or limb locations [Yao and Fei-Fei 2010b]. Another work by Yao et al. [Yao and Fei-Fei 2010a] deals with a somewhat different formulation of the problem. Their goal is to discriminate subtle situations where a human is holding an object without using it versus a human performing a particular action with the object (e.g. 'holding a violin' vs 'playing a violin'). Note how this model requires manually localized humans both at training and testing time.

All the above methods either operate on simplistic dataset or require a substantial amount of human supervision in order to learn new interactions. However, human-object interactions contain additional structure that can be exploited to reduce the amount of supervision

needed to learn corresponding action models. In fact interactions are often the main characteristic of an action (fig. 1.4): for example, the action 'tennis forehand' can be described as a human holding a tennis racket in a certain position. Characteristic features are the object *racket* and its spatial relation to the human. Similarly, the actions 'riding bike' and 'playing trumpet' are defined by an object and its spatial relation to the human.

In chapters 2 and 3 we put this reasoning into practice and use the person as a spatial anchor for learning new interactions. More specifically, we introduce a state-of-the-art human detector (sec. 2.4) able to robustly localize humans in uncontrolled and realistic images and videos. This provides a spatial coordinate frame common across images which reduces the uncertainty in the location of the object. Moreover, the appearance of the object and its spatial arrangement wrt the person is recurrent across images of the same interaction (see fig. 1.4). This allows to: (i) learn from weakly-supervised images known only to contain a person involved in a particular object interaction (chapter 2); (ii) learn motion models that describe interactions in realistic video (chapter 3) and are complementary to traditional low-level features.

1.3.2 Learning object detectors

The standard way to train state-of-the-art object detectors is to gather a large, diverse set of images and annotate them manually. The typical level of supervision needed is a bounding-box for each object instance [Felzenszwalb *et al.* 2009, Vedaldi *et al.* 2009, Viola and Jones 2001] and in general the performance of a detector increases with the number of annotated instances [Vijayanarasimhan and Grauman 2011].

Recently, object detection methods have made significant progress thanks also to the introduction of standardized benchmarks such as the PASCAL VOC Challenge [Everingham and others 2010] that allow to evaluate and compare different approaches and keep track of the community progress. However, compared to what human beings can achieve, current methods deal with a limited number of categories and achieve only modest accuracy.

The current limits are in part due to the cost of producing training data and therefore many work investigated the possibility to learn from unlabeled or weakly annotated data [Arora *et al.* 2007, Crandall and Huttenlocher 2006, Chum and Zisserman 2007, Fergus *et al.* 2003, Winn and Jojic 2005, Borenstein and Ullman 2004, Cao and Li 2007]. However, most of these approaches have been demonstrated to work on simplistic datasets such as Caltech4 or Weizmann horses where the object of interest covers a substantial part of the image and background clutter is limited. These conditions make it easier to spot recurring and discriminative object patterns and thus learn appropriate object models. More recently a number of methods [Blaschko *et al.* 2010, Chum and Zisserman 2007, Kim and Torralba 2009, Deselaers *et al.* 2010, Pandey and Lazebnik 2011, Siva and Xiang 2011] addressed the problem of weakly-supervised learning of object detectors in more realistic datasets. These methods typically try to approximately localize object instances while learning

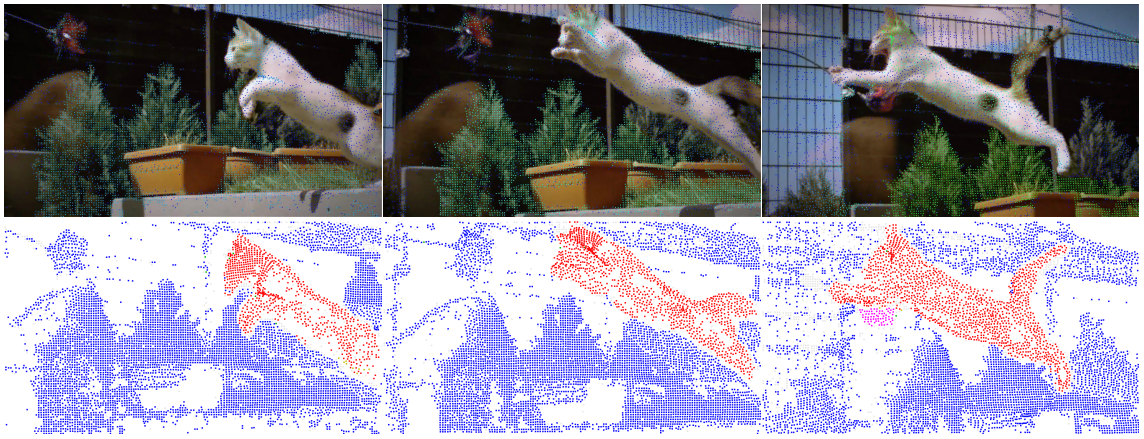

***Figure 1.5**: Motion segmentation obtained from dense point tracks [Sundaram et al. 2010] applied on a cat video downloaded from youtube.*

a model of the class. However, learning a detector without location annotation is very difficult and performance is still below fully supervised methods [Deselaers *et al.* 2010, Pandey and Lazebnik 2011, Siva and Xiang 2011]. All the methods described so far learn object detectors using static images as training data.

Interestingly, with a few exceptions [Ali *et al.* 2011, Leistner *et al.* 2011, Ramanan *et al.* 2006], learning from videos has been disregarded by the vision community. In one of the earliest works, Ramanan et al. [Ramanan *et al.* 2006] showed how to build part-based animal models for tracking and detection without explicit supervisory information. Ommer et al. [Ommer *et al.* 2009] learn detection models as 3D point clouds using structure-from-motion. They train from controlled, hand-recorded video and the model can detect objects in test video, but not in images. Leistner et al. [Leistner *et al.* 2011] train a part-based random forest object detector from images and use patches extracted from videos to regularize the learning of the trees. Their approach captures the appearance variation of local patches from video and is tested on rather simple benchmarks.

Yet, video offers a rich source of data and is becoming more easily accessible through internet sources such as YouTube. The benefits of video include: (i) it is easier to automatically segment the object from the background based on motion information, (ii) each video shows significant appearances variations of an object, and (iii) a set of videos provides a large number of training images, as each video consists of many frames.

Fig. 1.5 illustrates the points above. Standard tool for motion segmentation [Sundaram *et al.* 2010] are able to: (i) precisely differentiate the spatial extent of relevant object wrt background, (ii) capture appearance variation of the object, (iii) obtain as many training examples as frames in the video.

In chapter 4 we show how weakly-supervised video from youtube can be a valuable and inexpensive source of training data for learning object detectors. We propose a fully au-

tomatic pipeline for spatio-temporal localization of objects in videos, where the localized objects are then used as positive training data. Moreover we show that a model learned on a combination of weakly-supervised video and fully-supervised still images outperforms detectors learned from fully-supervised images alone.

1.4 Contributions

This thesis introduces novel methods for weakly-supervised visual learning tackling different areas of computer vision. In the following we summarize the main contributions detailed in the next chapters:

1. Chapter 2 deals with action recognition in static images and more specifically is focused on human-object interactions (sec. 1.3.1). By exploiting the structure of human-object interactions, we are able to learn action model automatically from a set of still images annotated only with the action label (weakly-supervised). Our approach relies on humans detected in a set of images depicting the action and determines the action object and its spatial relation wrt the human. Its final output is a probabilistic model of the human-object interaction, i.e. the spatial relation between the human and the object. Extensive experimental evaluation demonstrate that our weakly-supervised approach achieves the same performance of state-of-the-art fully-supervised methods despite using drastically less supervision. This method obtained the highest classification accuracy among 7 participants at the Pascal VOC Action Recognition Challege 2011 [Everingham *et al.*].

2. The concepts introduced in chapter 2 are extended to the video domain in chapter 3. We introduce an approach for learning human actions as interactions between persons and objects in realistic videos. Unlike previous work that typically represents actions with low-level features such as image gradients or optical flow, our approach offers a rich description of an action by explicitly localizing in space and time both the object and the person. We use the person as a spatial anchor and represent an action as the trajectory of the object wrt to the person position. Experimental results show that our explicit human-object model is an informative cue for action recognition. Furthermore is complementary to traditional low-level descriptors: their combination improves over their individual performance as well as over the state-of-the-art.

3. Chapter 4 explores techniques for exploiting the large amount of consumer video available on the internet to learn object detectors. Object detectors are typically trained on a large set of still images annotated by bounding-boxes. We instead propose an approach for learning from real-world web videos known only to contain

objects of a target class. We propose a fully automatic pipeline that localizes objects in a set of videos of the class and learns a static-image detector for it. To the best of our knowledge this is the first work addressing this particular problem. We demonstrate that learning detectors from video alone already delivers good performance using drastically less supervision. We additionally show that training from a combination of weakly annotated videos and fully annotated still images improves over training from still images alone.

1.5 Publications

The topics covered in this thesis have been published in the following papers:

1. A. Prest, C. Schmid, and V. Ferrari. Weakly supervised learning of interactions between humans and objects. IEEE Transactions on Pattern Analysis and Machine Intelligence (2012)

2. A. Prest, V. Ferrari, and C. Schmid. Explicit modeling of human- object interactions in realistic videos. IEEE Transactions on Pattern Analysis and Machine Intelligence (Accepted for publication, to appear)

3. A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. IEEE Conference on Computer Vision and Pattern Recognition (2012)

   Released the YouTube-Objects dataset comprising over 6 hours of weakly-annotated consumer videos of 10 object classes [2]

[2] http://groups.inf.ed.ac.uk/calvin/learnfromvideo

2

Weakly supervised learning of interactions between humans and objects in still images

2.1 Introduction

Human action recognition is one of the most challenging problems in computer vision. It is important for a wide range of applications, such as video indexing and surveillance, but also image search. It is a challenging task due to the variety of human appearances and poses.

Our approach defines an action as the interaction between a human and an object. Interactions are often the main characteristic of an action (fig. 2.9, 2.10, 2.11 and 2.14). For example, the action 'tennis serve' can be described as a human holding a tennis racket in a certain position. Characteristic features are the object *racket* and its spatial relation to the human. Similarly, the actions 'riding bike' and 'wearing a hat' are defined by an object and its relation to the human.

In this chapter we introduce a weakly supervised approach for learning interaction models between humans and objects from a set of images depicting an action. We automatically localize the relevant object as well as its spatial relation to the human (fig. 2.9, 2.10, 2.11 and 2.14). Our approach is weakly supervised in that it can learn from images annotated only with the action label, without being given the location of humans nor objects.

The rest of the chapter is organized as follows. In sec. 2.2 we review related work. Sec. 2.3 first gives an overview of our method, and then sections 2.4 to 2.6 explain its components in detail. Sec. 2.4 introduces our part-based model for human detection under different viewpoints and imaging conditions. This is a necessary step towards our human-object interaction model, which is the main contribution of this chapter (sec. 2.5). In sec. 2.6 we build a complete action recognition classifier by combining our interaction model with traditional low-level cues.

In sec. 2.7 we present experiments on the dataset of Gupta et al. [Gupta *et al.* 2009] and on a new human-object interaction dataset. The experiments show that our method, despite using far less supervision, obtains classification performance comparable to [Gupta *et al.* 2009] and [Yao and Fei-Fei 2010b]. Moreover, we demonstrate that our model learns meaningful human-object spatial relations. Ultimately, sec. 2.8 presents experiments on the PASCAL Action 2010 dataset [Everingham and others 2010], where our method outperforms the winner of the challenge for action classes involving humans and objects. Furthermore we show that how our method can also handle actions not involving objects (e.g. walking).

2.2 Related work

Most popular methods for action recognition operate on video. They either learn a spatio-temporal model of an action [Schuldt *et al.* 2004, Laptev and Perez 2007, Mikolajczyk and Uemura 2008] or are based on human pose [Sullivan and Carlsson 2002, Ikizler-Cinbis *et al.* 2009]. Spatio-temporal models measure the motion characteristics for a human action. They are, for example, based on bags of space-time interest points [Schuldt *et al.* 2004, Dollar *et al.* 2005, Laptev *et al.* 2008] or represent the human action as a distribution over motion features localized in space and time [Mikolajczyk and Uemura 2008, Laptev and Perez 2007, Willems *et al.* 2009]. Pose-based models learn the characteristic human poses from still images. The pose can, for example, be represented by a histogram-of-gradient (HOG) [Ikizler-Cinbis *et al.* 2009, Thurau and Hlavac 2008] or based on shape correspondences [Sullivan and Carlsson 2002].

The are relatively less works on action recognition in still images [Wang *et al.* 2006, Ikizler *et al.* 2008, Yang *et al.* 2010]. In one of the earliest work [Wang *et al.* 2006] the authors represent sport actions (figure skating, baseball, ...) by key poses or prototypes and match edge representations of actions to labeled key poses. They learn action clusters in an unsupervised fashion and manually provide action class labels after the clustering. The approach presented in [Ikizler *et al.* 2008] is also based on pose: they extract the human pose and within its outline they detect oriented rectangles and store them in a circular histogram. This representation is a discriminative feature for classifying actions from their dataset of images collected from the web. A recent work by [Yang *et al.* 2010] treats the pose of the person in the image as a latent variable to help recognition. Their approach is trained in an integrated fashion that jointly considers poses and actions.

Most related to our approach are the works of Yao et al. [Yao and Fei-Fei 2010b] and Gupta et al. [Gupta *et al.* 2009] who also learn human-object spatial interactions. However, these approaches operate in a fully supervised setting, requiring training images with annotated object locations as well as human silhouettes [Gupta *et al.* 2009] or limb locations [Yao and Fei-Fei 2010b]. Another work by Yao et al. [Yao and Fei-Fei 2010a] deals

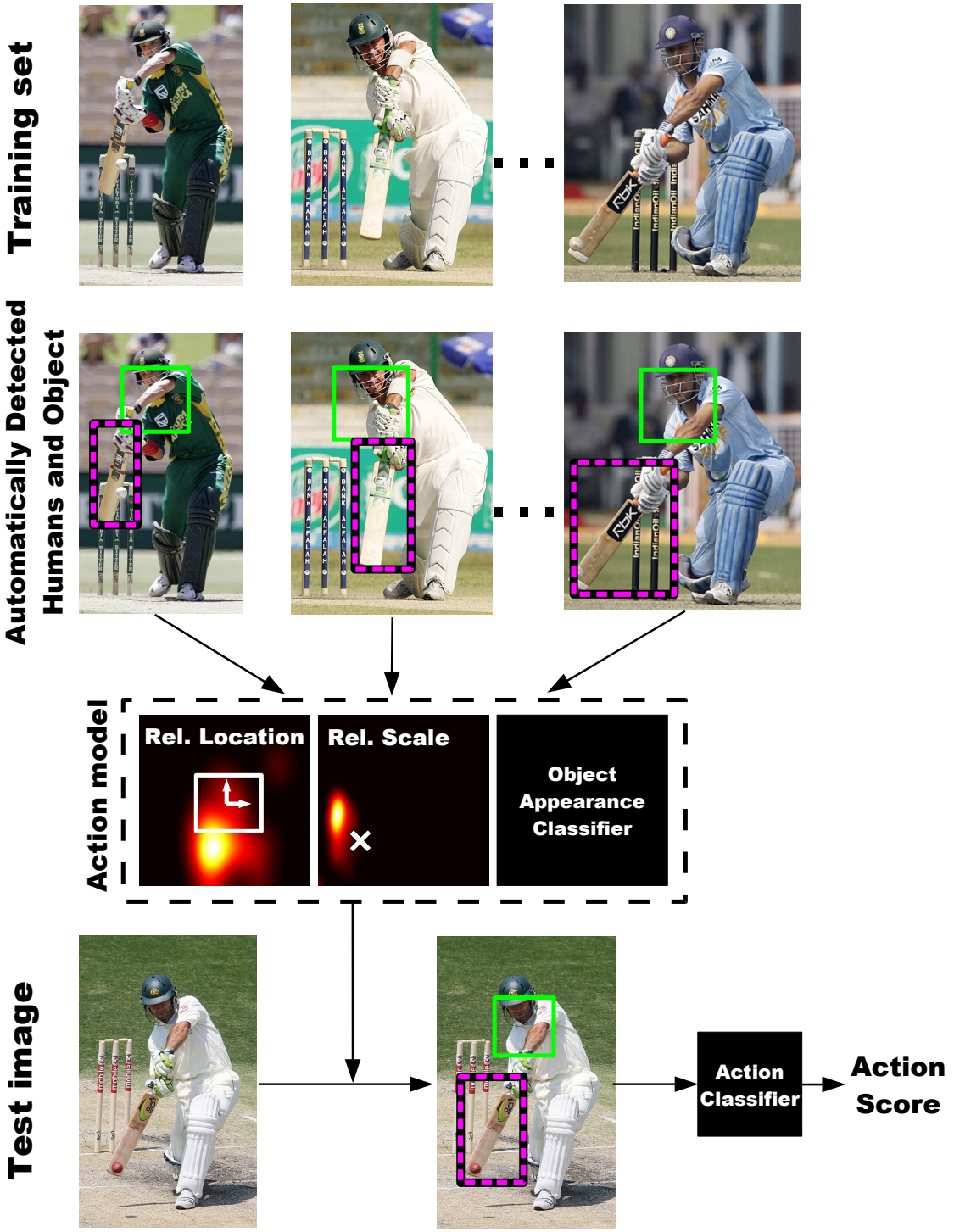

***Figure 2.1**: Overview of our approach. See main text for details.*

with a somewhat different formulation of the problem. Their goal is to discriminate subtle situations where a human is holding an object without using it versus a human performing a particular action with the object (e.g. 'holding a violin' vs 'playing a violin'). Note how this model requires manually localized humans both at training and testing time. The method presented in [Yao *et al.* 2011a] represent a further extension of their previous work [Yao and Fei-Fei 2010a]. They use attributes and parts for recognizing human actions in still images. Their model captures objects and other discriminative local patches (i.e. scene context, human body parts, ...) which are discriminative for a particular action.

A recent work [C. Desai 2010] models the contextual interaction between human pose and nearby objects, but requires manually annotated human and object locations at training time for learning the pose and object models. A previous work by the same authors [Desai *et al.* 2009] models spatial relations between object classes such as cars and motorbikes for object localization in a fully supervised setting.

Interactions are used to improve human pose estimation in [Gupta *et al.* 2008], by inferring pose parameters (i.e. joint angles) from the properties of objects involved in a particular human-object interaction.

Co-occurrence relations between humans and objects have been exploited for action recognition in videos by [Ikizler-Cinbis and Sclaroff 2010]. However, these relations are looser than what we propose, as there is no spatial modeling of the interaction.

2.3 Overview of the method

2.3.1 Training

Our method takes as input a set of training images showing the humans performing the action. Our approach runs over the following stages (fig. 2.1):

(1) Detect humans in the training set (sec. 2.4). Our overall detector combines several detectors for different human parts, including face, upper-body, and fully body. This improves coverage as it can detect human at varying degrees of visibility. The detector provides the human reference frame necessary for modeling the spatial interaction with the object in stages (2) and (3).

(2) Localize the action object on the training set (sec. 2.5.1). The basic idea is to find an object recurring over many images at similar relative positions with respect to the human and with similar appearance between images. Related to our approach are weakly supervised methods for learning object classes [Fergus *et al.* 2003, Winn *et al.* 2005, Deselaers *et al.* 2010], which attempt to find objects as recurring appearance patterns.

(3) Given the localized humans and objects from stages (1) and (2), learn the probability distribution of human-object spatial relations, such as relative location and relative size.

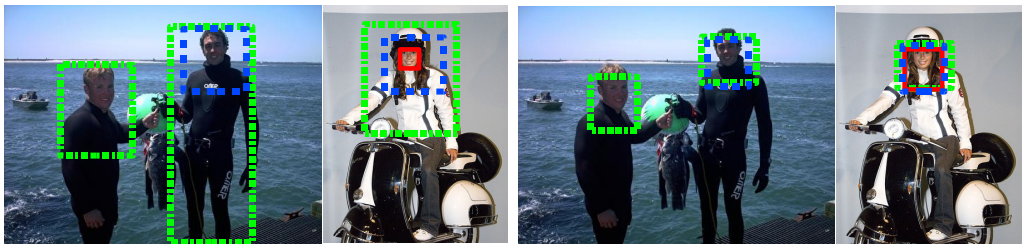

Figure 2.2*: Left: Detection windows returned by the individual detectors (Green: FB + UB1, Blue: UB2, Red: F). Right: corresponding regressed windows.*

This defines the human-object interaction model (sec. 2.5.4). Additionally we learn an object appearance classifier based on the localized objects from (2). This appearance classifier together with the human-object interaction model constitute the action model.

(4) Based on the information estimated in steps 1-3, we train a binary action classifier to decide whether a novel test image contains an instance of this action class (sec. 2.6).

2.3.2 Testing

Given a novel test image $\mathcal{I}$ and n different action models learned in the previous subsection, we want to assign one of the n possible action labels to $\mathcal{I}$ (fig. 2.1 bottom):

(1) Detect the single most prominent human in $\mathcal{I}$.

(2) For each action model, find the best fitting location for the action object given the detected human, the human-object interaction model and the object appearance classifier.

(3) Compute different features based on the information extracted in (1) and (2).

(3) Classify $\mathcal{I}$ in an action class, based on the information estimated in steps (1) and (4) (sec. 2.6). This uses the n classifiers trained in sec. 2.3.1 stage (4).

2.4 A part-based human detector

In real world images of human actions the person can be fully or partially visible (fig. 2.5, 2.10 and 2.11). In this context a single detector (full person, an upper-body or face) is insufficient. Our detector build on the one by Felzenszwalb et al. [Felzenszwalb *et al.* 2009]; it trains several detectors for different human parts, adds a state-of-the-art face detector and learns how to combine the different part detectors. Our combination strategy goes beyond the maximum score selection strategy of [Felzenszwalb *et al.* 2009] and is shown experimentally to outperform their approach (sec. 2.4.5). Furthermore, it provides the human reference frame necessary for modeling the spatial interaction with the object.

2.4.1 Individual part detectors.

We use four part detectors: one for the full human body (FB), two for the upper-body (UB1, UB2) and one for the face (F). For the fully body detector (FB) and the first upper-body detector (UB1) we use the two components of the human detector by [Felzenszwalb *et al.* 2009][1] learnt on the the PASCAL VOC07 training data [Everingham *et al.* 2007a]. Note that we use the two components as two separate part detectors. For the second upper body detector (UB2) we train [Felzenszwalb *et al.* 2009] on another dataset of near-frontal upper-bodies [Ferrari *et al.* 2008][2]. Therefore, UB2 is specialized to the frontal case, which occurs frequently in real images. Our experiments show UB2 to provide detections complementary to UB1 (sec. 2.4.5).

For the face detector (F) we use the technique of [Rodriguez 2006], which is is similar to the popular Viola-Jones detector [Viola and Jones 2001], but replaces the Haar features with local binary patterns, providing better robustness to illumination changes [Heusch *et al.* 2006]. The detector is trained for both front and side views.

2.4.2 Mapping to a common reference frame.

As the detection windows returned by different detectors cover different areas of the human body, they must be mapped to a common reference frame before they can be combined. Here we learn regressors for this mapping (fig. 2.2).

For each part detector we learn a linear regressor $\mathrm{R}(w, p)$ mapping a detection window w to a common reference frame. A regressor R is defined by

$$\mathrm{R}(w, p) = (x - Wp_1,\ y - Hp_2,\ Wp_3,\ Wp_3p_4) \tag{2.1}$$

where $w = (x, y, W, H)$ is a detection window defined by the top-left co-ordinates (x, y), its width W and its height H. The regression parameters $p = (p_1, p_2, p_3, p_4)$ are determined from the training data as follows.

We have a set of n training pairs of detection windows w^i and corresponding manually annotated ground-truth reference windows h^i. We find the optimal regression parameters p^* as

$$p^* = \arg\max_p \sum_{i=1}^{n} \mathrm{IoU}(h^i, \mathrm{R}(w^i, p)) \tag{2.2}$$

where $\mathrm{IoU}(a, b) = |a \cap b|/|a \cup b|$ is the intersection-over-union between two windows a, b. The optimal parameters p^* assure the best overlap between the mapped detection windows $\mathrm{R}(w^i, p)$ and the ground-truth references h^i.

[1] Code available at http://people.cs.uchicago.edu/∼pff/latent.

[2] Data available at www.robots.ox.ac.uk/∼vgg/software/UpperBody

Fig. 2.3 shows an example of the original stickman annotation and the common reference frame derived from it. The height of the reference frame is given by the distance between the top point of the head stick and the mid point of the torso stick. The width is fixed to 90% of the height.

2.4.3 Clustering part detections.

After mapping detection windows from the part detectors to a common reference frame, detections of the same person result in similar windows. Therefore, we find small groups of detections corresponding to different persons by clustering all mapped detection windows for an image in the 4D space defined by their coordinates.

Clustering is performed with a weighted dynamic-bandwidth mean-shift algorithm based on [Comaniciu *et al.* 2001]. At each iteration the bandwidth is set proportionally to the expected localization variance of the regressed windows (i.e. to the diagonal of the window defined by the center of the mean-shift kernel in the 4D space). This automatically adapts the clustering to the growing error of the part detectors with scale.

To achieve high recall it is important to set a very low threshold on the part detectors. This results in many false-positives which cause substantial drift in the traditional mean-shift procedure. To maintain a robust localization, at each iteration we compute the new cluster center as the mean of its members *weighted* by their detection scores. The final mean-shift location in the 4D space also gives a weighted average reference window for each cluster, which is typically more accurately localized than the individual part detections in the cluster.

2.4.4 Discriminative score combination.

Given a cluster C containing a set of part detections, the goal is to determine a single combined score for the cluster. Each cluster C has an associated representative detection window computed as the weighted mean of the part detection windows in C.

To compute to the score of a cluster, we use the 4D vector c where each dimension corresponds to one of the detectors. The value of an entry c_d is set to the maximum detection score for detector d within the cluster. If the cluster does not contain a detection for a detector d, we set $c_d = \tau_d$, with τ_d the threshold at which the detector is operating (see sec. 2.4.5). Given the 4D score vector for each cluster, we learn a linear SVM to separate positive (human detections) from negative examples. The score for a test image is then the confidence value of the SVM. Section 2.4.5 explains how we collect positive ($\mathcal{T}^+$) and negative ($\mathcal{T}^-$) training examples. The training set for this score-combiner SVM is the same used to train the regressors.

Figure 2.3*: Example of an annotated image from the ETHZ PASCAL Stickmen dataset. Left: the original stickman annotation. Right: the common reference frame we derived from the sticks.*

2.4.5 Experimental evaluation.

The experimental evaluation is carried out on the ETHZ PASCAL Stickmen dataset [Eichner and Ferrari 2009] [3]. It contains 549 images from the PASCAL VOC 2008 person class. In each image, one person is annotated by line segments defining the position and orientation of the head, torso, upper and lower arms (fig. 2.3). As we want the common reference frame to be visible in most images, we set it as a square window starting from the top of the head and ending at the middle of the torso (fig. 2.2). Note that this choice has no effect on the combined human detector.

We build our positive training set $\mathcal{T}^+$ out of the first 400 images and use the remaining 149 as a positive test set $\mathcal{S}^+$. The negative examples are obtained from Caltech-101 [Fergus and Perona 2003] as well as from PASCAL VOC [Everingham *et al.* 2007a] [Everingham *et al.* 2008]. We end up with 5158 negative images: 3956 are randomly selected as the negative training set $\mathcal{T}^-$ while the remaining form the negative test set $\mathcal{S}^-$.

The optimal regressor parameters p^* are learnt on the positive training set $\mathcal{T}^+$ (as described in sec.2.4.2).

The score-combiner SVM is trained on the clusters obtained from the entire training set $\mathcal{T}^+ \cup \mathcal{T}^-$. All clusters from $\mathcal{T}^-$ are labeled as negative examples. Clusters from $\mathcal{T}^+$ are labeled as positive examples if their representative detection has an IoU with a ground-truth person bounding-box greater than 50%. All other clusters from $\mathcal{T}^+$ are discarded, as their ground-truth label is unknown (although an image in ETHZ PASCAL Stickmen might contain multiple persons, only one is annotated). Note that before clustering we

[3] Available at http://www.vision.ee.ethz.ch/∼calvin/datasets.html.

only keep detections scoring above a low threshold τ_d, such as to remove weak detections likely to be false positives.

Fig. 2.4 shows a quantitative evaluation on our test set $\mathcal{S}^+ \cup \mathcal{S}^-$ as a precision-recall curve. The recall axis indicates the percentage of annotated humans that were correctly detected (true positives, IoU with the ground-truth greater than 50%). All detections in $\mathcal{S}^-$ are counted as false positives. Notice how in $\mathcal{S}^+$ only one human per image is annotated. Hence, only true positives in $\mathcal{S}^+$ are counted in the evaluation and all other detections are discarded, as their ground-truth label is unknown. Precision is defined as the ratio between the number of true positives and the total number of detections at a certain recall value.

Our combined human detector UB1+FB+UB2+F brings a considerable increase in average precision compared to the state-of-the-art human detector of [Felzenszwalb *et al.* 2009], which it incorporates. For a fair comparison, its detection windows are also regressed to a common reference frame (using the same regressor as in our combined detector).

Note that the person model of [Felzenszwalb *et al.* 2009] uses its two components (FB and UB1) in a 'max-score-first' combination: if two detections from the two different components overlap by more than 50% IoU, then the lower scoring one is discarded. In the experiment UB1+FB we use our novel combination strategy to combine only the two components UB1 and FB. This performs significantly better than the original model [Felzenszwalb *et al.* 2009], further demonstrating the power of our combination strategy. In all experiments all detection windows are regressed to the same common reference frame as ours.

Although the face detector performs much below the other detectors, it is valuable in close-up images, where the other detectors do not fire.

2.5 Learning human-object interactions

This section presents our human-object interaction model and how to learn it from weakly supervised images. The goal is to automatically determine the object relevant for the action as well as its spatial relation to the human. The intuition behind our human-object model is that some geometric properties relating the human to the action object are stable across different instances of the same action. Let's imagine a human playing a trumpet: the trumpet is always at approximately the same relative distance with respect to the human. We model this intuition with spatial cues involving the human and the object. We measure them relative to the position and scale of the reference frame provided by the human detector from sec. 2.4. This makes the cues comparable between different images.

Our model (subsec. 2.5.1) incorporates several cues (subsec. 2.5.3). Some relate the human to the object while others are defined purely by the appearance of the object. Once the

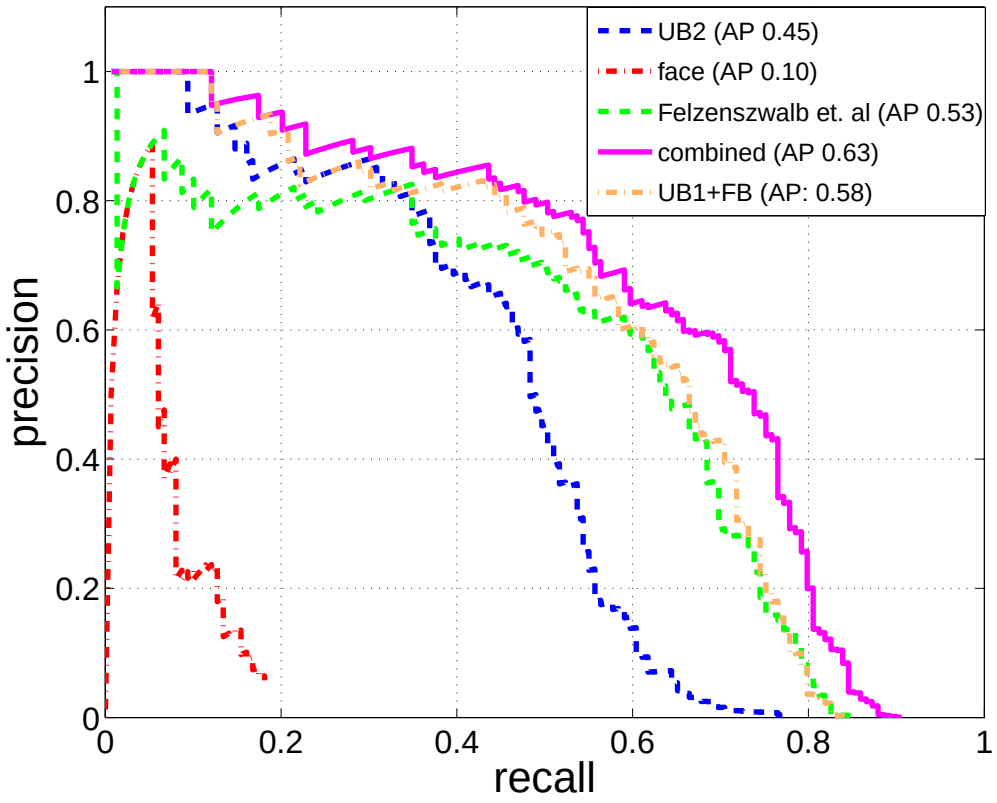

***Figure 2.4**: Precision-recall curve for the individual detectors and the combined ones. We consider a detection as correct when the Intersection-Over-Union (IoU) with a ground-truth annotation is at least 50%. In parenthesis are average precision values (AP), defined as the area under the respective curve.*

action objects have been localized in the images, we use them together with the human locations to learn probability distributions of human-object spatial relations (subsec. 2.5.4). Experimental results show that these relations are characteristic for the action, e.g. a bike is below the person riding it, whereas a hat is on top of the person wearing it (sec. 2.7). These distributions constitute our human-object interaction model.

2.5.1 The Human-Object model

Our model inputs a set of training images $\{\mathcal{I}^i\}$ showing an action (e.g. 'tennis forehand' (fig. 2.5 left) and 'croquet' (fig. 2.5 right)). We retain for each image i the single highest-scored human detection h^i, and use it as an anchor for defining the human-object spatial relations. Furthermore, for each $\mathcal{I}^i$ we have a set $\mathcal{X}^i = \{b^i_j\}$ of candidate windows po-

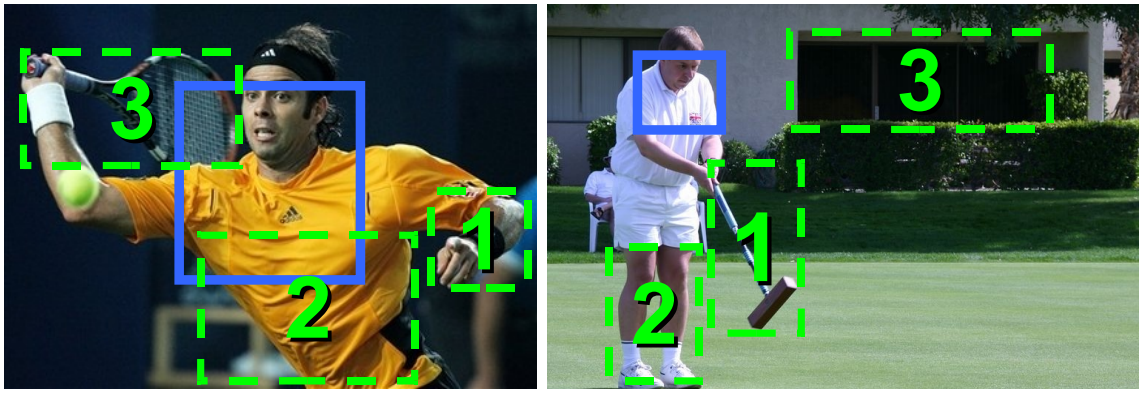

***Figure 2.5**: Two images with three candidate windows each. The blue boxes indicate the location of the human calculated by the detector. The green boxes show possible action object locations.*

tentially containing the action object (fig. 2.5). We use the generic object detector [Alexe *et al.* 2010] to select 500 windows likely to contain an objects rather than background (sec. 2.5.2).

Our goal is to select one window $b^i_j \in \mathcal{X}^i$ containing the action object for each image $\mathcal{I}^i$. We model this selection problem in energy minimization terms. Formally, the objective is to find the configuration $\mathcal{B}^*$ of windows (one window per image), so that the following energy is minimized

$$E(\mathcal{B}|\mathcal{H},\Theta) = \sum_{b^i_j \in \mathcal{B}} \Theta_U(h^i, b^i_j) + \sum_{(b^i_j, b^l_m) \in \mathcal{B}\times\mathcal{B}} \Theta_H(b^i_j, b^l_m, h^i, h^l) + \sum_{(b^i_j, b^l_m) \in \mathcal{B}\times\mathcal{B}} \Theta_P(b^i_j, b^l_m) \tag{2.3}$$

We give here a brief overview of the terms in this model, and explain them in more detail in sec. 2.5.3.

Θ_U is a sum of unary cues measuring (i) how likely a window b^i_j is to contain an object of any class ($\theta_o(b^i_j)$); (ii) the amount of overlap between the window and the human ($\theta_a(h^i, b^i_j)$)

$$\Theta_U(h^i, b^i_j) = \theta_o(b^i_j) + \theta_a(h^i, b^i_j) \tag{2.4}$$

Θ_H is a sum of pairwise cues capturing spatial relations between the human and the object. They encourage the model to select windows with similar spatial relations to the

human across images (e.g. Δ_d measures the difference in relative distance between two human-object pairs). These cues are illustrated in fig. 2.7.

$$\begin{aligned}\Theta_H(b_j^i, b_m^l, h^i, h^l) \\ =&\Delta_d(b_j^i, b_m^l, h^i, h^l) + \Delta_s(b_j^i, b_m^l, h^i, h^l) \\ +&\Delta_l(b_j^i, b_m^l, h^i, h^l) + \Delta_o(b_j^i, b_m^l, h^i, h^l)\end{aligned} \tag{2.5}$$

Finally, Θ_P is a sum of pairwise cues measuring the appearance similarity between pairs of candidate windows in different images. These cues prefer $\mathcal{B}^*$ to contain windows of similar appearance across images. They are χ^2 distances on color histograms (Δ_c) and bag-of-visual-words descriptors (Δ_i).

$$\Theta_P(b_j^i, b_m^l) = \Delta_c(b_j^i, b_m^l) + \Delta_i(b_j^i, b_m^l) \tag{2.6}$$

We normalize the range of all cues to $[0, 1]$ but do not perform any other reweighting beyond this.

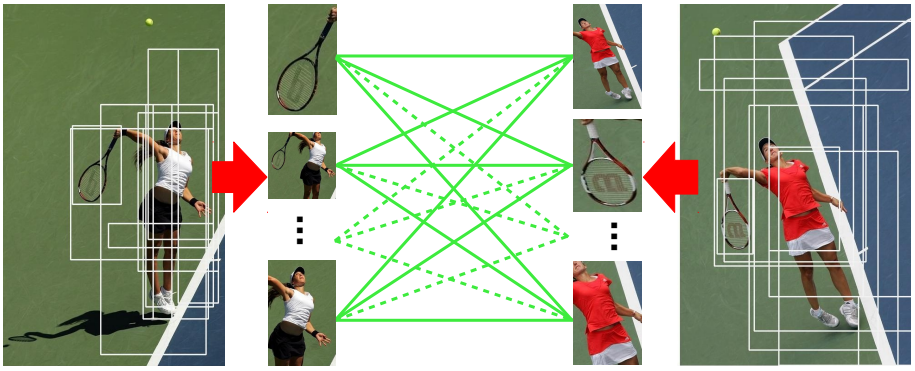

***Figure 2.6**: A pair of training images from the 'tennis serve' action. Candidate windows are depicted as white boxes. We employ a fully connected model, meaning that pairwise potentials (green lines) connect each pair of candidate windows between each pair of training images.*

As the pairwise terms connect all pairs of images, our model is fully connected. Every candidate window in an image is compared to every candidate window in another. Fig. 2.6 shows an illustration of the connectivity in our model. We perform inference on this model using the TRW-S algorithm [Kolmogorov 2006] obtaining a very good approximation of the global optimum $\mathcal{B}^* = \arg\min E(\mathcal{B}|\mathcal{H}, \Theta)$.

2.5.2 Candidate Windows

To obtain the candidate windows $\mathcal{X}$ and the unary cue θ_o we use the objectness measure of [Alexe *et al.* 2010], which quantifies how likely it is for a window to contain an object

of *any* class rather than background. Objectness is trained to distinguish windows containing an object with a well-defined boundary and center, such as cows and telephones, from amorphous background windows, such as grass and road. Objectness combines several image cues measuring distinctive characteristics of objects, such as appearing different from their surroundings, having a closed boundary, and sometimes begin unique within the image.

We use objectness as a location prior in our model, by evaluating it for all windows in an image and then sampling 500 windows according to their scores. These form the set of states for a node, i.e. the candidate windows the model can choose from.

This procedure brings two advantages. First, it greatly reduces the computational complexity of the optimization, which grows with the square of the number of windows (there are millions of windows in an image). Second, the sampled windows and their scores θ_o attract the model toward selecting objects rather than background windows.

For the experiments we used the code of [Alexe *et al.* 2010] available online [4] without any modifications or tuning. It takes only about 3 seconds to compute candidate windows for one image.

2.5.3 Cues

Unary cues.

Each candidate window b is scored separately by the unary cues θ_o and θ_a.

The cue $\theta_o(b) = -\log(p_{obj}(b))$, where $p_{obj}(b) \in [0,1]$ is the objectness probability [Alexe *et al.* 2010] of b which measures how likely b is to contain an object of any class (sec. 2.5.2).

The cue $\theta_a(h,b) = -\log(1 - \mathrm{IoU}(h^i, b^i_j))$ measures the overlap between a candidate window and the human h (with $\mathrm{IoU}(\,\cdot\,,\,\cdot\,) \in [0,1]$). It penalizes windows with a strong overlap with the human, since in most images of human-object interactions the object is near the human, but not on top of it. This cue proved to be very successful in suppressing trivial outputs such as selecting a window covering the human upper-body in every image, i.e. is the most frequently recurring pattern in human action datasets.

Human-object pairwise cues.

Candidate windows from two different images $\mathcal{I}^i, \mathcal{I}^l$ are pairwise connected as shown in fig. 2.6. Human-object pairwise cues compare two windows b^i_j, b^l_m according to different spatial layout cues. We define 4 cues measuring different spatial relations between the

[4] Source code at www.vision.ee.ethz.ch/∼calvin/software.html.

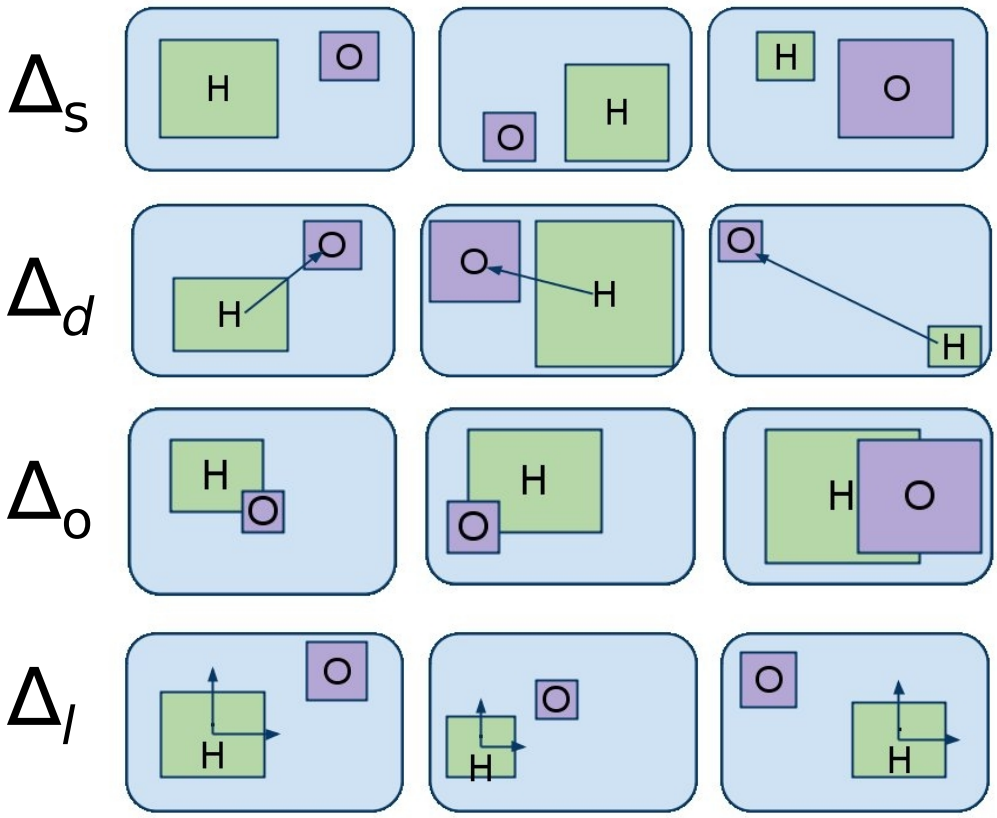

Figure 2.7*: For each human-object cue we show three possible configurations of human-object windows. The two left-most configurations have a low pairwise energy, while the right-most has a high energy compared to any of the first two.*

human and the object (fig. 2.7). These cues prefer pairs of candidate windows with a similar spatial relation to the human in their respective images. Such recurring spatial relations are characteristic for the kind of human-object interactions we are interested in (e.g. tennis serve).

Let

$$\mathrm{l}(b_j^i, h^i) = ((x_j^i - x^i)/W^i \ , \ (y_j^i - y^i)/H^i) \tag{2.7}$$

be the 2D location $\mathrm{l}(b_j^i, h^i)$ of a candidate object window $b_j^i = (x_j^i, y_j^i, W_j^i, H_j^i)$ in the reference frame defined by the human $h^i = (x^i, y^i, W^i, H^i)$ in image $\mathcal{I}^i$.

With this notation, the four cues are

1) The difference in the relative scale between the object and the human in the two images

$$\begin{aligned}\Delta_s(b_j^i, b_m^l, h^i, h^l) = \max(&\mathrm{a}(h^i, b_j^i)/\mathrm{a}(h^l, b_m^l), \\ &\mathrm{a}(h^l, b_m^l)/\mathrm{a}(h^i, b_j^i)) - 1\end{aligned} \tag{2.8}$$

where

$$\mathrm{a}(h^i, b^i_j) = \mathrm{area}(b^i_j)/\mathrm{area}(h^i) \tag{2.9}$$

is the ratio between the area (in pixels) of a candidate window and the human window.

2) The difference in the Euclidean distance between the object and the human

$$\Delta_d(b^i_j, b^l_m, h^i, h^l) = \mathrm{abs}\left(\left|\left|\mathrm{l}(b^i_j, h^i)\right|\right| - \left|\left|\mathrm{l}(b^l_m, h^l)\right|\right|\right) \tag{2.10}$$

3) The difference in the overlap area between the object and the human (normalized by the area of the human)

$$\Delta_o(b^i_j, b^l_m, h^i, h^l) = \mathrm{abs}\left(\frac{b^i_j \cap h^i}{\mathrm{area}(h^i)} - \frac{b^l_m \cap h^l}{\mathrm{area}(h^l)}\right) \tag{2.11}$$

where $a \cap b$ indicates the overlapping area (in pixel) between two windows a and b.

4) The difference in the relative location between the object and the human

$$\Delta_l(b^i_j, b^l_m, h^i, h^l) = \left|\left|\mathrm{l}(b^i_j, h^i) - \mathrm{l}(b^l_m, h^l)\right|\right| \tag{2.12}$$

Object-only pairwise cues.

The similarity $\Theta_P(b^i_j, b^l_m)$ between a pair of candidate windows b^i_j, b^l_m from two images is computed as the χ^2 difference between histograms describing their appearance. We use two descriptors. The first is a a color histogram $\Delta_c(b^i_j, b^l_m)$. The second is a bag-of-visual-words on a 3-level spatial pyramid using SURF features [Bay *et al.* 2008] $\Delta_i(b^i_j, b^l_m)$ (whose vocabulary is learnt from the positive training images and is composed of 500 visual words). These cues prefer object windows with similar appearance across images.

2.5.4 Learning Human-Object interactions

Given the human detections $\mathcal{H}$ and the object windows $\mathcal{B}^*$ minimizing equation (2.3), we learn the interactions between the human and the action object as two relative spatial distributions.

More precisely, we focus on relative location (eq. (2.7)) and relative scale (eq. (2.9)).

We estimate a 2D probability density function for the location of the object with respect to the human (eq. 2.7) as:

$$\mathrm{k}_l(\mathcal{B}^*, \mathcal{H}) = \sum_i \frac{1}{\sqrt{2\sigma}} e^{-\mathrm{l}(b^i, h^i)(1/2\sigma^2)} \tag{2.13}$$

where $b^i \in \mathcal{B}^*$ is the selected object window in image $\mathcal{I}^i$, $h^i \in \mathcal{H}$ is the reference human detection in that image, and the scale σ is set automatically by a diffusion algorithm [Botev 2007].

A second density is given by the scale of the object relative to the human (eq. (2.9)):

$$\mathrm{k}_s(\mathcal{B}^*, \mathcal{H}) = \sum_i \frac{1}{\sqrt{2\sigma}} e^{-\mathrm{a}(b^i, h^i)(1/2\sigma^2)} \tag{2.14}$$

The learnt spatial relations for various actions are presented in subsec. 2.7.4.

Additionally we train an object appearance classifier θ_t. This classifier is a SVM on a bag-of-words representation [Zhang *et al.* 2007] using dense SURF descriptors [Bay *et al.* 2008]. As positive training samples we use the selected object windows $\mathcal{B}^*$. As negative samples we use random windows from images of other action classes.

The spatial distributions k_l and k_s together with the object appearance classifier θ_t constitute the action model $\mathcal{A} = (k_l, k_s, \theta_t)$.

2.6 Action recognition

The previous section described how we automatically learn an action model from a set of training images $\{\mathcal{I}\}$. Given a test image $\mathcal{T}$ and n action models $\{\mathcal{A}^a\}_{a=1,\ldots,n}$, we want to determine which action is depicted in it.

In sections 2.6.1 to 2.6.3 we present three descriptors, each capturing a different aspect of an image. The human-object descriptor (sec. 2.6.1) exploits the spatial relations and the object appearance model in $\mathcal{A}$ (sec. 2.5) to localize the action object and then describes the human-object configuration. Sec. 2.6.2 and 2.6.3 present two descriptors capturing contextual information both at a global (sec. 2.6.2) and a local (sec. 2.6.3) level. Finally, in sec. 2.6.4, we show how we combine the different descriptors for classifying $\mathcal{T}$.

2.6.1 Human-object descriptor

We compute a low-dimensional descriptor for an image (the same procedure is applied equally to either a training or a test image): (1) detect humans and keep the highest scoring one h as anchor for computing Human-Object relations; (2) compute a set of candidate object windows $\mathcal{B}$ using [Alexe *et al.* 2010] (sec. 2.5.2); (3) for every action model $\{\mathcal{A}^a\}_{a=1,\ldots,n}$ select the window $b^a \in \mathcal{B}$ minimizing the energy

$$E(\mathcal{B}|h, \mu^a) = \theta^a_t(b) + \theta^a_{k_l}(h, b) + \theta^a_{k_s}(h, b) \tag{2.15}$$

where $\theta^a_{k_l}(h, b_j)$ and $\theta^a_{k_s}(h, b_j)$ are unary terms based on the probability distributions k_l and k_s learned during training (sec. 2.5.4); $\theta^a_t(b^i)$ is the object appearance classifier, also learned during training. The optimal window can be found efficiently as the complexity of this optimization is linear in $|\mathcal{B}|$.

For each action model μ^a we create a descriptor vector containing the energy of the three terms in eq. (2.15), evaluated for the selected window b^a. The overall human-object descriptor for the image is the concatenation over all n actions and has dimensionality $3n$. Based on this concatenated representation, the system can learn the relative merits of the various terms in the context of all actions. This is useful to adapt to correlations in the appearance and relative location of the objects between actions (e.g. if two actions involve similar relative positions of the object with respect to the human, the appearance energy will be given higher weight).

2.6.2 Whole-image descriptor

As shown by [Gupta *et al.* 2009], describing the whole image using GIST [Oliva and Torralba 2001] provides a valuable cue for action classification. This descriptor can capture the context of an action, which is often quite distinctive [Li and Fei-Fei 2007].

2.6.3 Pose-from-gradients descriptor

Both [Gupta *et al.* 2009] and [Yao and Fei-Fei 2010b] use human pose as a feature for action recognition. In those approaches pose is represented by silhouettes [Gupta *et al.* 2009] or limb locations [Yao and Fei-Fei 2010b], which are expensive to annotate manually on training images. In the same spirit of leveraging on human pose for action classification, but avoiding the additional annotation effort, we propose a much simpler descriptor to capture pose information.

Given an image and the corresponding human detection h we extract the GIST descriptor [Oliva and Torralba 2001] from an image window obtained by enlarging h by a constant factor so as to include more of the arm pose. Fig. 2.8 shows example human detections and the corresponding enlarged windows. While this descriptor does not require any additional supervision on the training images, it proved successful in discriminating difficult cases (see results in sec. 2.7.3). Moreover, it takes further advantage of using a robust human detector, such as the one in sec. 2.4.

2.6.4 Action classifiers

For training, we extract the descriptors of sections 2.6.1-2.6.3 from the same training images $\{I^i\}$ used for learning the human-object model (notice how only the action class label

Figure 2.8: *Human pose has a high discriminative power for distinguishing actions. The solid window is the original human detection, while the dashed window shows the area from which the pose-from-gradients descriptor is extracted.*

is necessary as supervision, and not human or object bounding-boxes [Gupta *et al.* 2009, Yao and Fei-Fei 2010b], human silhouettes [Gupta *et al.* 2009], or limb locations [Yao and Fei-Fei 2010b]). We obtain a separate RBF kernel for each descriptor and then compute a linear combination of them. Given the resulting combined kernel we learn a multi-class SVM. The combination weights are set by cross validation to maximize the classification accuracy [Gehler and Nowozin 2009].

Given a new test image $\mathcal{T}$, we compute the three descriptors and average the corresponding kernels according to the weights learned at training time. Finally we classify $\mathcal{T}$ (i.e. assign $\mathcal{T}$ an action label) according the multi-class SVM learned during training.

2.7 Experimental Results on the Sports and TBH datasets

We present here action recognition results on two datasets: the 6 sports actions of [Gupta *et al.* 2009], and a new dataset of 3 actions we collected, called the *Trumpets, Bikes and Hats* (TBH) dataset. Section 2.7.1 describes the datasets. Section 2.7.2 presents the experimental setup, namely the two levels of supervision we evaluate on. Section 2.7.3 reports quantitative results and comparisons to [Gupta *et al.* 2009] and [Yao and Fei-Fei 2010b]. The learned human-object interactions are illustrated in sec. 2.7.4.

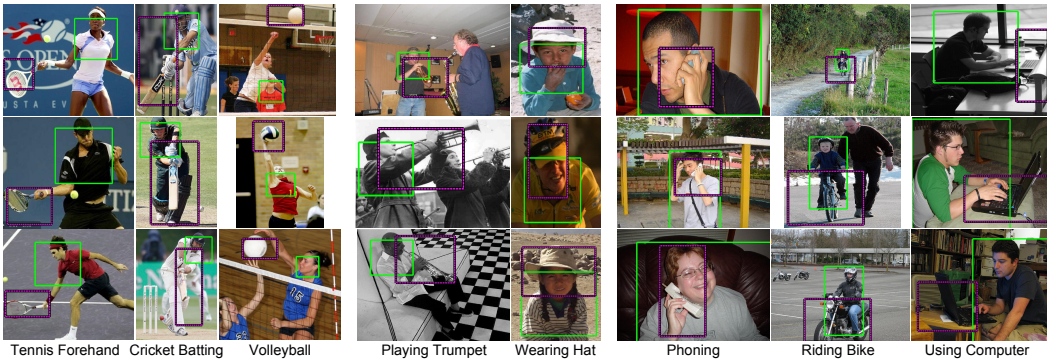

Figure 2.9: *Columns 1-5: example of action-object windows localized by our method in weakly supervised training images of the Sports dataset [Gupta* et al. *2009] (columns 1-3) and of the TBH dataset (columns 4-5). Both the human (green) and the object (dashed pink) are found automatically. Each column shows* 3 *images from the same class. The method is able to handle multi-modal human-object spatial configurations. Columns 6-8: action-object windows automatically selected from images of the PASCAL Action 2010 dataset [Everingham and others 2010]. The human window is localized manually in all images of the dataset, see the PASCAL protocol. The object is localized automatically by our method.*

2.7.1 Datasets

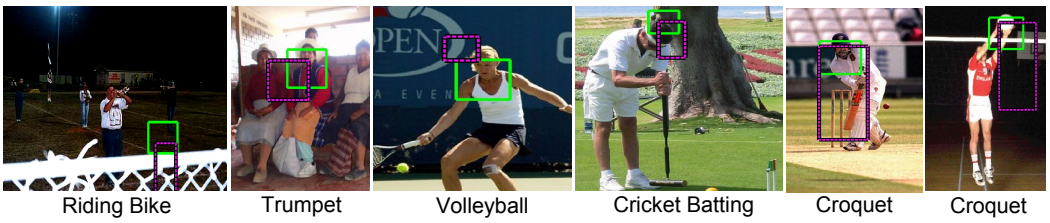

Figure 2.12: *Example for failures of our method on several test images (after training in the WS setting). Action labels indicate the (incorrect) classes the images were assigned to. The main reasons are: missed humans due to unusual pose or poor visibility (first, fourth and sixth image), similarities between different action classes (fifth image), truncation or poor visibility of the action object (second and third image).*

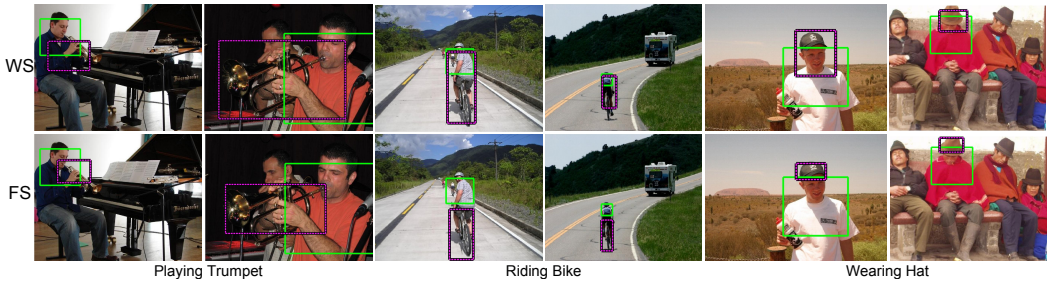

Figure 2.10: *Example results on the TBH dataset for test images that were correctly classified by our approach. Two images are shown for each action class (from left to right, 'playing trumpet', 'riding bike' and 'wearing hat'). First row: results from the weakly supervised setting WS. Second row: results from the fully supervised setting FS.*

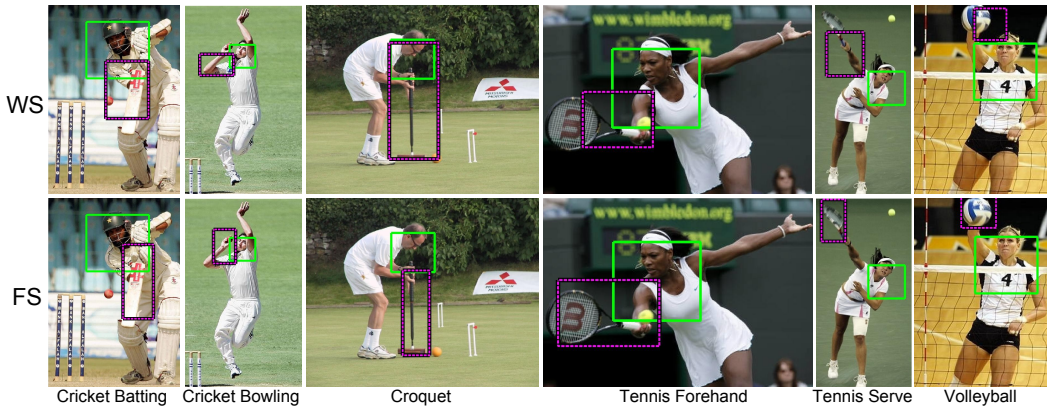

Figure 2.11: *Example results from the sports dataset of [Gupta* et al. *2009] for test images that were correctly classified by our approach. One image per class is shown (from left to right: 'cricket batting', 'cricket bowling', 'croquet', 'tennis forehand', 'tennis serve' and 'volleyball'). First row: weakly supervised setting setting. Second row: fully supervised setting.*

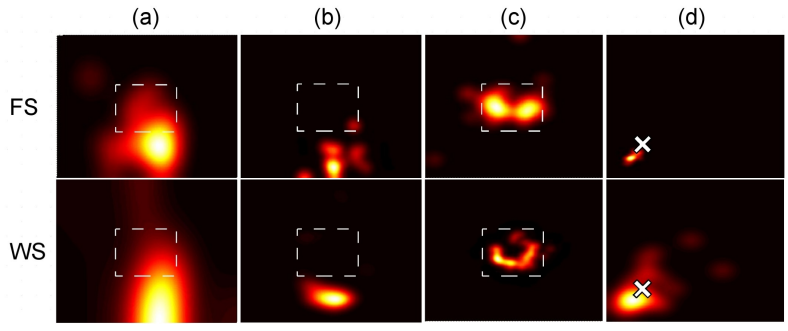

Figure 2.13: *Human-object spatial distributions learned in the FS setting (top) and in the WS setting (bottom). (a)-(c): relative location of the action object with respect to the human (k_l in sec. 2.5.4). Dashed boxes indicate the size and location of the human windows. (a) 'Cricket Batting', (b) 'Croquet', (c) 'Playing Trumpet'. (d): distribution of the object scale relative to the human scale for the action 'Volleyball' (k_s in sec. 2.5.4). The horizontal axis represents the x-scale and the vertical the y-scale. A cross indicates the scale of the human.*

***Table 2.1**: Classification results on the TBH human action dataset: (first row) our method with weak supervision, (second row) our method with full supervision, (other rows) variants of our approach. See text for details.*

	Pose from Gradients	Object appearance classifier	Whole-scene	Pose from Grad. + Whole-scene + Obj. appear. class.	Full model
Ours WS	**54**	**53**	**58**	**71**	**74**
Ours FS	**58**	**61**	**58**	**74**	**79**
Ours WS-HumanPFF	**45**	**51**	**58**	**66**	**69**
Ours WS-HumanGT	**66**	**54**	**58**	**74**	**75**
Ours WS-AltCands	**54**	**54**	**58**	**71**	**71**

TBH dataset

We introduce a new action dataset called TBH. It is built from Google Images and the IAPR TC-12 dataset [Grubinger *et al.* 2006], and contains 3 actions: 'playing trumpet', 'riding bike', and 'wearing hat'.

We use Google Images to retrieve images for the action 'playing trumpet'. We manually select the first 100 images depicting the action in a set of images obtained by searching for "person OR man OR woman", followed by the action verb ("playing") and the object name ("trumpet"). The amount of negative images that have been manually discarded has been 25%. We split these 100 positive images into training (60) and testing (40), i.e. the same proportions as the sports dataset [Gupta *et al.* 2009].

For the actions 'riding bike' and 'wearing hat' we collected images from the IAPR TC-12 dataset. Each image in this large dataset has an accompanying text caption describing the image. We run a natural language processor (NLP) [Johansson and Nugues 2008] on the text captions to retrieve images showing the action. In detail, a caption should contain: (i) a subject, specified as either 'person', 'man', 'woman', or 'boy'; (ii) a verb-object pair. The verb is specified in the infinitive form, while the object as a set of synonyms (e.g 'hat' and 'cap'). Due to the high quality of the captions, this process returns almost only relevant images. We manually removed just 1 irrelevant image from each class. The resulting dataset contains 117 images for 'riding bike' (70 training, 47 testing) and 124 images for 'wearing hat' (74 training, 50 testing). In the resulting TBH dataset, images are only annotated by label of the action class they depict.

Sports dataset [Gupta *et al.* 2009]

This dataset is composed of 6 actions of people doing sports. These actions are: 'cricket batting', 'cricket bowling', 'croquet', 'tennis forehand', 'tennis backhand' and 'volleyball

***Table 2.2**: Classification results on the PASCAL Action 2010 dataset: We show average precision results for individual classes. In the last column we show results from the best contestant in the challenge, Koniusz et al. [Everingham and others 2010]. Each entry in the first 9 rows is the average precision of one class, while the last two rows present mean average precision over several classes. Column 'Full model' include our Human-Object spatial relations (i.e. the interaction model).*

	Pose from Gradients	Object appearance classifier	Whole-scene	Pose from Grad. + Whole-scene + Obj. appear. class.	Full model	Koniusz et al.
Phoning	**21**	**23**	**18**	**39**	**55**	**53**
Playing instrument	**19**	**16**	**29**	**32**	**36**	**54**
Reading	**19**	**45**	**12**	**64**	**69**	**36**
Riding bike	**46**	**42**	**22**	**55**	**71**	**81**
Riding horse	**46**	**43**	**39**	**53**	**50**	**89**
Taking photo	**15**	**91**	**18**	**88**	**90**	**33**
Using computer	**32**	**57**	**29**	**69**	**81**	**59**
Running	**51**	**45**	**35**	**56**	**59**	**87**
Walking	**33**	**30**	**34**	**41**	**44**	**69**
mAP all classes	**31**	**43**	**26**	**55**	**62**	**62**
mAP Human-Object classes	**28**	**45**	**24**	**57**	**65**	**58**

smash'. Each action has 30 training images and 20 test images. These images come with a rich set of annotations. The approaches of [Gupta *et al.* 2009] and [Yao and Fei-Fei 2010b] are in fact trained with full supervision, using all these annotations. More precisely, for each training image they need:

A1 action label

A2 ground-truth bounding-box for the action object

A3 manually segmented human silhouette [Gupta *et al.* 2009] or limb locations [Yao and Fei-Fei 2010b].

A4 [Gupta *et al.* 2009] also requires a set of training images for each action object, collected from Google Images (e.g. by querying for 'tennis racket' and then manually discarding irrelevant images).

2.7.2 Experimental setups

Weakly supervised (WS).

Our method learns human actions from images labeled only with the action they contain (A1), i.e. weakly supervised images (WS).

At training time we localize objects in the training set by applying the model presented in sec. 2.5 (fig. 2.9). Given the localized objects and the humans locations we learn spatial relations as well as an object appearance classifier (sec. 2.5.4).

At test time we recognize human actions in test images by applying the procedure described in sec. 2.6.

Fully supervised (FS).

In order to fairly compare our approach with [Gupta *et al.* 2009] and [Yao and Fei-Fei 2010b], we introduce a fully supervised variant of our model, where we use A1 and A2. Instead of A3 we just use ground-truth bounding-boxes on the human, which is less supervision than silhouettes [Gupta *et al.* 2009] or limb locations [Yao and Fei-Fei 2010b]. It is then straightforward to learn the human-object relation models and the object appearance classifier (sec. 2.5.4) from these ground-truth bounding-boxes. We also train a sliding-window detector [Felzenszwalb *et al.* 2009] for the action object using the ground-truth bounding-boxes A2. This detector then gives the appearance cue θ_t in eq. 2.15.

In the following we denote with FS our fully supervised setting using one human bounding-box and one object bounding box per training image. Instead, we denote by FS** the setting using A1-A3 [Yao and Fei-Fei 2010b] and FS* the setting using A1-A4 [Gupta *et al.* 2009].

In the FS setup, we recognize human actions in test images by applying the procedure described in 2.6. In step (2) of sec. 2.6.1 we run the action object detector to obtain candidate windows $\mathcal{B}$, i.e. all windows returned by the detector, without applying any threshold nor non-maxima suppression.

2.7.3 Experimental evaluation

***Table 2.3**: Classification results on the sports dataset [Gupta* et al. *2009]: 1st row: our method with WS; 2nd row: our method with FS; 3rd row: [Gupta* et al. *2009] with FS*; 4th row: [Yao and Fei-Fei 2010b] with FS** (they only report results for their full model). Each entry is the classification accuracy averaged over all 6 classes. Column 'Full model' in rows 1 and 2 includes our Human-Object spatial relations.*

	Human pose	Pose from Gradients	Object appearance classifier	Whole-scene	Pose from Grad. + Whole-scene + Obj. appear. class.	Full model
Ours WS	-	**54**	**32**	**67**	**76**	**81**
Ours FS	-	**58**	**46**	**67**	**80**	**83**
[Gupta *et al.* 2009] FS*	**58**	-	-	**66**	-	**79**
[Yao and Fei-Fei 2010b] FS**	-	-	-	-	-	**83**

Sports dataset [Gupta *et al.* 2009]

Table 2.3 presents results on the sports dataset [Gupta *et al.* 2009], where the task is to classify each test image into one of six actions. In the WS setup (first row), combining the object appearance classifier (sec. 2.5.4), the pose-from-gradients descriptor and the whole-image classifier improves over using any of them alone and already obtains good performance (76%). Importantly, adding the human-object interaction model ('Full model' column) raises performance to 81%, confirming that our model learns human-object spatial relations beneficial for action classification. Fig. 2.10 and fig. 2.11 show humans and objects automatically detected on the test images by our full method.An important point is that the performance of our model trained in the WS setup is 2% better than the FS* approach of [Gupta *et al.* 2009] and 2% below the FS** approach of [Yao and Fei-Fei 2010b]. This confirms the main claim of the chapter: our method can effectively learn actions defined by human-object interactions in a WS setting. Remarkably, it reaches performance comparable to state-of-the-art methods in FS settings which are very expensive in terms of training annotation.

The second row of table 2.3 shows results for our method in the FS setup. As expected, the object appearance classifier performs better than the WS one, as we can train it from ground-truth bounding-boxes. Again the combination with the pose-from-gradients descriptor and the whole-scene classifier significantly improves results (now to 80%). Furthermore, also in this FS setup adding the human-object spatial relations raises performance ('Full model'). The classification accuracy exceeds that of [Gupta *et al.* 2009] and is on par with [Yao and Fei-Fei 2010b]. We note how [Yao and Fei-Fei 2010b, Gupta *et al.* 2009] use human body part locations or silhouettes for training, while we use only human bounding-boxes, which are cheaper to obtain. Interestingly, although trained with much less supervision, our pose-from-gradients descriptor performs on par with the human pose descriptor of [Gupta *et al.* 2009].

TBH dataset

Table 2.1 shows results on the TBH dataset, which reinforce the conclusions drawn on the sports dataset: (i) combining the object appearance classifier, pose-from-gradients and whole-scene classifier is beneficial in both WS and FS setups; (ii) the human-object interaction model brings further improvements in both setups; (iii) the performance of the full model in the WS setup is only 5% below that of the FS setup, confirming our method is a good solution for WS learning.

We note that the performance gap of the object appearance classifier between FS and WS is smaller than on the sports dataset. This might be due to the greater difference between action objects in the TBH dataset, where a weaker object model already works well.

Finally, we note how the whole-scene descriptor has lower discriminative power than on the sports dataset (67% across 6 classes vs. 58% across 3 classes). This is likely due to the greater intra-class variability of backgrounds and scenes in the TBH dataset. Fig. 2.10 and 2.11 show example results for automatically localized action objects on the test data from the two datasets. While in the FS setup our method localizes the action objects more accurately, in many cases it detects it already well in the WS setup, in spite of being trained without any bounding-box. Failure cases are shown and discussed in fig. 2.12.

Influence of the human detector

To demonstrate the influence of our human detector (sec. 2.4) on action classification results, we evaluate two variants of our WS setup which use alternative ways to select a human reference frame (both at training and test time). The first variant (WS-HumanPFF) uses the highest-scoring human detection returned by [Felzenszwalb *et al.* 2009]. The second variant (WS-HumanGT) uses the ground-truth human annotation as the reference frame. We report in table 2.1 results on the TBH dataset, which has a high variability of human poses and viewpoints.

The difference between rows 'WS' and 'WS-HumanPFF' demonstrates that using our detector (sec. 2.4) results in significantly better action recognition performance over using [Felzenszwalb *et al.* 2009] alone (+5%). Interestingly, using our human detector leads to performance close to using ground-truth detections (row 'WS-HumanGT') (-1%).

Influence of the choice of candidate windows

In all experiments so far, we have used the objectness measure of [Alexe *et al.* 2010] to automatically propose a set of candidate windows $\mathcal{X}^i$, from which our algorithm chooses the most consistent solution over a set of training images (sec. 2.5.1). To show the impact of the objectness measure, we compare to a simple baseline based on the intuition that image patches close to the human are more likely to contain the action object. This baseline samples arbitrary windows overlapping with the human detection h^i. More precisely, for each training image i we randomly sample 10^6 windows uniformly and score each window w with $s = 1 - \mathrm{abs}(0.5 - \mathrm{IoU}(\mathrm{w}, \mathrm{h^i}))/0.5$. This score is highest for windows that overlap about 50% with the human, and lowest for windows either completely on top of it or not overlapping with it at all (i.e. background). This is a good criterion, as the action object is typically hold in the human's hand, and so it partially overlaps with it. To form the set of candidate windows, we random sample 500 windows according to s.

We report in table 2.1 results on the TBH dataset (WS-AltCands). This alternative strategy for sampling candidate windows leads to moderately worse action recognition results than

Figure 2.14: *Example results on the PASCAL Action 2010 test set [Everingham and others 2010]. Each column shows two images from the validation set for the same class. From left to right: 'playing instrument', 'reading', 'taking photo', 'riding horse' and 'walking'.*

when using objectness windows (-3%). Moreover, it is interesting to note how the spatial-relations learned based on the alternative windows are weaker as they do not bring a positive contribution when combined in the full model (cf. 4th and 5th columns).

2.7.4 Learned human-object interactions

Fig. 2.13 compares human-object spatial relations obtained from automatically localized humans and objects in the WS setup to those derived from ground-truth bounding-boxes in the FS setup (sec. 2.5.4). The learnt relations are clearly meaningful. The location of the Cricket Bat (first column) is near the chest of the person, whereas the croquet mallet (second column) is below the torso. Trumpets are distributed near the center of the human reference frame, as they are often played at the mouth (third column). As the fourth column shows, the relative scale between the human and the object for the 'Volleyball' action indicates that a volley ball is about half the size of a human detection (see also rightmost column of fig. 2.11).

Importantly, the spatial relations learned in the WS setting are similar to those learnt in the FS setting, albeit less peaked. This demonstrates that our weakly supervised approach does learn correctly human-object interactions.

2.8 Experimental Results on the PASCAL Action 2010 dataset

Dataset and protocol

The PASCAL Action [Everingham and others 2010] dataset contains 9 action classes, 7 of which involve a human and an object: 'phoning', 'playing instrument', 'reading', 'riding bike', 'riding horse', 'taking photo' and 'using computer'. The two actions involving no object are 'running' and 'walking'. Each class has between 50 and 60 images divided equally into training and testing [5] subsets. Each image is annotated with ground-truth bounding-boxes on the humans performing the action (there might be more than one). For images with multiple human annotations, we duplicate the image and assign each human to a different image. In this way we maintain our method unchanged while also making our results fully comparable with previous work.

We perform experiments following the official protocol of the PASCAL Challenge [Everingham and others 2010], where human ground-truth bounding-boxes are given to the algorithm both at training and at test time. We train a separate 1-vs-all action classifier with our method from sec. 2.5 and 2.6, using the ground-truth human annotations as $\mathcal{H}$. However, object locations are not given and they are automatically found by our method (fig. 2.9).

At test time we evaluate the classification accuracy for each action separately by computing a precision-recall curve. This means that each action classifier is applied to all annotated humans in the test images from all classes, and the resulting confidence values are used to compute the precision-recall curve. We report *average precision*, i.e. the area under the precision-recall curve, which is the official measure of the PASCAL Challenge [Everingham and others 2010].

Experimental evaluation

The first 9 rows of table 2.2 show the average precision for each of the 9 actions. We present the mean Average Precision over classes (mAP) in the last two rows. Fig. 2.14 shows results on example test images. Note how the object appearance classifier and human-object interaction components of our model are trained in a weakly supervised manner, as the location of the action object is not given (neither at training nor at test time). The results demonstrate that these components improve the performance of our method compared to using information on the human alone ('Pose from gradients' column). Also note how the whole-scene classifier is only moderately informative on this

[5] Since the complete annotations for the test set were not available at the time of submission, we tested on the validation set instead

dataset, leaving most of the contribution to the overall performance to the object and interaction components ('Full model').

Our full model achieves a 5% improvement compared to the best method in the challenge, i.e. Koniusz et al. [Everingham and others 2010], when averaged on the 7 classes involving both humans and objects (last row of table 2.2). Moreover, when considering all classes it performs on par with it (second last row). As the 'running' and 'walking' rows show, our method can also handle classes involving no object, delivering good performance even though it was not designed for this purpose. Surprisingly, for the 'walking' class, the human-object spatial relations bring a strong improvement. The reason is that our method selects images patches on the legs as the "action object", as they are a recurring pattern which is distinctive for walking (last column of fig. 2.14).

2.9 Experimental Results from the PASCAL Action 2011 challenge

Our human-object action model also participated in the Pascal Action Classification Challenge 2011 [Everingham and others 2010]. More in detail, we combined our human-object action model (sec. 2.6.1) with a powerful approach based on low-level features [Sharma *et al.* 2012]. The method of [Sharma *et al.* 2012], dubbed DSAL, learns discriminative saliency maps for images, highlighting the regions which are more discriminant for a certain classification task. They use the saliency maps to weight the visual words for improving the discriminative capacity of bag of words features. Their approach is motivated by the observation that for many human actions and attributes, local regions are highly discriminative e.g. for running the bent arms and legs are highly discriminant. Along with that they also combine features based on SIFT, HOG, Color and texture.

Table 2.4: *Action Classification Results of the Pascal Challenge 2011: we compare the original method of [Sharma* et al. *2012] (DSAL), its combination with our human-action model (sec. 2.6.1) and the result from the best contestant.*

	jump	phone	play instrument	read	ride bike	ride horse	run	take photo	use computer	walk	AVG
DSAL	**62**	**40**	**61**	**34**	**81**	**84**	**80**	**23**	**53**	**50**	**57**
DSAL+ours	**72**	**51**	**78**	**38**	**87**	**90**	**84**	**25**	**59**	**59**	**64**
Stanford	**66**	**41**	**60**	**42**	**90**	**92**	**87**	**29**	**62**	**66**	**63**

In table 2.4 we show the classification results for [Sharma *et al.* 2012] alone (DSAL), its combination with our human-action model (DSAL+ours) and the best contestant, which is a method based on [Yao and Fei-Fei 2010a]. Note that the challenge follows the exact

same protocol described in the previous section. Our human-action model together with the low-level features of [Sharma *et al.* 2012] obtains the highest average accuracy. This result confirms once more that the combination of our high-level human-object model together with discriminative low-level features is very powerful. As seen in the previous section, this holds true also for actions that do not involve an interaction with an object (i.e. walk, run).

3

Explicit modeling of human-object interactions in realistic videos

3.1 Introduction

The previous chapter modeled human-object interactions in *still images*. Here we extend the idea to video, where we propose an approach that has an explicit notion of the action object and represents an action by spatio-temporal descriptors dedicated to human-object interactions in video. In this chapter we go considerably beyond chapter 2 by (i) modeling and learning the spatio-temporal dynamics of interactions in *videos*, and (ii) evaluating action *localization* as opposed to mere *classification*.

Our approach is based on features such as the relative motion of the object with respect to the human, which is typically highly distinctive for the action. Measuring these features involves automatically localizing the human and the action object and tracking them over time as shown in fig. 3.1. Our method is especially designed to do this in realistic videos, such as feature films. It does not involve any component that depends on background subtraction, which makes it suitable for any camera and background motion. Moreover, the method builds on state-of-the-art object detection techniques [Felzenszwalb *et al.* 2009, Prest *et al.* 2011] operating in single frames, and robustly links detections over time even across many frames where the object was missed, again using a state-of-the-art approach [Sundaram *et al.* 2010]. Finally, our technique takes advantage of the temporal continuity in video to reduce the amount of supervision needed to learn an appearance model of the action object as well as the interaction model. As a result, it can be trained with a modest amount of annotation: for each video clip of the action class we only need a spatio-temporal cuboid on the person and a bounding-box on the action object in *one* frame.

We evaluate our method on the highly challenging task of spatio-temporal action localization on the *Coffee & Cigarettes* dataset [Laptev and Perez 2007], and on the simpler task of action classification on the datasets of [Gupta *et al.* 2009] and of [Messing *et al.*

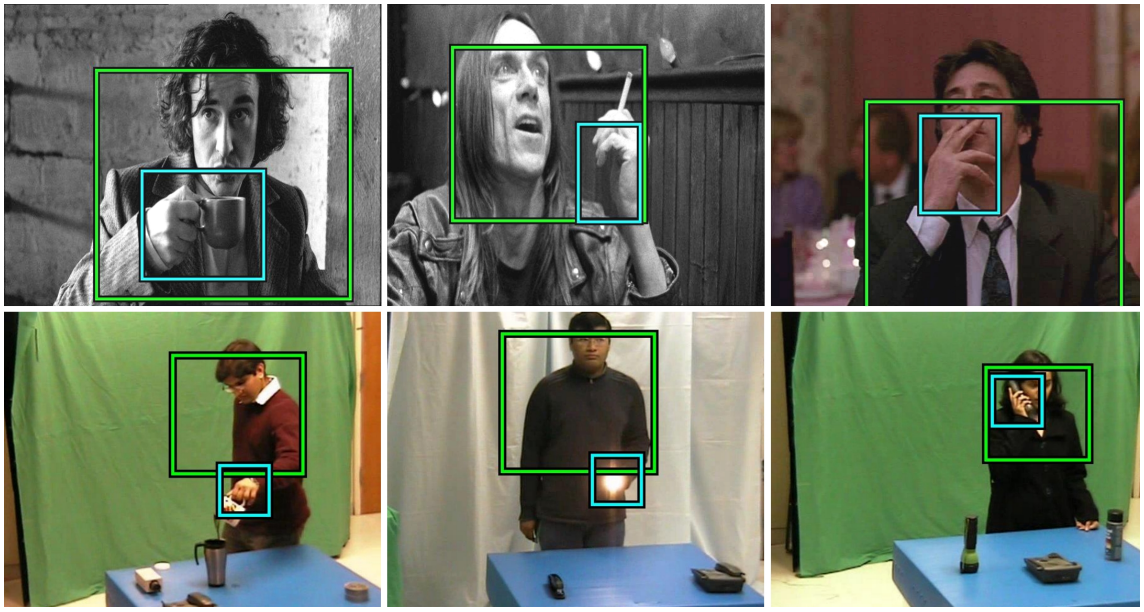

Figure 3.1: **Human-object interactions**. *Top row: one drinking and two smoking instances from the Coffee & Cigarettes dataset [Laptev and Perez 2007]. Second row: examples from the dataset of [Gupta* et al. *2009]. Human and object locations automatically produced by our method are indicated in green and cyan respectively.*

2009]. Our experiments demonstrate that (i) our human-object interaction model enables action localization and classification already on its own (sec. 3.7); (ii) it captures information complementary to existing low-level descriptors such as 3D-HOG computed over human tracks [Kläser *et al.* 2010]. Their combination performs better than either alone and improves over the state-of-the-art on *Coffee & Cigarettes* [Kläser *et al.* 2010]; (iii) our approach matches the performance of Gupta et al. [Gupta *et al.* 2009] on their dataset while using less supervision for training. In the rest of the chapter we refer to this dataset as the Gupta video dataset; (iv) our approach outperforms the recent work of [Matikainen *et al.* 2010, Messing *et al.* 2009] on the Rochester Daily Activities dataset.

The rest of the chapter is organized as follows. Sec. 3.3 first gives an overview of our method, and then sections 3.4 to 3.6 explain its components in detail. Sec. 3.4 explains our algorithm to robustly detect and track humans and objects in realistic videos. For this we employ state-of-the-art methods for detecting humans [Prest *et al.* 2011] and objects [Felzenszwalb *et al.* 2009], as well as for tracking them over time [Sundaram *et al.* 2010]. This is a necessary step towards our human-object interaction model, which is the main contribution of this chapter (sec. 3.5). In sec. 3.6 we build a complete action recognition classifier by combining our interaction model with traditional low-level cues, and finally present experiments in sec. 3.7.

3.2 Related work

Many existing approaches for action recognition rely on simple measurements such as optical flow or spatio-temporal gradients extracted from video clips. An example are the popular bags of spatio-temporal features, initially introduced in [Dollar *et al.* 2005, Schuldt *et al.* 2004, Zelnik-Manor and Irani 2001]. These techniques extract spatio-temporal features over video clips, quantize them and use a frequency histogram to represent the clips. Recent extensions model the temporal structure of actions as a composition of smaller sub-parts [Gaidon *et al.* 2011, Laptev *et al.* 2008, Niebles *et al.* 2010]. Furthermore, they determine the temporal extent of video clips optimal for a bag-of-features representation in realistic movies [Duchenne *et al.* 2009, Satkin and Hebert 2010].

Another line of work describes the human tracks based on low-level features such as optical flow [Efros *et al.* 2003] or based on the silhouette of the humans [Bobick and Davis 2001, Yilmaz and Shah 2005, Gorelick *et al.* 2007]. Specifically, [Yilmaz and Shah 2005, Gorelick *et al.* 2007] propose human-centered approaches for action recognition based on spatio-temporal volumes (STV) obtained by accumulating silhouette information over time. They then extract information such as speed, direction and shape to characterize the STV. In [Bobick and Davis 2001] they extract silhouettes from a single view and aggregate differences between subsequent frames of an action sequence resulting in a binary motion energy image. Temporal information is included through a motion history image. The method proposed in [Efros *et al.* 2003] operates on sports footage. They compensate camera movement by tracking the person and calculate optical flow in person-centered tracks.

In [Wu *et al.* 2010] a method based on particle filtering is used for modeling crowd flow and detect anomalies.

All of the above mentioned human-centric approaches operate either with static cameras, i.e., human can be located based on background subtraction, or with simple backgrounds from which human can be extracted easily, as for example football or ice hockey fields. More recent human-centric approaches [Laptev and Perez 2007, Mikolajczyk and Uemura 2008, Kläser *et al.* 2010, Rodriguez *et al.* 2008] deal with action localization in realistic video. Laptev and Perez [Laptev and Perez 2007] aggregate local spatio-temporal features over time into a spatio-temporal grid. They use keyframe priming to refine the output of their method. In [Mikolajczyk and Uemura 2008] authors also adopt a human-centric approach where vocabularies of local motion and shape features are combined with a voting approach. Liu et al. [Liu *et al.* 2009] propose a combination of static and motion low-level features and efficient techniques for mining the most discriminative ones in realistic youtube videos. The method proposed in [Kläser *et al.* 2010] localizes actions in space and time by first extracting human tracks and then detecting specific actions within the tracks using a sliding window classifier. Actions are described by track-aligned 3D-HOG features. These features are shown to be complementary to our human-object interaction descriptors and are incorporated in our final classifier.

The weakly-supervised approaches by Ikizler et al. [Ikizler and Forsyth 2008, Ikizler-Cinbis *et al.* 2009] attempt to decrease the amount of supervision necessary for training action classifiers. Training videos for learning actions are obtained inexpensively from YouTube [Ikizler-Cinbis *et al.* 2009]. Their approach is robust to the low-quality video as well as complex scenes necessary for such video material.

Several works tackle the problem of recognizing human-object interactions in video [Filipovych and Ribeiro 2008, Filipovych and Ribeiro 2010, Gupta *et al.* 2009, Matikainen *et al.* 2010, Messing *et al.* 2009]. Messing et al. [Messing *et al.* 2009] introduce a dataset of human-object interactions recorded in controlled conditions and propose a descriptor based on the velocity history of tracked point features. Matikainen et al. [Matikainen *et al.* 2010] extends this descriptor to include relations between pairs of tracked points and quantize them into vocabularies. In contrast with our work, both these approaches are based on low-level features to describe actions. The work most closely related to ours is by Gupta et al. [Gupta *et al.* 2009]. They model the action object and the human-object motion for classifying interactions between humans and objects. However, the motion features used in their approach are more fragile: they rely on hand trajectories to model how objects are reached and grasped. In particular, the velocity profile of the reaching hand and the time interval between a reach and a grasp motion proved to be powerful features in their experiments. Nevertheless, these fine-grained features rely on motion extracted based on background subtraction, which limits its applicability to static cameras and backgrounds (as opposed to uncontrolled video such as feature films). Moreover, [Gupta *et al.* 2009] requires substantial annotation effort for training, including the location of the person, of its hands, and a pixelwise segmentation of the action object in all video frames. Filipovych and Ribeiro [Filipovych and Ribeiro 2008, Filipovych and Ribeiro 2010] model human-object interactions based on the trajectory and appearance of spatio-temporal interest points. Their approach is demonstrated in controlled videos taken by a static camera against a static, uniform background. Importantly, the scene is seen from the actor's viewpoint. This is substantially different from the type of video we consider.

3.3 Overview of our method

In this section we present an overview of our approach to action recognition, based on explicitly modeling the human-object interaction (fig. 3.2). We summarize the stages of the pipelines for training the model for an action class (sec. 3.3.1) and for localizing it in space and time in a novel test video (sec. 3.3.2).

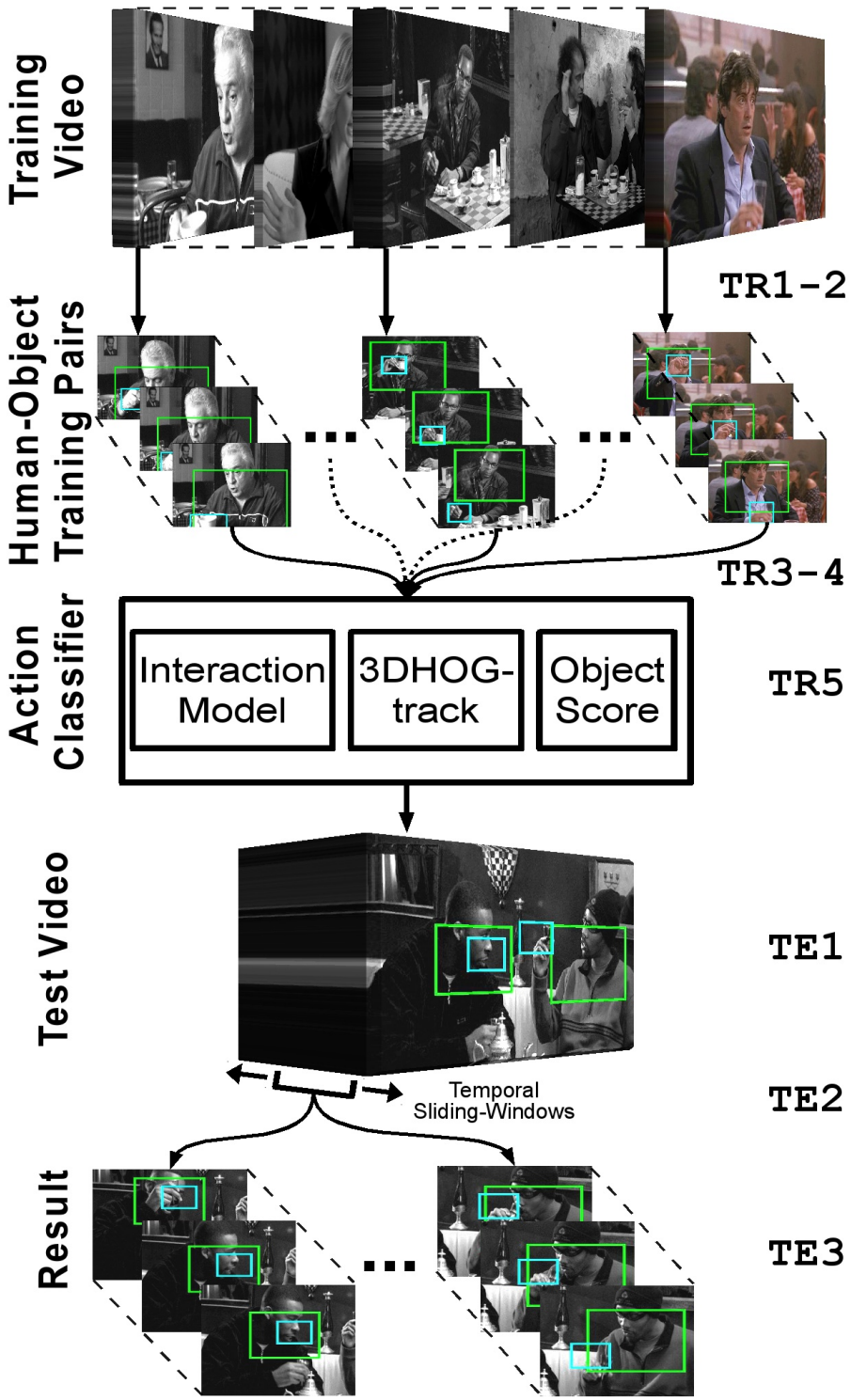

***Figure 3.2*: Overview of our method.** *We show the training pipeline (*$\mathrm{TR}1-5$*) and the test pipeline (*$\mathrm{TE}1-3$*). See text for details.*

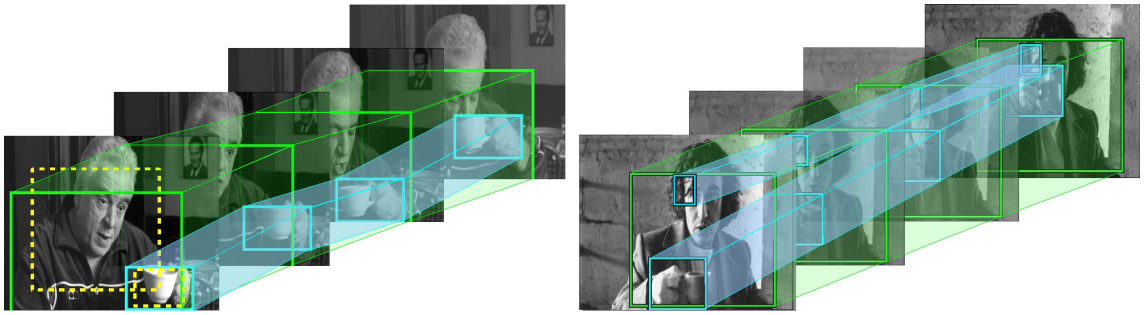

***Figure 3.3*: Tracking at training and test time.** *(Left) The training stage* TR1 *tracks the annotated bounding-boxes (dashed yellow) throughout the temporal extent of the annotation cuboid (persons in green and objects in cyan). (Right) The test stage* TE1 *detects both humans and objects automatically and tracks them throughout the video. For illustration we show here only two object tracks, out of many more (a positive one covering the cup, and a negative one on the actor's face).*

3.3.1 Training

Input. In order to train the model for an action class, our method takes as input: (i) a long video including instances of the action class; (ii) spatio-temporal cuboids, constant in the spatial dimension. Each annotation cuboid defines the location in time and space of a human performing an instance of the action class; (iii) for each annotation cuboid, the location of the action-object is annotated in *one frame* within the temporal extent of the cuboid. In the following we describe each step of the training (TR) procedure, marked as $\mathrm{TR1}-5$.

TR1. We localize and track the humans in the training video. We first apply the human detector of [Prest *et al.* 2011] independently on each frame and then link the resulting detections over time into tracks (sec. 3.4). For each annotation cuboid, we select the track which best overlaps with it and cut it to the precise temporal extent. This results in our set of *positive human tracks*. There is exactly one such track for each cuboid.

As the overall goal of our work is to learn the relative motion between humans and objects that is characteristic for the action class, we also need to track action-objects. For each annotation cuboid, we track the object starting from the single annotated frame forward and backward in time until either end of the temporal extent of the cuboid (sec. 3.4). These form the *positive object tracks* (fig. 3.3, left). Again, there is exactly one such track for each cuboid. For each cuboid we now associate its human and object track into a *positive human-object pair*.

TR2. We use the object windows in all frames of all positive object tracks as positive samples for training an action-object detector using the recent method of [Felzenszwalb *et al.* 2009] (sec. 3.4.1).

TR3. We use the detector from TR2 on the negative parts of the training video (i.e. parts not overlapping in time with any cuboid), and then run our tracker to link the resulting detections over time, obtaining *negative object tracks*. These are valuable 'hard negatives'. We now form *negative human-object pairs* by associating human and object tracks detected in the negative part, which are close in space and time (sec. 3.5.2)

TR4. For each human-object pair we compute an interaction descriptor capturing the relative location, relative area and relative motion of the object wrt the human (sec. 3.5.1). Moreover, we also compute the low-level 3DHOG-track descriptors [Kläser *et al.* 2010] for each human track in a pair. As a third descriptor, we use the score of the object detector trained in TR2 on the object track in a pair (sec. 3.6).

TR5. We use the descriptors from positive and negative human-object pairs to train a discriminative action classifier (sec. 3.6).

3.3.2 Testing

Input. Given an input test video we localize the action class in space and time. Note the complexity of the task: we localize a short action in a full length movie.

TE1. We compute human tracks on the test video with the same technique as in TR1 (sec. 3.4). However, as we now have no cuboid annotations, we retain all human tracks for the later stages. We also compute candidate object tracks by first running the single-frame action-object detector learned in TR2, and then running our tracker to link the resulting detections over time (fig. 3.3 right). We then associate human and object tracks into human-object pairs (sec. 3.5.2).

TE2. These raw pairs are unlikely to precisely cover the temporal extent of the action. In order to obtain an appropriate temporal extent of the action, we use a multi-scale temporal sliding window to produce multiple *candidates* with different temporal extents for each test pair (sec. 3.5.3).

TE3. For each candidate pair, we compute the three descriptors as in TR4 and score it with the action classifier trained in TR5. As a last step, we suppress multiple detections of the same action instance: we remove any candidate with significant overlap in space and time with a higher-scored candidate.

3.4 Tracking humans and objects

Our approach for modeling human-object interactions depends on the availability of human and object tracks in the same time period. For robustness, it is important to ensure

***Figure 3.4*: The DPT-MS tracker at test time**. *In the first frame (left) the cup is automatically detected by an object detector. In the subsequent frames no detections are found on the cup. DPT-MS produces an object track (dashed window) by propagating the detection from the first frame according to point tracks (white segments). In the last frame (right) the cup is again detected. DPT-MS adds this detection to the track and uses it to update the confidence score, but not the location of the track.*

the highest possible recall for both human and object tracks, as missing either of the two prevents the system from recognizing the action.

It is an elusive goal to design robust detectors and trackers to deal with difficult, small objects such as cigarettes or cups. Instead, we propose a tracking-by-detection approach that can be run on top of weak single-frame detectors, and produces a large number of candidate tracks in order to miss as few positive tracks as possible, see sec. 3.7.1 for an experimental evaluation. Then, in sec. 3.5 we introduce a highly discriminative descriptor that allows to mine for relevant human-object track pairs out of this pool of candidates.

3.4.1 Detection

Humans. Detecting humans in the C&C dataset is particularly hard due to their variety of appearance, pose, viewpoints and lighting conditions. The previous work of Klaser et al. [Kläser *et al.* 2010] used a human detector based on HOG features [Dalal and Triggs 2005] trained on C&C to learn the specific features of this dataset.

We take a more general approach by employing the generic part-based human detector presented in [Prest *et al.* 2011]. This detector combines four part detectors dedicated to different regions of the human body (including full-body, upper-body, and face). It was trained from external still images without using any C&C images [Prest *et al.* 2011, sec. 2]. Two of the four components of this combined detector are taken from the popular person detector of [Felzenszwalb *et al.* 2009].

Objects. Detecting small objects such as cups and cigarettes is an even harder task than detecting humans. In addition to being small, these objects present a high degree of pose and appearance variability. For this task we rely on the detection approach of Felzenswalb et al. [Felzenszwalb *et al.* 2009], which demonstrated excellent results on the PASCAL

VOC object detection challenge [Everingham *et al.* 2007a]. We use the windows from the positive object tracks obtained in $\mathrm{TR2}$ as positive training data. As negative training data we randomly sample windows from Caltech-101 [Fergus and Perona 2003].

3.4.2 Tracking

Tracking is needed at various stages of our approach. During training we need to track each action object starting from the initialization in a single annotated frame ($\mathrm{TR1}$). This is a traditional tracking task [Yang *et al.* 2005, Wu *et al.* 2008, Grabner *et al.* 2008]. Furthermore, during $\mathrm{TR1}$ we need to link over time human detections obtained automatically in individual frames. This is instead a tracking-by-detection task [Breitenstein *et al.* 2009, Ferrari *et al.* 2008, Kläser *et al.* 2010]. During testing ($\mathrm{TE1}$) tracking-by-detection is needed again for both humans and objects (as at this point we have an object detector from $\mathrm{TR2}$).

Previous works [Breitenstein *et al.* 2009, Ferrari *et al.* 2008, Kläser *et al.* 2010, Ramanan *et al.* 2007] have been successful in tracking people in realistic videos by linking the output of a person detector run independently on each frame (tracking-by-detection). However, tracking small objects such as cups or cigarettes in this manner is much harder because detectors tend to miss the object in many frames. As a consequence, the object motion is typically broken into many short tracks. Furthermore, tracking-by-detection does not work when we do not have a detector yet, i.e. when the object to be tracked is given only as a bounding-box in a single frame. This corresponds to the traditional tracking scenario where the target is annotated in one frame [Yang *et al.* 2005, Wu *et al.* 2008, Grabner *et al.* 2008].

We propose here a general-purpose tracking method to robustly track multiple targets in an integrated manner that encompasses both the traditional tracking of a target annotated in one frame and the tracking-by-detection scenario. Inspired by [Sivic *et al.* 2005], our algorithm takes as input any number of detection windows of the target, and propagates them forward and backward in time based on point-tracks. During this process, multiple windows that spatially meet in a frame are automatically merged in a single output track.

Our tracker, referred to by *dense point tracks [Sundaram* et al. *2010] – median shift* (DPT-MS), works as follows:

1. *Input*. A sequence of frames $\{s, \ldots, e\}$ and a set of detections $\mathcal{D}^i$ for each frame $i \in \{s, \ldots, e\}$. At least one detection in one frame is required for the algorithm to run. If more are provided, the algorithm will try to link them over time (tracking-by-detection). Any in-between situation is supported, e.g. where some targets have a single initialization window and others have a sparse set of windows output by a detector. For producing point tracks we compute long-term point tracks using the code of [Sundaram *et al.* 2010] over the entire sequence.

2. *Initialization*. Let f be the first frame for which a detection is available. For each detection $\mathcal{D}_j^f \in \mathcal{D}^f$ create a new track $\mathcal{T}_j$, and add it to the overall track set $\mathcal{T}$.

3. *Forward pass*. Loop over frames i from f to e

   (a) Loop over tracks $\mathcal{T}_j \in \mathcal{T}$

      i. *Update location*. The position of $\mathcal{T}_j^{i+1}$ of track $\mathcal{T}_j$ in frame $i+1$ is the position of $\mathcal{T}_j^i$ shifted by the median displacement between frame i and $i+1$ of the point tracks inside window $\mathcal{T}_j^i$.

      ii. *Include a detection*. If a detection $\mathcal{D}_k^{i+1}$ in frame $i+1$ substantially overlaps with $\mathcal{T}_j^{i+1}$, then it is assigned to $\mathcal{T}_j^{i+1}$. The detection $\mathcal{D}_k^{i+1}$ is then removed from $\mathcal{D}^{i+1}$. This step has no other effect for the moment. The detections assigned to a track will be used in step 6) to compute its confidence score.

   (b) *Add new tracks*. For each detection $\mathcal{D}_k^{i+1}$ that was not included into an existing track in step 3.(a).ii, we start a new track and add it to $\mathcal{T}$.

4. *Backward pass*. Store away the current tracks. Restart the process from step 2, this time over the reversed sequence from f to s.

5. *Concatenate forward-backward tracks*. Assemble the final tracks by concatenating the tracks from forward pass to the (reverse) tracks from backward pass.

6. *Confidence scores.* The confidence of a track is the average over the scores of the windows it contains, where the windows scores are normalized between 0 and 1. Windows which are not supported by any detection (see the two central images in fig. 3.4) are given a score of 0, thus penalizing the overall average.

An important problem this tracker addresses is that detectors of small objects such as cigarettes and cups tend to produce sparse detections in time. As we observed in *Coffee & Cigarettes*, it is common to have tens of frames without detecting the object. DPT-MS links detections even in this situation, see figure 3.4 for an illustration. Moreover, it can be used to track any object by providing a single initialization window in one frame, as the tracker updates the position of a window over time according to the median motion of its point tracks. Once a track is initialized, it does not require additional detections to survive, as opposed to [Ferrari *et al.* 2008, Kläser *et al.* 2010, Sivic *et al.* 2009]. Finally, note how DPT-MS tracks any number of detections in parallel without substantial increase in computation time.

Robust point tracks. For obtaining point tracks in step 1, we rely on the recent work on obtaining dense point trajectories [Sundaram *et al.* 2010] from large-displacement optical flow [Brox and Malik 2011] (LDOF). LDOF is a variational technique that integrates discrete point matches, namely the midpoints of regions, into a continuous energy

formulation. The energy is optimized by a coarse-to-fine scheme to estimate large displacements also for small scale structures. As opposed to traditional optical flow, the algorithm [Sundaram *et al.* 2010] tracks points over multiple frames, not only over two.

3.5 Modeling human-object interactions

In this section we model the interaction between a human track $\mathcal{H}$ and an object track $\mathcal{O}$ in terms of relative position and motion features (stages TR4 and TE3). These features are computed for a human-object track pair, which have been formed before. Positive human-object pairs are formed easily at training time, as there is only one possible pair for each annotation cuboid (TR1). Instead, forming negative training pairs, and all pairs at test time, requires a dedicated procedure which we describe in sec. 3.5.2. In sec. 3.5.1 we start by presenting our interaction descriptor, which we compute for any human-object pair.

3.5.1 Interaction descriptor

In the following we describe the relative location, area and motion of the object track wrt the human track in the time interval $[t_{min}, t_{max}]$ in which they both exist (i.e. the intersection of their temporal extents). Note that both $\mathcal{H}$ and $\mathcal{O}$ have a window $\mathcal{H}^t$ and $\mathcal{O}^t$ in every frame $t \in [t_{min}, t_{max}]$, as our tracker never skips a frame (sec. 3.4.2).

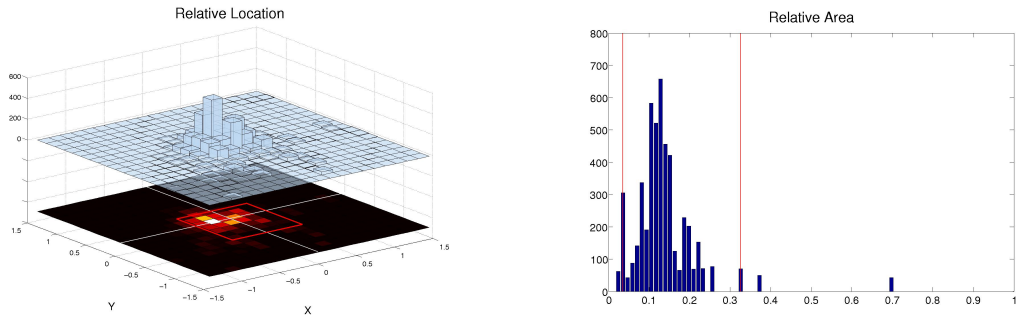

***Figure 3.5*: Learning the interaction ranges.** *Histograms of relative location (left) and relative area (right) accumulated over all positive training human-object pairs. The learned ranges are shown in red. The left plot shows that the location of the cup is typically in the middle of the human window along the horizontal axis and slightly above it along the vertical axis.*

At every frame t in the interval $[t_{min}, t_{max}]$ we compute three features:

1. *Relative location.* The relative location $\mathrm{l}(\mathcal{H}^t, \mathcal{O}^t)$ of the object window $\mathcal{O}^t$ wrt to the human window $\mathcal{H}^t$ in frame t

$$\mathrm{l}(\mathcal{H}^t, \mathcal{O}^t) = ((\mathcal{O}^t_x - \mathcal{H}^t_x)/\mathcal{H}^t_W \ , \ (\mathcal{O}^t_y - \mathcal{H}^t_y)/\mathcal{H}^t_H) \tag{3.1}$$

   where subscripts indicate a window's center x, y, width W and height H.

2. *Relative area.* The area of $\mathcal{O}^t$ relative to $\mathcal{H}^t$

$$\mathrm{a}(\mathcal{H}^t, \mathcal{O}^t) = \mathrm{area}(\mathcal{H}^t)/\mathrm{area}(\mathcal{O}^t) \tag{3.2}$$

3. *Relative motion.* The relative motion of the object wrt to the human is an important cue for distinguishing actions. We define this as the 2D vector

$$\mathrm{m}(\mathcal{H}^t, \mathcal{O}^t) = \mathrm{l}(\mathcal{H}^t, \mathcal{O}^t) - \mathrm{l}(\mathcal{H}^{t-1}, \mathcal{O}^{t-1}) \tag{3.3}$$

   the difference between the relative location $\mathrm{l}(\mathcal{H}^t, \mathcal{O}^t)$ in frame t and $\mathrm{l}(\mathcal{H}^{t-1}, \mathcal{O}^{t-1})$ in frame $t-1$. We represent this vector by its magnitude and direction.

We compute an interaction feature at every frame of a human-object pair and then aggregate them into a single descriptor of fixed dimensionality as follows. For each feature we accumulate its values over the time interval in a histogram. We independently $L1$-normalize each histogram and then concatenate them to obtain the final interaction descriptor. The 2D *relative location* and *relative motion* cues are quantized into 16-dimensional histograms each and *relative area* is quantized into 4 dimensions. This results in a total of 36 dimensions. Interestingly, we did not observe any improvement by using a higher dimensionality.

3.5.2 Forming human-object pairs

We describe here how to associate human and object tracks when collecting negative human-object pairs during training (stage $\mathrm{TR3}$) and when forming pairs during testing (stage $\mathrm{TE2}$). A simple approach would be to take all temporally overlapping pairs of human and object tracks. However, this would lead to a huge number of pairs, which would make action detection very slow. Instead, we perform here a preselection stage, where we associate pairs based on two interaction features from sec. 3.5.1.

Learning interaction ranges. Previous works on human-object interactions [Gupta *et al.* 2009, Prest *et al.* 2011, Yao and Fei-Fei 2010b, Yao and Fei-Fei 2010a] have shown the importance of limiting the spatial range of an action-object wrt a human. We learn the interaction range for the *relative location* and *relative area* features. After the training step $\mathrm{TR1}$, we have a set of positive human-object track pairs. For each frame in every human-object pair, we compute the two interaction features, see fig. 3.5 for their distribution.

For each feature, we then select the range of the feature such that 90% of the mass of the distribution is contained in it. Note how this threshold operates at a frame level thus discarding 10% of the outlying mass of the distribution and preserving relevant geometric information from the remaining frames.

Forming pairs. The ranges learned for the spatial interaction features are used to select spatially consistent pairs from the set of temporally overlapping ones. Fig. 3.5 illustrates the feature distributions and learned ranges for the drinking action from *Coffee & Cigarettes*.

3.5.3 Temporal chunking at test time.

In the above pairing scheme, the temporal extent of a test pair is simply the time interval during which both tracks exist. Instead, we would like to focus on the temporal segment where the action takes place. For this reason we introduce a multi-scale temporal sliding-window mechanism for the test human-object pairs. For our experimental results we use three temporal scales, which are learned from the training cuboids. Given the temporal duration of these cuboids, k-means determines three clusters. The durations corresponding to the cluster centers are used as temporal scales. The step size is fixed to 10 frames in all our experiments. The output of this procedure is a large number of overlapping test pairs which are then scored by our action classifier $\mathrm{TE3}$ (sec. 3.6). As a final step, we apply non-maxima suppression in order to suppress multiple detections of the same action instance: we remove any candidate with significant overlap in space and time with a higher-scored candidate.

3.6 Action classifier

This section presents how to train the action classifier (stages $\mathrm{TR4}$ and $\mathrm{TR5}$). We train multiple classifiers based on different features capturing complementary aspects of actions. The goal of each classifier is to decide whether a human-object track pair $(\mathcal{H}, \mathcal{O})$ is an instance of the action class. In a final step, we combine the output of all classifiers into a single action classifier. This is used to score candidate track pairs during testing (stage $\mathrm{TE3}$).

Human-object interaction classifier. The training stage $\mathrm{TR4}$ outputs an interaction descriptor (sec. 3.5.1) for each training $(\mathcal{H}, \mathcal{O})$ pair. We train an SVM classifier with an intersection kernel [Maji *et al.* 2008] to separate descriptors from positive and negative pairs.

Action-object classifier. For each training pair $(\mathcal{H}, \mathcal{O})$, we collect the score of the object detector in each frame of the object track $\mathcal{O}$. The maximum value over the track is taken

as the output of this classifier. Given that the object might be hard to recognize in many frames due to viewpoint changes and localization inaccuracy, the maximum value gives the track a high score as long as at least one frame has a high score.

3DHOG-track classifier. We compute the 3DHOG-track features [Kläser *et al.* 2010] on the human track $\mathcal{H}$. This feature extends the HOG image descriptor to videos by extracting 3D HOG descriptors for spatio-temporal subvolumes of the track. It goes beyond a rigid spatio-temporal cuboid [Laptev and Perez 2007, Willems *et al.* 2009], as it adjusts piecewise to the spatial extent of the tracks. This introduces a more flexible representation, where the descriptor remains centered on the action. The 3DHOG-track feature is complementary to our human-object interaction descriptor, as it captures low-level appearance and motion information. Experimental results demonstrate their complementarity (sec. 3.7.1). We train a non-linear SVM classifier with RBF kernel to separate positive and negative training track pairs.

Combined action classifier. We linearly combine the output of the three above classifiers by training a linear SVM on the 3D vector of outputs from positive and negative training pairs. At test time, stage $\mathrm{TE3}$, we use this classifier to score all test pairs (obtained as in sec. 3.5.2).

3.7 Experimental results

We present an evaluation of our method on three existing dataset of human-object interactions: *Coffee & Cigarettes* [Laptev and Perez 2007] (sec. 3.7.1), the Gupta video dataset [Gupta *et al.* 2009] (sec. 3.7.2) and the Rochester Daily Activities dataset [Messing *et al.* 2009]. These datasets are complementary. *Coffee & Cigarettes* focuses on accurate spatio-temporal localization of two actions in a full-length realistic movie. In contrast, the Gupta video dataset and the Rochester Daily Activities dataset have more action classes, but the videos are taken in a controlled laboratory environment and each video clip contains only a single action. Furthermore, the task is multi-class classification of whole clips. Each clip contains only the action performed and the actor is in the image center, so the protocol does not evaluate action localization in space and time. We also investigate the performance of human and object tracks and human-object track pairs on *Coffee & Cigarettes* (sec. 3.7.1).

3.7.1 Evaluation on Coffee & Cigarettes

The film *Coffee & Cigarettes* consists of 11 short stories, each with different scenes and actors. The C&C dataset [Kläser *et al.* 2010, Laptev and Perez 2007] focuses on the actions drinking and smoking.

For drinking, the training set contains 41 video clips from 6 short stories. Additionally, it contains 32 samples from the movie *Sea of Love* and 33 samples recorded in a lab. This results in a total of 106 positive drinking samples for training. We collect 50000 negative samples (human-object pairs) from the 6 training short stories by selecting sequences which do not overlap with any of the positive samples.

For testing, instances of the drinking action are localized in 2 short stories not used for training, i.e., in 24 minutes of video, which contain 38 drinking samples corresponding to a total of 1.8 minutes.

The smoking training set contains 78 samples: 70 samples from 6 short stories of C&C (the ones used for training the drinking action) and 8 from *Sea of Love*. Analogously to the drinking action, we use 50000 human-object pairs from the 6 short stories of C&C not overlapping with any annotation as negative training samples. For testing, instances of the smoking action are localized in 3 short stories not used for training, i.e., in 21 minutes of video, which contain 42 smoking samples corresponding to a total of 2.3 minutes. Note the difficulty of spatio-temporal detection of such short actions in realistic full-length videos.

The training annotations [Laptev and Perez 2007] come in the form of cuboids $\mathcal{A}$ which define the location in time and space of humans performing the action. For each training cuboid we complement these original annotations with a bounding-box delimiting the action-object in *one frame*.

Evaluating the DPT-MS tracker

In this section we evaluate our tracker presented in sec. 3.4. While tracking humans in the C&C dataset is quite easy [Kläser *et al.* 2010], it is very challenging to track small objects such as cups and cigarettes which are central to recognizing actions. These objects are often very small, occluded by the person and in difficult lighting conditions.

We evaluate DPT-MS for tracking cup and cigarette objects in the training sequences for the drinking and smoking actions. We operate our tracker in a traditional scenario: we use the object location annotated in one frame of every positive training clip as initialization (sec. 3.3.1) and then run the tracker through the temporal extent of the action (typically < 100 frames).

For evaluation *only*, we manually marked the object bounding-box in each frame of the training clips (throughout the chapter these annotations are never used for training). We count a bounding-box output by the tracker as a correct detection if it overlaps with the ground-truth object by more than 50%. We measure recall R as number of correct detections divided by the number of frames where the object is visible. All other tracker outputs are counted as false-positives. Precision P is the number of correct detection. The F-measure combines these two measures as $F = 2PR/(P + R)$.

		[Grabner and Bischof 2006]	[Grabner *et al.* 2008]	[Kalal *et al.* 2010]	DPT-MS
Drink	recall	0.748	0.798	0.939	0.829
	precision	0.756	0.821	0.964	0.923
	f − measure	0.752	0.809	0.951	0.873
Smoke	recall	0.774	0.720	0.868	0.823
	precision	0.779	0.768	0.911	0.824
	f − measure	0.777	0.743	0.889	0.823

***Table 3.1*: Evaluation of our DPT-MS tracker.** *We compare to other trackers using recall, precision and f-measure.*

Results are presented in tab. 3.1 where we compare to three state-of-the-art techniques [Grabner and Bischof 2006, Grabner *et al.* 2008, Kalal *et al.* 2010]. Interestingly, DPT-MS outperforms the more complex approaches [Grabner and Bischof 2006, Grabner *et al.* 2008] on this dataset, although the approach of Kalal et al. [Kalal *et al.* 2010] does even better. However, it is important to note that, unlike [Grabner and Bischof 2006, Grabner *et al.* 2008, Kalal *et al.* 2010], DPT-MS is specifically designed to handle both traditional tracking (i.e. initialized from a manual annotation in one frame) as well as tracking-by-detection, which simultaneously tracks a large number of candidate windows and connects them over time. In sec. 3.7.1 we show that this is a crucial requirement for obtaining a sufficient recall in detecting and tracking the object of interest in the C&C dataset.

Finally, DPT-MS is computationally very efficient. Computing the point tracks of [Sundaram *et al.* 2010] takes 2 seconds per frame and represents nearly all the runtime of DPT-MS. The rest of the procedure (sec. 3.4.2) tracks simultaneously 1000 candidate windows over two frames in only 10 milliseconds and it is linear in the number of windows. In comparison, although [Kalal *et al.* 2010] tracks one window over two frames in 20 milliseconds, it would take 20 seconds to track 1000 windows, making it impractical on a full-length movie such as the C&C dataset.

Evaluating human and object tracks

Humans. In order to compare the human detection and tracking performance of our method with the one from Klaeser et al. [Kläser *et al.* 2010] we evaluate on their dataset. This dataset is composed of 137 frames of C&C [Laptev and Perez 2007], for which a total of 260 ground-truth bounding-boxes are available. These frames are extracted from sequences of the movie that are not part neither of the training nor the test set. Unlike the original C&C annotations that provide the location of humans performing the action, this dataset contains the location of *every* human in an image. A person is considered to be correctly localized when the predicted and ground-truth bounding-boxes overlap more than the PASCAL VOC criterion (i.e. Intersection-over-Union above 50%). Performance is summarized by average precision (AP).

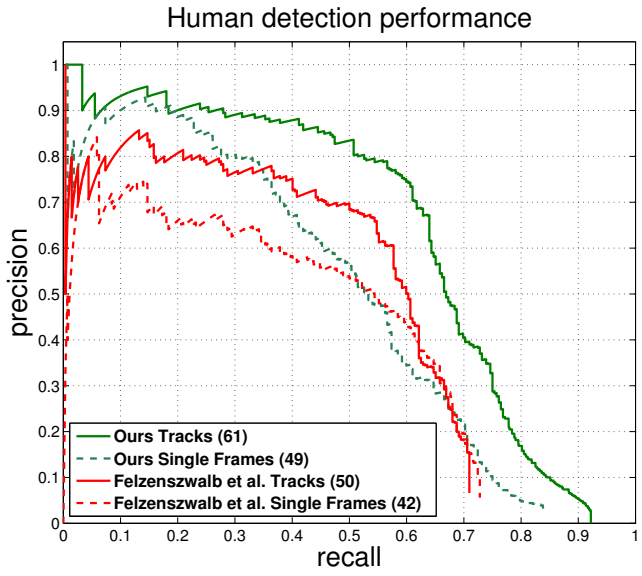

Figure 3.6: **Human detection performance**. *See text for details.*

Fig. 3.6 compares four methods. The two *Single frames* methods run a human detector on each test image independently: (i) the popular human detector of [Felzenszwalb *et al.* 2009], trained on the PASCAL 2007 VOC training set [Everingham *et al.* 2007a]; and (ii) our detector [Prest *et al.* 2011], which complements [Felzenszwalb *et al.* 2009] with additional detectors specialized for the face and upper-body regions (sec. 3.4.1). We can observe that the combination of different human part detectors [Prest *et al.* 2011] is beneficial on this difficult *Coffee & Cigarettes* dataset, improving over [Felzenszwalb *et al.* 2009] by 7% AP.

The *Tracks* methods link the detections output by the corresponding detector using the tracker presented in sec. 3.4.2. For this evaluation, detections are first computed on each frame in a short temporal interval around a test image, and then linked using the tracker. However, evaluation is only done on the 137 test frames, as for the *Single frames* methods. The associated score is the one of the track, i.e., the average detection score over the track.

The *Tracks* methods outperform substantially both their corresponding single-frame methods, confirming the contribution of our DPT-MS tracker. The "Ours Tracks" method gets +12% AP over the single-frame detector of [Prest *et al.* 2011]. Moreover, it also achieves 9% higher AP than the human tracker of [Kläser *et al.* 2010] (AP 52%). This is remarkable, as [Kläser *et al.* 2010] was trained specifically on C&C, while our detector is trained using only external material ([Prest *et al.* 2011, sec. 2]).

Objects. We evaluate object detection performance on frames selected from the test part of C&C. We sample either one or two frames from every positive sample depending on its temporal length. This results in 54 frames for drinking and 47 for smoking. We also evaluate on negative images (i.e. not containing the object): for each class we select a number

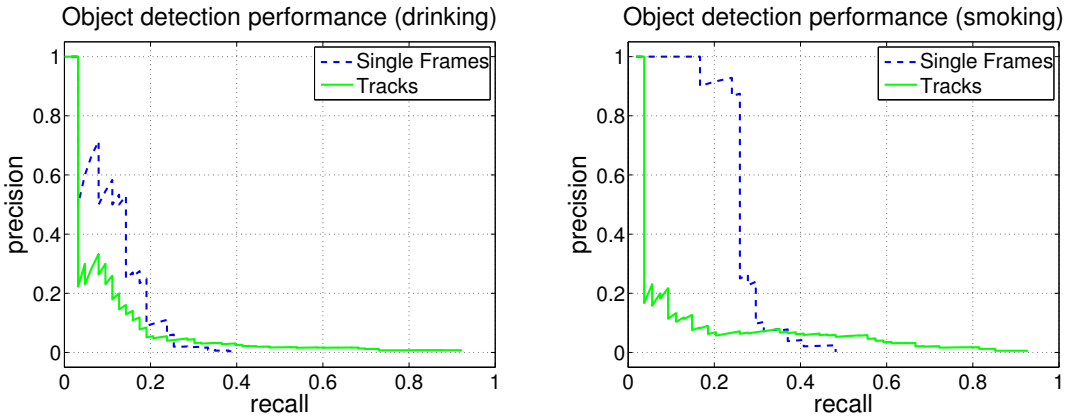

***Figure 3.7*: Object detection performance**. *See text for details.*

of negative images that reflects the proportion between positive and negative frames in the test set. This results in 500 negative images for drinking and 349 for smoking. As discussed in sec. 3.4.1, we train the object detection model of [Felzenszwalb *et al.* 2009] from all windows in the positive object tracks automatically obtained and negative images from Caltech-101. The only manual annotation used for training was a bounding-box in *one frame* of each action instance.

Fig. 3.7 compares the performance of the object detectors on the test part of the dataset in the *Single Frames* and the *Tracks* modes. The *Tracks* mode, although introducing some additional false-positives, doubles the maximum recall compared to the *Single Frames* mode, and detects more than 90% of all object instances. This fits the goal stated at the beginning of sec. 3.4: to produce a pool of candidate tracks which misses as few true object instances as possible. The lower performance of the *Tracks* method in terms of Average Precision is inherent to the context we operate in: we deal with detections which are sparse in time (typically less than 30% of a positive track's frames are supported by a detection) and every frame where a detection is missing penalizes the overall score of the track. As a result, the average track score loses significance. The track score could certainly be made more robust to outliers, but this was not necessary in our context, which requires maximum recall. Note also that this is not a problem when a reliable detector is available, as is the case for human detection (fig. 3.6).

We stress that object detection and tracking in this dataset is very difficult due to the highly cluttered scenes, varying lighting conditions, and especially the small size of the objects. In fact 76% of the objects cover less than 1.5% of the image surface.

Human-object pairs. In order to localize an action with our human-object interaction model, the human as well as the object track need to be present. In order to miss as few as possible human-object pairs performing the action, we keep all human and object tracks, i.e., we operate at the maximum recall level (right-most datapoints in fig. 3.6 and 3.7). The corresponding numbers are reported in tab. 3.2. Note that the number of tracks and

	Drinking	**Smoking**
$\lvert\mathcal{H}\rvert$	8924 (94%)	12558 (93%)
$\lvert\mathcal{O}\rvert$	49319 (92%)	71737 (93%)
$\lvert(\mathcal{H},\mathcal{O})\rvert$	418980 (90%)	1619284 (89%)

***Table 3.2**: Number of tracks and recall (in parentheses) for humans $\mathcal{H}$, objects $\mathcal{O}$ and human-object pairs $(\mathcal{H},\mathcal{O})$ on the Coffee & Cigarettes test set.*

	Drinking	Smoking
Interaction classifier	**32**	**16**
Object classifier	4	6
3DHOG-track classifier	52	22
Combination	**62**	**33**
Laptev et al. [Laptev and Perez 2007]	43	-
Willems et al. [Willems *et al.* 2009]	45	-
Klaeser et al. [Kläser *et al.* 2010]	54	25

***Table 3.3**:* **Average precision for spatio-temporal localization on C&C.** *First three rows: our individual classifiers. Fourth row: our full method combing the three classifiers. Last three rows: competing methods ([Laptev and Perez 2007, Willems* et al. *2009] do not report AP for smoking).*

the recall are reported for the final test datasets, i.e., the two and three short stories used to evaluate drinking and smoking localization.[1]

Given this set of human and object tracks, we form human-object pairs based on the approach described in sec. 3.5.2, i.e., use preselection based on relative location and area. This results in 418980 track pairs with a recall of 90% for drinking, and 1619284 track pairs with a recall of 89% for smoking (last row of table 3.2). This shows that the recall is sufficiently high to support the localization of most action instances. Note how this would not be the case if we kept only the 50% highest scoring human and object tracks. That would reduce the total number of human-object pairs by about four times, and recall would drop to 43% for drinking and 39% for smoking.

In the next section we will show that our interaction descriptor is sufficiently distinctive to discard the large number of track pairs which do not contain the action.

Evaluating action detection (localization in space and time)

We now evaluate the performance of our approach for spatio-temporal localization of the actions drinking and smoking on the *Coffee & Cigarettes* dataset and compare to the

[1] This explains the difference in recall for human tracks wrt figure 3.6, where the evaluation is performed on a different subset of C&C.

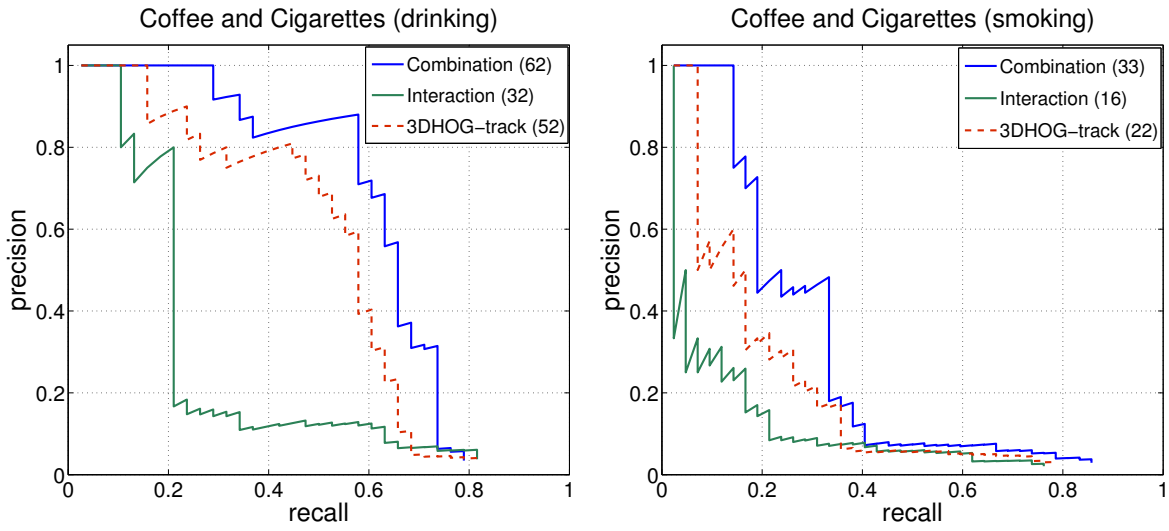

***Figure 3.8*: Precision-recall curves for C&C.** *Performance for spatio-temporal localization of the actions drinking (left) and smoking (right). For each method we present its average precision (AP) in parenthesis.*

state-of-the-art. We adopt the evaluation protocol of [Laptev and Perez 2007]: an action is correctly detected if the predicted spatio-temporal detection overlaps at least 20% with the ground-truth cuboid. The overlap between a ground-truth annotation cuboid $\mathcal{A}$ and a human-object pair $(\mathcal{H}, \mathcal{O})$ is given by $(\mathcal{A} \cap \mathcal{H})/(\mathcal{A} \cup \mathcal{H})$ (i.e. for evaluating our method we use the human track within a pair, as this corresponds to the standard protocol).

Fig. 3.8 shows precision-recall curves for drinking and smoking actions obtained with our combined method ('Combination') and the individual classifiers. Table 3.3 reports the average precision (AP) and compares to [Laptev and Perez 2007, Willems *et al.* 2009, Kläser *et al.* 2010]. The classifier based on the score of the object detector (second row) performs very poorly, which confirms that a human-object interaction cannot be defined purely based on the appearance of the object involved. The human-object interaction model we propose achieves good performance already when used on its own (first row). This shows how the relative location and motion of the object wrt the human is a distinctive feature characterizing the human-object interaction. More importantly, combining it with the low-level 3DHOG-track descriptor [2] improves on both and leads to a significant improvement over the state-of-the-art [Kläser *et al.* 2010] (+8% AP). This demonstrate that our interaction model is complementary to traditional low-level descriptors. Fig. 3.9 and 3.10 show some of the top-scored human-object pairs according to the combined action classifier.

[2]Our reimplementation of the 3DHOG-Track classifier achieves a slightly lower performance than the one reported by [Kläser *et al.* 2010] ($52/22$ vs $54/25$). This might be because [Kläser *et al.* 2010] uses a finer temporal sliding window for the test tracks (7 scales vs our 3).

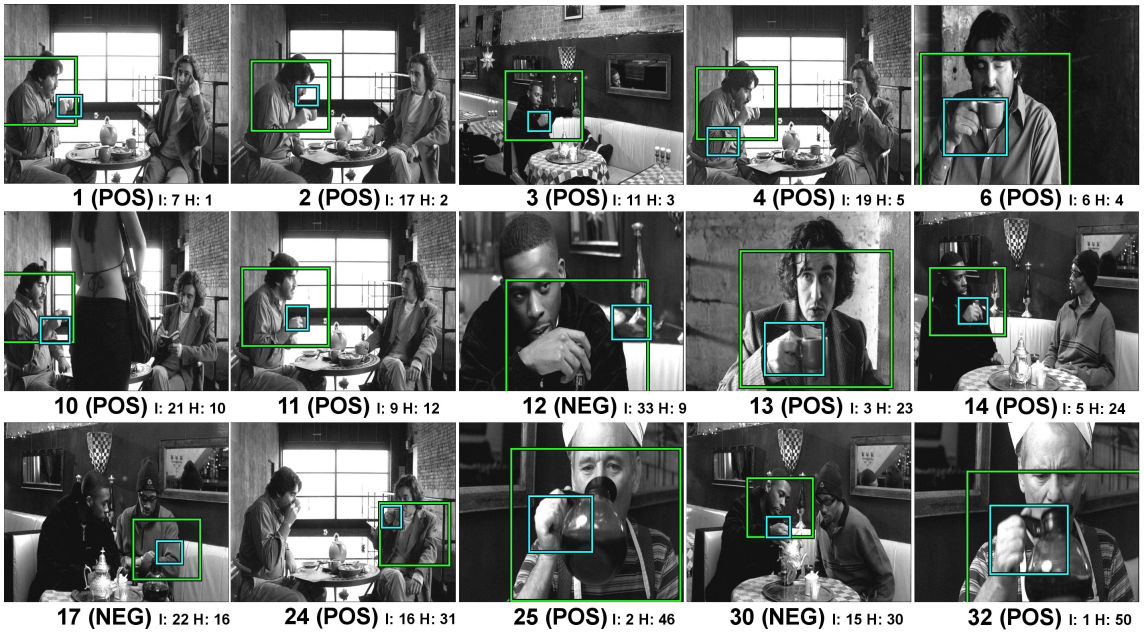

Figure 3.9: **Drinking results.** *Human-object pairs localized in test videos. The ordering corresponds to the ranking of the combined action classifier. We also show the rank of the individual classifiers separately (I: interaction classifier, H: 3DHOG-track classifier). These results show that the interaction and 3DHOG-track classifiers complement each other. Samples 13 and 14 have a relatively low 3DHOG-track score, whereas the interaction classifier successfully captures the discriminative motion of the object track. In contrast, for samples 2 and 4 the object track is incorrect, resulting in a lower interaction score rank, whereas the 3DHOG-track classifier correctly scores these samples highly. It is interesting, how our method finds object tracks also on unconventional objects such as the jug in samples 25 and 32, which receive top scores by the interaction classifier. For these examples 3DHOG-track fails due to the unusual object appearance. This confirms the ability of the interaction classifier to generalize the appearance of objects and describe their relative motion wrt to the human. Failure cases of the interaction classifier are often due to other objects moving in a similar way as action objects. For example in sample 30 the actor is pouring water from a teapot, resulting in a trajectory similar to the drinking action. For the 3DHOG-track classifier, a typical failure case is when low-level features perform poorly, as is the case in scenes with difficult lighting conditions, as in sample 14, or when the object has an unusual appearance, as in 25 and 32. Other failures by 3DHOG-track (classifying a negative as positive) are due to the actor being in a pose similar to the action, but not performing it, as in sample 12. Note that the interaction classifier receives a relatively low score as there is no motion.*

Figure 3.10: **Smoking results.** *Human-object pairs localized in test videos. The ordering corresponds to the ranking of the combined action classifier. We also show the rank of the individual classifiers (I: interaction classifier, H: 3DHOG-track classifier). In many cases the interaction and 3DHOG-track classifiers agree and assign both a high score to a positive sample. Complementary scores are obtained for samples 16, 23, 30 and 49: the interaction classifier correctly penalizes these negative samples without correct object motion, whereas 3DHOG-track is unable to distinguish them and assigns high scores. Note that sample 49 is a true negative, as it represents a person holding a cigarette and not smoking. For sample 19 the object track does not cover the correct object, thus the interaction classifier gives a low score, whereas the 3DHOG-track classifier assigns a high score. (*) For the smoking action we point out that the imprecise temporal extent of the annotations sometimes leads to an incorrect evaluation: samples 7 and 8 show a person smoking, but do not meet the spatio-temporal overlap threshold with the annotations.*

	Gupta video
Interaction classifier	**80**
Object classifier	37
3DHOG-track classifier	63
Combination	**93**
Gupta et al. [Gupta *et al.* 2009]	93

Table 3.4: **Average classification accuracy on the Gupta video dataset.**

3.7.2 Multi-class classification on Gupta video dataset

The Gupta video dataset [Gupta *et al.* 2009] contains 60 video clips with 10 actors performing 6 different actions, i.e. drinking from a cup, spraying from a bottle, answering a phone call, making a phone call, pouring from a cup and lighting a flashlight. For each action, the videos are split into 5 training and 5 test videos. Unlike the C&C dataset, these videos are shot in controlled conditions inside a laboratory with a static camera and a static background of uniform color. Furthermore, the video clips are restricted to the temporal extent of the action. Fig. 3.11 shows frames extracted from the Gupta video test set. Since the annotations used in [Gupta *et al.* 2009] are not available online, we have re-annotated the dataset to the same level as in 3.7.1: for each video one cuboid on the human performing the action and a bounding-box delimiting the object in *one frame*.

We train an action classifier for each of the six actions using as negative examples the training videos from the other classes. If two actions share the same object we merge the object tracks from the training videos and learn a single detector in step TR2 (this happens for cup and phone). Given a test video, we evaluate the action classifier score for each of the six actions and return as class label the one with the highest score. Note that the sliding window mechanism of sec. 3.5.3 is not required, as the video clips are already temporally segmented to the extent of the action. For evaluation we measure the percentage of test videos for which the algorithm predicts the correct label, as in [Gupta *et al.* 2009].

Table 3.4 shows the multi-class action classification results. Remarkably, the proposed interaction model already achieves 80% accuracy on its own and outperforms the 3DHOG-track. This demonstrates how our explicit modeling of the object motion trajectory is a strong cue for action classification. The interaction model performs better on this dataset than on C&C, because the objects are easier to track in these simpler imaging conditions.

The performance obtained with our combined action classifier is on par with the result from [Gupta *et al.* 2009]. Note that [Gupta *et al.* 2009] explicitly takes advantage of the static camera and background used in these videos, rendering it unsuitable for more complex videos such as C&C. Moreover, our method needs substantially less manual annotation for training than [Gupta *et al.* 2009], which requires the location of the person's hand and a pixelwise segmentation of the object in every frame of all training videos.

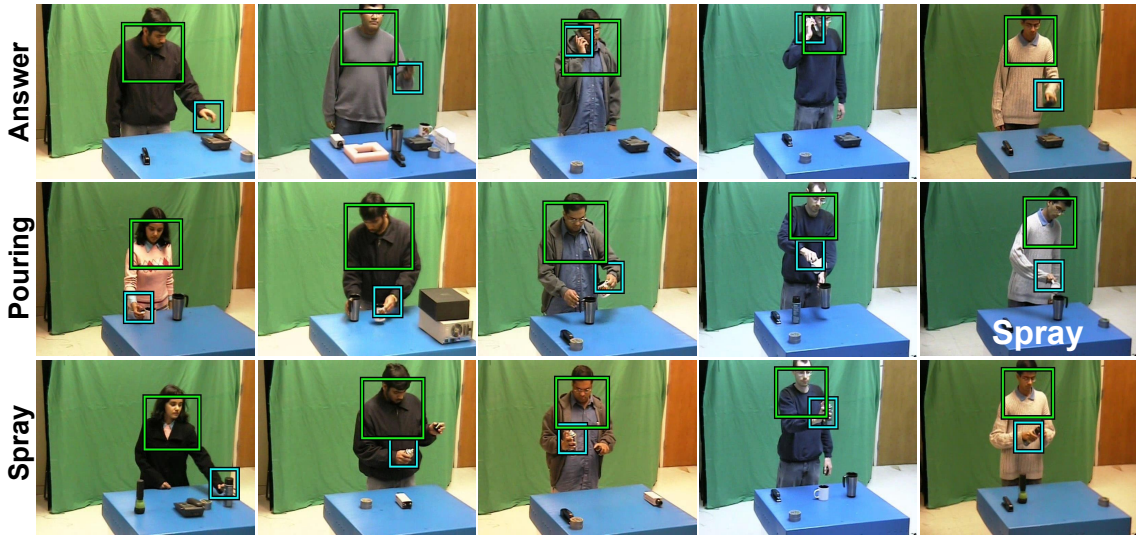

Figure 3.11: **Human-object pairs localized on the Gupta video test set with the combined classifier.** *Every row shows one frame from each of the five test sequences of a class. Actions in this dataset follow precise motion patterns: each row displays samples selected to follow the temporal pattern. We also show one of the two misclassified samples, indicated with the incorrect class label overlaid.*

	Rochester Daily Activities
Interaction classifier	74
Combination (our full method)	92
[Messing *et al.* 2009] (full method)	89
[Messing *et al.* 2009] (point tracks)	67
[Matikainen *et al.* 2010] (point tracks)	70

Table 3.5: **Average classification accuracy on the Rochester Daily Activities dataset.**

Figure 3.12 presents the confusion matrices, showing that most errors made by the interaction classifier are due to the similarity of the action 'lighting torch' with 'pouring water' and 'spraying'. These were distinguished in [Gupta *et al.* 2009] based on the color of the action-object, a feature which is not used here. Misclassifications between 'answering' and 'calling' are due to their similar motion. In the case of the combined classifier, there are only two misclassified samples, i.e, "light" is misclassified as "answer" and "pour" as "spray", see figure 3.11. These could probably be removed if colour information was used.

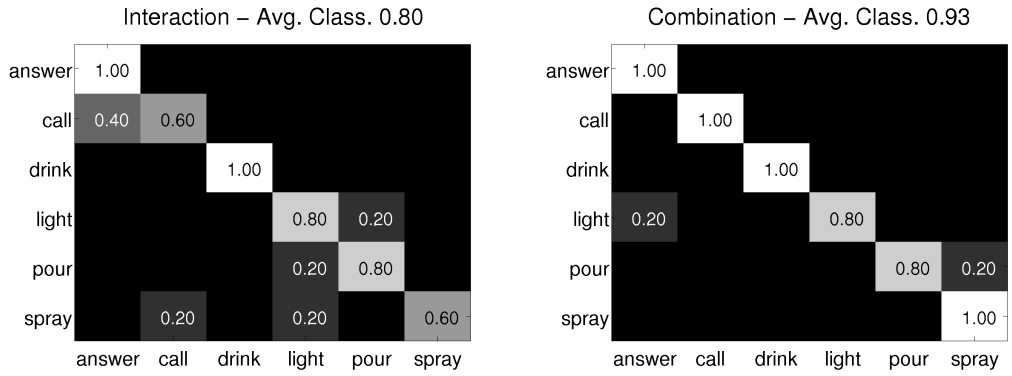

***Figure 3.12*: Confusion matrices on the Gupta video dataset.** *(Left) performance of the interaction classifier; (Right) combined action classifier.*

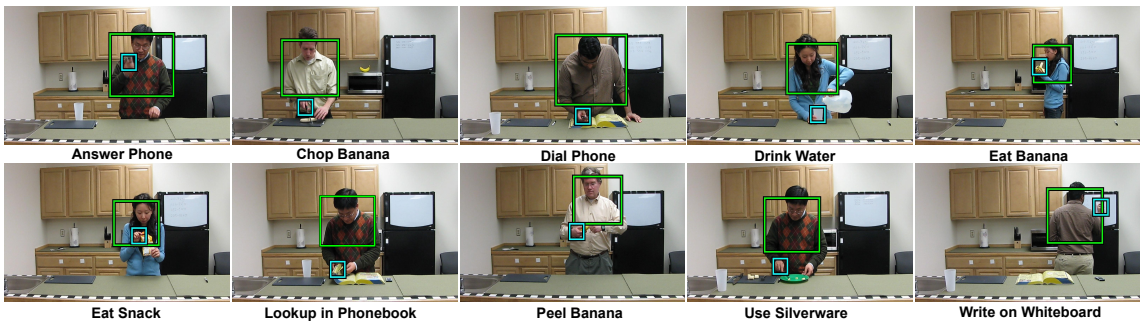

***Figure 3.13*: Human-object pairs localized on the Rochester Daily Activities dataset.** *We show one example for every class together with the automatically determined location of humans (green) and objects (cyan).*

3.7.3 Multi-class classification on Rochester Daily Activities dataset

The Rochester Daily Activities [Messing *et al.* 2009] dataset contains 10 activities of daily living recorded in a controlled environment with a static camera and background (fig. 3.13). Each activity was performed three times by five persons, for a total of 150 videos. Unlike the more challenging C&C dataset, the videos are restricted to the temporal extent of the action.

We first compare classification performance using our interaction descriptor (sec. 3.5.1) with two other recent motion descriptors aimed at capturing distinctive motion patterns in human-object interactions [Messing *et al.* 2009, Matikainen *et al.* 2010]. More specifically we compare to the Velocity Histories descriptor of [Messing *et al.* 2009], and the Sequencing Code Map Trajectory of [Matikainen *et al.* 2010]. Importantly, these methods are based on tracked low-level point features, unlike our method which explicitly detects the person and the object, tracks them and models their relative motion. Following the

evaluation procedure of [Messing *et al.* 2009], we train on all videos by four persons, and test on all videos of the fifth person. We repeate this leave-one-out test for each person and report average performance. As tab. 3.5 shows, our interaction classifier outperforms both competing methods (compare the "interaction classifer" row with the "point tracks" rows). This confirms the better descriptive power of explicit high-level modeling. Figure 3.14 presents a class-wise analysis of the classification result using our interaction descriptor.

We also report in tab. 3.5 the performance of our full method using our combined action classifier (sec. 3.6). It is 3% higher compared to the full method of [Messing *et al.* 2009], who also combines motion information with complementary contextual information.

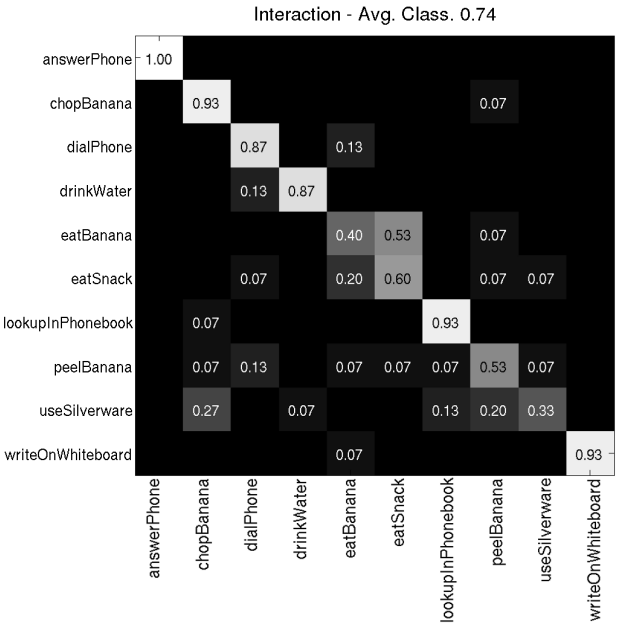

Figure 3.14: **Confusion matrix on the Rochester Daily Activities dataset.** *Performance of the interaction classifier.*

4

Learning Object Class Detectors from Weakly Annotated Video

4.1 Introduction

In chapter 2 we demonstrated how the high-level structure of visual concepts can be exploited to reduce the amount of annotation needed for learning human action models in static images. In chapter 3 we extended the idea to video and used motion to model human-object interactions as a rich description based on human and object tracks and their relative trajectories. This chapter leverages on the two previous efforts to tackle the more general problem of training object class detectors. In particular we want to tap into the large potential of consumer video available online (i.e. YouTube) and use it as an inexpensive source of training data (i.e. weakly-supervised) for learning object category models.

The standard way to train state-of-the-art object detection methods is to gather a large, diverse set of images and annotate them manually, possibly supported by crowd sourcing platforms such as Mechanical Turk [1]. The typical level of annotation needed is a bounding-box for each object instance [Felzenszwalb *et al.* 2009, Vedaldi *et al.* 2009, Viola and Jones 2001]. In general the performance of a detector increases with the number of annotated instances [Vijayanarasimhan and Grauman 2011]. Because manual annotation can be inflexible, expensive and tedious, recent work investigated methods that can learn from unlabeled or weakly annotated data [Arora *et al.* 2007, Blaschko *et al.* 2010, Chum and Zisserman 2007, Lee and Grauman 2011, Winn and Jojic 2005]. However, learning a detector without location annotation is very difficult and performance is still below fully supervised methods [Deselaers *et al.* 2010, Pandey and Lazebnik 2011].

In this chapter, we leave the common path of learning from images and instead exploit *video* as a source of training data. Interestingly, with a few exceptions [Ali *et al.* 2011, Leistner *et al.* 2011, Ramanan *et al.* 2006], learning from videos has been disregarded

[1] www.amazon.com/mturk

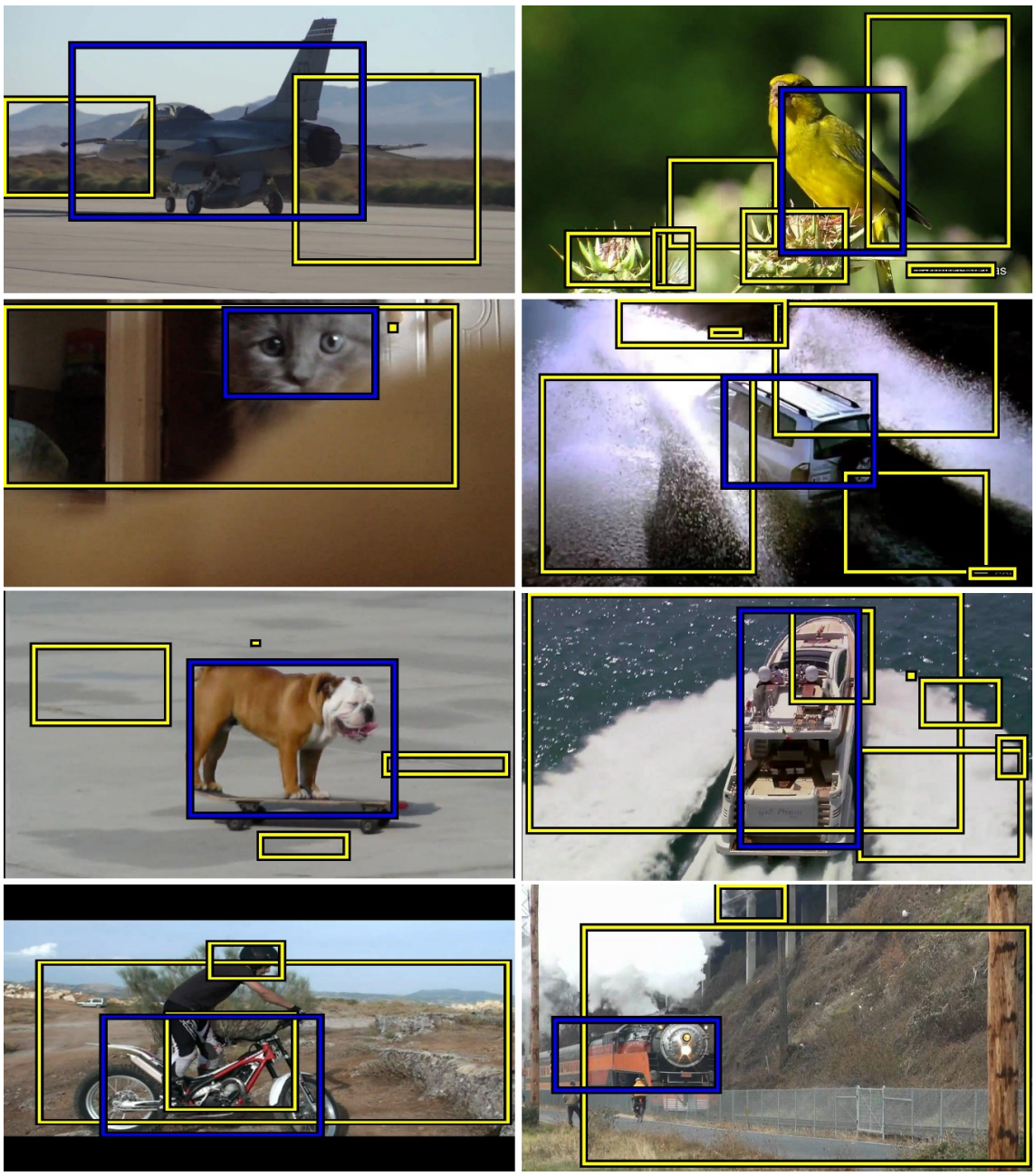

Figure 4.1: **Learning from Video.** *Yellow boxes represent tubes extracted by our method on the YouTube-Objects dataset. Blue boxes indicate the automatically selected tubes.*

by the vision community. Yet, video offers a rich source of data and is becoming more easily accessible through internet sources such as YouTube. The benefits of video include: (i) it is easier to automatically segment the object from the background based on motion information, (ii) each video shows significant appearances variations of an object, and (iii) a set of videos provides a large number of training images, as each video consists of many frames.

The main contribution of this chapter is an approach that learns high quality object detectors from real-world weakly annotated videos. We propose a fully automatic processing pipeline that localizes objects of a target class in a set of training videos (cf. figure 4.1) and learns a class-specific detector. Our approach requires only one label per video, i.e., whether it contains the class or not. It does not use any other information, such as the number or location of objects. In fact, the method does not even assume that all frames in the video contain the target class. To demonstrate the technique, we collect a video dataset from YouTube, coined *YouTube-Objects*.

Although we focus on learning from videos, we want to produce a detector capable of detecting objects in images at test time, such as the PASCAL07 [Everingham *et al.* 2007b]. However, individual frames extracted from real-world videos are often of lower quality than images taken by a high-quality camera. Video frames typically suffer from compression artifacts, motion blur, low color contrast, and lower signal-to-noise ratio of the sensor. This makes video frames somewhat different to images at the signal level. Hence, we cast the learning of detectors from videos as a *domain adaptation* task, i.e., learning while shifting the domain from videos to images. As it turns out, this is crucial for learning an effective detector from a combination of images and videos simultaneously.

Experiments on our YouTube-Objects dataset demonstrate that our technique can automatically localize target objects in videos and learn object detectors for several classes using only the video class label as manual input. As test data we use the challenging PASCAL07 object detection data set, and show that (i) detectors trained *only* from video already yield reasonable performance; (ii) detectors trained jointly from both video and images using domain adaptation perform better than when training from only the images (i.e. the training images of PASCAL07). In practice, our augmented detector outperforms the popular detector of [Felzenszwalb *et al.* 2009] on several classes.

In the next section, we review related work. In sec. 4.3 we introduce our technique for localizing objects in videos, and in sec. 4.4 we explain how to learn an object detector from them. In sec. 4.5 we state the task of improving object detectors for images by using video data as a domain adaptation problem and present an effective solution. In the experimental sec. 4.6 we evaluate our approach using PASCAL07 as test set.

4.2 Related Work

Learning from video. Most existing works on object detection train from images, i.e. [Felzenszwalb *et al.* 2009, Viola and Jones 2001, Vedaldi *et al.* 2009]. There is only a limited amount of work on learning object detectors from videos. In one of the earliest works, Ramanan et al. [Ramanan *et al.* 2006] showed how to build part-based animal models for tracking and detection without explicit supervisory information. The tracking is based on a simple hidden Markov model and the detector is a pictorial structure based on a 2D kinematic chain of rectangular segments. The model allows to detect new instances of the animal. Ommer et al. [Ommer *et al.* 2009] learn detection models as 3D point clouds using structure-from-motion. They train from controlled, hand-recorded video and the model can detect objects in test video, but not in images. Leistner et al. [Leistner *et al.* 2011] train a part-based random forest object detector from images and use patches extracted from videos to regularize the learning of the trees. Their approach captures the appearance variation of local patches from video and is tested on rather simple benchmarks.

Also related to our approach are works on tracking-by-detection [Grabner and Bischof 2006, Kalal *et al.* 2010]. Typically, a target object is marked by hand in one frame, or initialized with a preexisting detector, then a classifier is trained on-line in order to redetect the object in each frame. These approaches continuously adapt a detector specific to one video and do not attempt to train a generic class detector. In contrast, Ali et al. [Ali *et al.* 2011] proposed a semi-supervised boosting variant that uses space-time coherence of video frames to determine the similarity among objects used for training a detector. The method requires a subset of fully annotated frames in each training video. Testing is performed on videos of the same scene, but at different time instances.

Our technique differs from the above ones in several respects: (i) we localize target objects in multiple training videos fully automatically and, then, use them to train an explicit object detector, e.g. [Felzenszwalb *et al.* 2009]; (ii) our technique can train on videos alone and yet yield reasonable detectors for images; (iii) we operate on realistic video sequences downloaded from YouTube. The difficulty of applying state-of-the art vision algorithms to real-world videos from the web has been recently reported in [Zanetti *et al.* 2008].

Weakly-supervised learning from images. Many works address the problem of weakly-supervised learning of object classes in static images. They tackle different challenges such as image classification [Fergus *et al.* 2003, Crandall and Huttenlocher 2006, Arora *et al.* 2007, Galleguillos *et al.* 2008, Nguyen *et al.* 2009], segmentation [Winn and Jojic 2005, Russell *et al.* 2006, Alexe *et al.* 2010, Cao and Li 2007] and detection [Todorovic and Ahuja 2006, Chum and Zisserman 2007, Lee and Grauman 2009, Lee and Grauman 2011, Bagon *et al.* 2010, Blaschko *et al.* 2010, Deselaers *et al.* 2010, Pandey and Lazebnik 2011, Kim and Torralba 2009, Siva and Xiang 2011].

Most of these methods have been demonstrated to work on simplistic datasets such as Caltech4 [Arora *et al.* 2007, Crandall and Huttenlocher 2006, Fergus *et al.* 2003, Galleguillos *et al.* 2008, Lee and Grauman 2009, Nguyen *et al.* 2009, Winn and Jojic 2005] or Weizmann horses [Borenstein and Ullman 2004, Winn and Jojic 2005, Cao and Li 2007] where the object of interest is rather centered in the training images and occupies a substantial portion of it. Viewpoint and scale variations are also limited making it easier to spot recurring and discriminative object patterns and thus learn appropriate object models.

More recently a number of methods [Chum and Zisserman 2007, Kim and Torralba 2009, Deselaers *et al.* 2010, Pandey and Lazebnik 2011, Siva and Xiang 2011] addressed the problem of weakly-supervised learning of object detectors in more challenging dataset. These methods typically try to approximately localize object instances while learning a model of the class. The approach of [Chum and Zisserman 2007] iteratively refines the expected location of the object based on an initialization provided by discriminative local features. This approach tends to fail in heavily cluttered scenes or when the searched object only occupies a small portion of the image. The approach of [Pandey and Lazebnik 2011] is based on an iterative multi-stage global search for possible object locations on the challenging PASCAL07. However the authors manually provide the aspect ratio for the search object, reducing the overall difficulty. In [Siva and Xiang 2011] the localization of objects in the training images is treated as a Multiple Instance Learning problem, where each image is considered as a bag, either positive or negative. [Deselaers *et al.* 2010] propose a technique to select one window per training image out of a large pool of candidates, so as to maximize the appearance similarity of the selected windows. Our technique of sec. 4.3.3 can be seen as an extension of [Deselaers *et al.* 2010] to video.

4.3 Localizing objects in real-world videos

The goal of this section is to localize objects of a target class in realistic videos, e.g., collected from YouTube. The localized objects will then be used as positive training data for learning class-specific object detectors (sec. 4.4 and 4.5). We explore a Class-Generic approach, where motion is used to separate moving objects from the background allowing for generic object localization. A Class-Specific approach instead relies on pre-trained models from PASCAL07 to drive class-specific localization.

We start with an overview of our pipeline, see fig. 4.2. The input is a collection of realistic videos, all labeled as containing the target class. Each video is typically a collage of heterogeneous footage recorded at different times and places. Sec. 4.3.1 explains how we partition a video into shots, each corresponding to a different scene. In sec. 4.3.2 we localize objects in the shots. We explore two different approaches to accomplish this task:

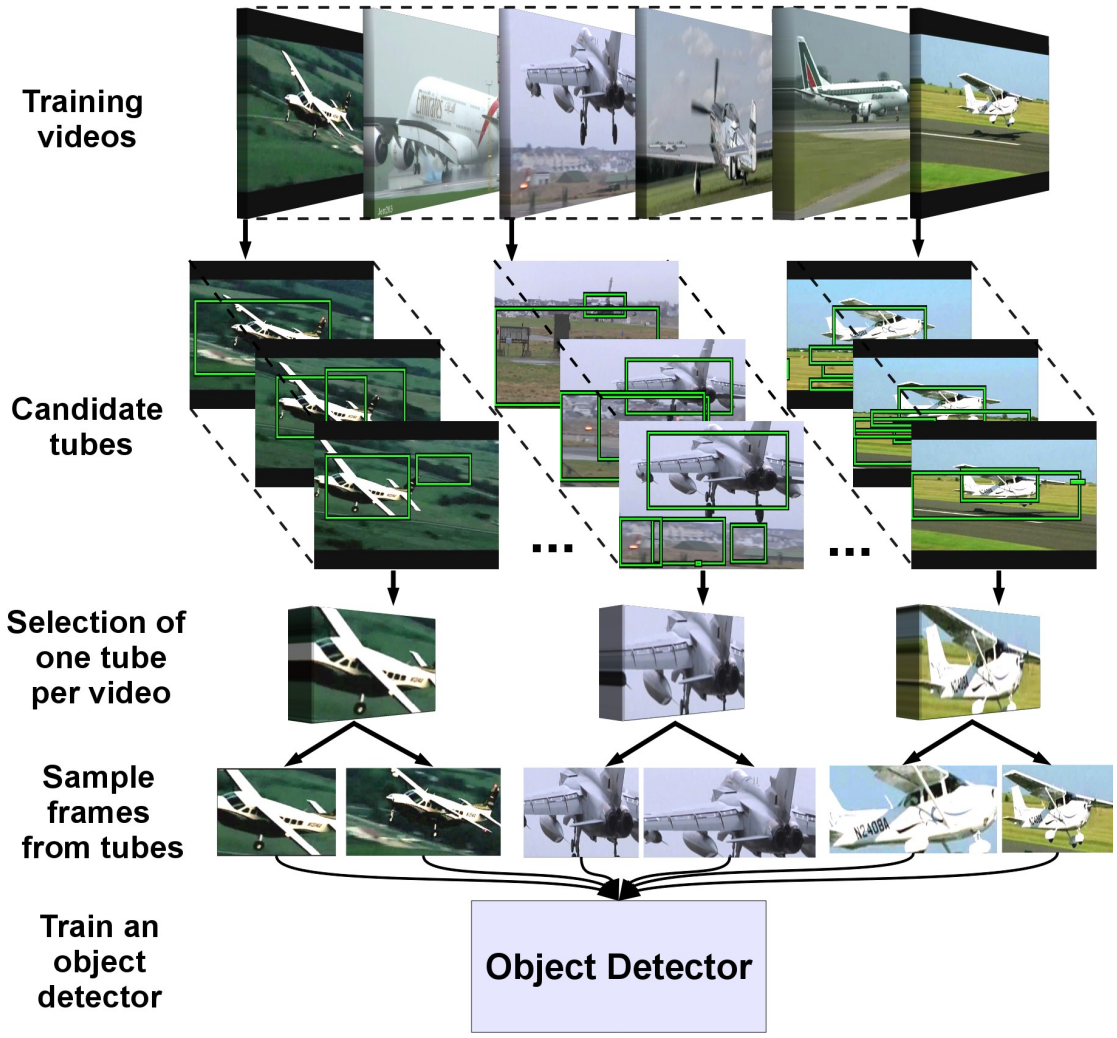

***Figure 4.2**: Overview of our approach.*

- **Class-Generic**: We extract segments of coherent motion from each shot, using the technique of Brox and Malik [Brox and Malik 2010] (fig. 4.4 top row). We then robustly fit a spatio-temporal bounding-box that we call *tube* to each segment (fig. 4.4 bottom row). There are between 3 and 15 tubes per shot. Typically, there is one tube on the object of interest and several on other objects or the background.

- **Class-Specific**: We use class-specific detectors pre-trained on the PASCAL07 training set. We detect the target object in all frames of the video category and we use motion to link the detections over time obtaining spatio-temporal tubes.

The Class-Generic and the Class-Specific approaches could be loosely related to the concepts of weak-supervision and full-supervision. This is in the sense that the Class-Generic variant localizes objects of the target class using the video label as the only human-

provided input. The Class-Specific instead relies on the additional PASCAL07 annotations used to train the class-specific detector.

Given the tubes over all shots in the input training set, we jointly select one tube per shot by minimizing an energy function which measures the similarity of tubes, their visual homogeneity over time, and how likely they are to contain objects (cf. sec. 4.3.3). The tubes selected by these criteria are likely to contain instances of the target class (fig. 4.4 bottom row, blue boxes). The selected tubes are the output of our localization algorithm. We use them to train a detector for the target class, cf. sec. 4.4.2.

4.3.1 Temporal partitioning into shots

Consumer videos are often the concatenation of footage recorded at different times and places. This results in abrupt changes of the visual content of the video, called *shot changes*. These are very informative as they usually indicate that a new object or a new scene is filmed, thus breaking motion consistency. We detect shot changes by thresholding color histogram differences in consecutive frames [Kim and Kim 2009]. We operate at a low threshold to ensure all shot changes are found. This partitions each video into multiple shots.

4.3.2 Forming candidate tubes

Class-Generic. We extract motion segments from each shot using the recent approach of Brox and Malik [Brox and Malik 2010] which is based on large-displacement optical flow (LDOF). LDOF is a variational technique that integrates discrete point matches, namely the midpoints of regions, into a continuous energy formulation. The energy is optimized by a coarse-to-fine scheme to estimate large displacements even for small scale structures. As opposed to traditional optical flow, this algorithm tracks points over multiple frames, not only over two.

The motion segments are obtained by clustering the dense point tracks based on the similarity in their motion and proximity in location. This works very well for rigid objects, where an entire object is typically put in a single segment. Moreover, in many cases this provides a consistent segmentation even if parts of an object move somewhat differently than the average motion of the whole object (e.g. the jumping cat in fig. 4.4). This process outputs a set of motion segments, defined by a collection of spatio-temporal point tracks (fig. 4.4 top row).

The last step is to fit a spatio-temporal bounding-box to each motion segment $\mathcal{M}$. At each frame t, we want to derive a bounding-box b^t from the set of point-tracks $\mathcal{M}^t$. Simply taking the enclosing box of $\mathcal{M}^t$ would lead to an unreliable estimate, as the outer region of a motion segment often contains spurious point-tracks. To alleviate this problem we

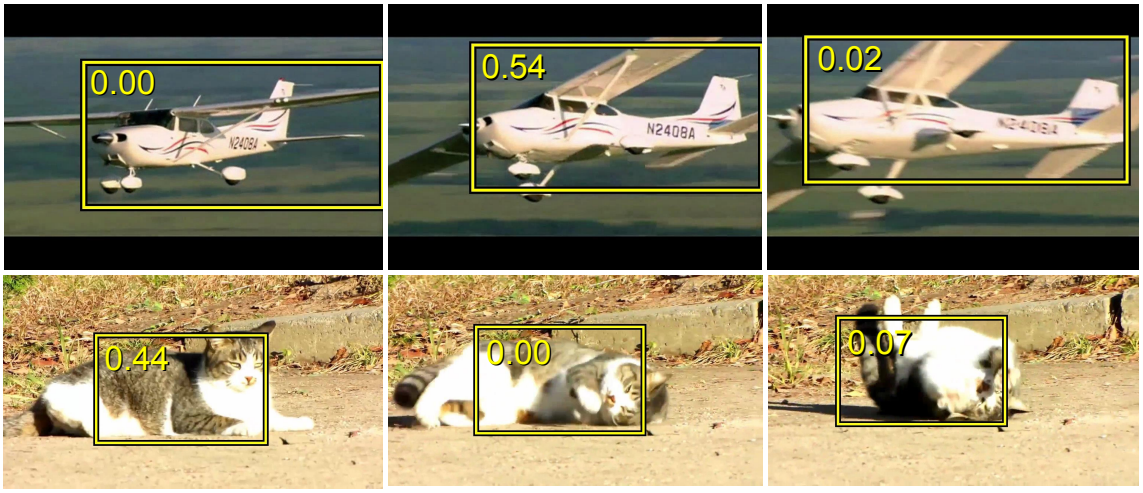

Figure 4.3*: Each of the sequences show a tube extracted with the Class-Specific variant of sec. 4.3.2. We overlay the normalized score that the pre-trained model attributes to the bounding-boxes in each frame. Particular views or deformations of the objects are scored high while others much lower. By using the DPT-MS tracker we are able to aggregate views and deformations that are known by the pre-trained model with novel ones, that represent valuable additional knowledge for the model.*

select a subset of point tracks $\hat{\mathcal{M}} \subset \mathcal{M}$ which are required to be in the top 80-percentile of the median location of $\mathcal{M}^i$ for all $i \in \{t, \ldots, t+4\}$. The box b^t is then defined as the bounding-box of $\hat{\mathcal{M}}^t$. This method is robust to outlier point tracks in $\mathcal{M}$ and leads to a temporal series of bounding-boxes which vary smoothly over time and are well-anchored on the moving object.

Class-Specific. We want to explore an alternative scenario which relies on pre-trained models learned on PASCAL07 to localize objects in video. Besides its comparative importance wrt the Class-Generic approach, this variant represents a more profound transfer learning effort. In particular most learning approaches today learn every category from scratch. Our goal here is instead to rely on class-specific models trained on a given domain (i.e. PASCAL07) to discover new samples in a different domain (i.e. YouTube-Objects) . Samples from a different domain represent an ideal complement to the existing knowledge of the detector to increase its intra-class generalization capability and hence its performance.

To accomplish this task we use the cascaded variant [Felzenszwalb *et al.* 2010] of the popular detector of [Felzenszwalb *et al.* 2009] as a class-specific detector pre-trained on the PASCAL07 training set. We detect the target object in all frames of the video category and track the detections over time using the DPT-MS tracker (sec. 3.4) leading to spatio-temporal tubes.

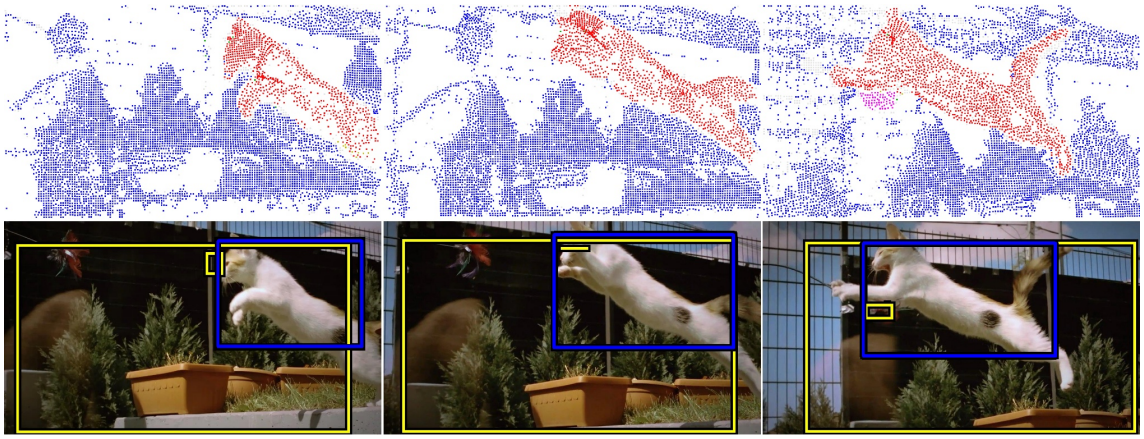

Figure 4.4: **Localizing objects in videos.** *(Top) Results of the motion segmentation. (Bottom) Tubes fit to the motion segments (the one selected by our approach is in blue).*

On the one hand this approach is inevitably biased towards detecting objects instances similar to those the model already observed at training time, thus limiting the "knowledge gain". On the other hand using motion allows the selection of previously-unseen variations in viewpoint or deformation as illustrated in fig. 4.3. A high-scoring detection act as a reliable anchor to initialize a tube. Tracking the detection over time using the DPT-MS tracker allows to discover different viewpoints or deformations of the same object in neighbouring frames. Note how these variations could hardly be detected by the model itself, as confirmed by the low score the model attributes to this newly discovered samples (see fig. 4.3).

To ensure the highest recall in detecting the objects of interest we operate the detector at a low threshold leading to an average of 500 detections per image. This results in a number of tubes per shot which is about 100 times higher compared to the Class-Generic approach. To reduce the number of tubes to a quantity manageable by the following stages (see sec. 4.3.3) we first apply non-maxima suppression in order to suppress multiple detections of the same object instance. We remove any candidate with significant overlap in space and time with a higher-scored candidate. Ultimately we retain the top 50 scoring tubes for each shot.

As a summary, the tubes allow to gather a variety of viewpoints and deformations of the target class which is an unique advantage of video data over standard static images dataset (i.e. PASCAL07).

4.3.3 Joint selection of tubes

At this point we have a set of candidate tubes $\mathcal{T}_s = \{\mathcal{T}_s^t\}$ in each shot s. Typically one tube contains the object of interest and the remaining ones background and other

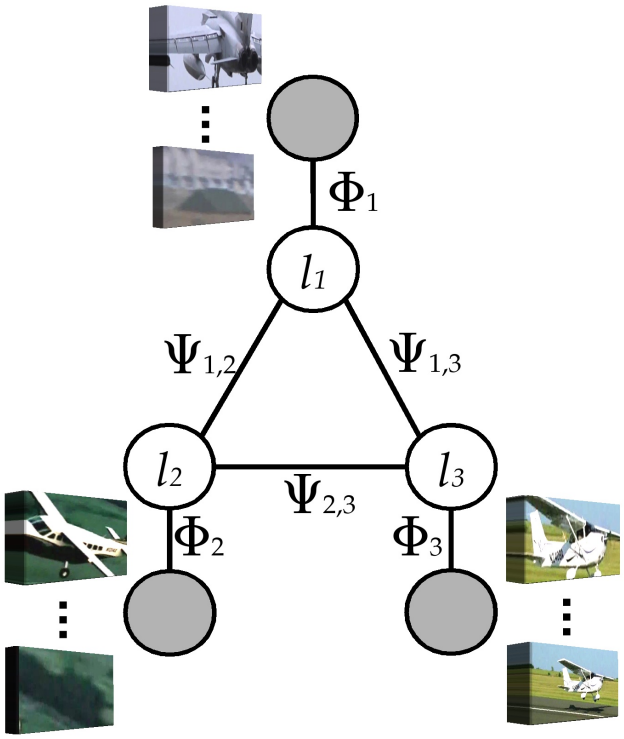

Figure 4.5: **Graphical model** *for the joint selection of one tube per shot. Candidate tubes from different shots are connected through pairwise potentials* $\mathbf{\Psi}$.

objects (fig. 4.4 bottom row). We describe in the following how to select the tubes $l_s \in \{1, ..., |\mathcal{T}_s|\}$ most overlapping with the object in each shot. We model this selection problem as minimizing an energy defined jointly over all N shots in all training videos. Formally, we define the energy of a configuration of tubes $L = (l_1, ..., l_N)$ to be

$$E(L|\alpha) = \sum_s \Phi_s(l_s) + \sum_{s,q} \Psi_{s,q}(l_s, l_q) \tag{4.1}$$

The pairwise potential $\mathbf{\Psi}$. It measures the appearance dissimilarity between tubes l_s, l_q in two different shots s, q. It encourages selecting tubes that look similar. It is a linear combination of two dissimilarity functions Δ that compare the appearance of the tubes over several frames according to two image features (details below)

$$\Psi_{s,q}(l_s, l_q) = \alpha_1 \Delta^{\text{BoW}}_{s,q}(l_s, l_q) + \alpha_2 \Delta^{\text{Phog}}_{s,q}(l_s, l_q) \tag{4.2}$$

The pairwise terms connect all pairs of shots. Every tube in a shot is compared to every tube in all other shots (also from other videos, fig. 4.5).

The unary potential Φ. It defines the cost of selecting tube l_s in shot s. It is a linear combination of four terms

$$\begin{aligned} \Phi_s(l_s) = \quad & \alpha_3 \Delta^{\mathrm{BoW}}_{s,s}(l_s, l_s) + \quad \alpha_4 \Delta^{\mathrm{Phog}}_{s,s}(l_s, l_s) \\ + \quad & \alpha_5 \Gamma_s(l_s) + \quad \alpha_6 \Omega_s(l_s) \end{aligned} \tag{4.3}$$

The first two terms prefer tubes which are visually homogeneous over time. They penalize tubes fit to incorrect motion segments, which typically start on an object but then drift to the background. For measuring homogeneity we use the same Δ functions as above, but we compare some frames of l_s to other frames of l_s.

The Γ term is the percentage of the bounding-box perimeter touching the border of the image, averaged over all frames in the tube. It penalizes tubes with high contact with the image border, which typically contain background (e.g., the blue segment in fig. 4.4 top row).

The Ω term is the objectness probability [Alexe *et al.* 2010] of the bounding-box, averaged over all frames in the tube. It measures how likely a box is to contain an object of any class, rather than background. It distinguishes objects with a well-defined boundary and center, such as cows and telephones, from amorphous background windows, such as grass and road. For this it measures various characteristics of objects in general, such as appearing different from their surroundings and having a closed boundary (see [Alexe *et al.* 2010] for more details).

Ultimately, when selecting tubes extracted with the Class-Specific variant of our localization method (sec. 4.3.2) an additional term Υ is added to eq. 4.3. Υ corresponds to the class-specific score assigned to the tube according to the pre-trained PASCAL07 model. The score is computed as the average score from the pre-trained detector on all the bounding-boxes composing the tube. More detail can be found in the DPT-MS tracker section (3.4).

Minimization. The objective of tube selection is to find the configuration L^* of tubes that minimizes E. We perform this minimization using the TRW-S algorithm [Kolmogorov 2006], which delivers a very good approximation of the global optimum $L^* = \arg\min_{\mathrm{L}} E(L|\Theta)$ in our fully connected model. TRW-S also returns a lower bound on the energy. When this coincides with the returned solution, we know it found the global optimum. In our experiments, the lower bound is only 0.05% smaller on average than the returned energy. Thus, we know that the obtained configurations are very close to the global optimum. The tubes selected by L^* are the output of our localization algorithm. They are used as input to train a detector for the target class in sec. 4.4.

Our technique is related to [Deselaers *et al.* 2010], where an energy function was minimized to select one window per image. We extended this idea to video, redefining

the problem to select one tube per shot, and introducing potentials relevant for spatio-temporal data.

Comparing tube appearance. To compute the Δ appearance dissimilarity function, a subset of boxes within a tube is represented by two complementary visual features. (1) BoW, a bag-of-word of dense SURF [Bay *et al.* 2008] features quantized to a visual vocabulary of 500 words, learned from 200 random frames. We use a 3-level spatial pyramid to enforce spatial consistency [Lazebnik *et al.* 2006]. (2) Phog, PHOG features [Bosch *et al.* 2007] capturing local shape as a distribution of HOG-features [Dalal and Triggs 2005] organized in a spatial pyramid [Lazebnik *et al.* 2006]. For each tube, we compute BoW and Phog for the bounding-boxes in 5 frames sampled uniformly over the temporal extent of the tube. The function $\Delta^{\mathrm{f}}_{s,q}(l_s, l_q)$ is the median of the χ^2 dissimilarity of the descriptors f over all 25 pairs of frames (one frame from tube l_s in shot s and the other frame from tube l_q in shot q).

Weights α. The scalars α weight the terms of our energy model. We learn the optimal α using constraint generation [Tsochantaridis *et al.* 2005] on a separate set of 50 held out shots of cars. We manually annotated the location of the car in one frame of each shot. The constraint generation algorithm efficiently finds the weights that maximize the localization performance wrt the ground-truth annotations. The α learned from this small held-out dataset is then used for all classes in our experiments.

4.4 Learning a detector from the selected tubes

The previous section selects one tube per shot likely to contain an instance of the target class. This corresponds to automatically localizing a bounding-box in each frame of the shot covered by the tube. We now describe how to train an object detector from this data. The main technical issue is sampling high quality bounding-boxes from the large pool offered by all selected tubes (sec. 4.4.1). The sampled bounding-boxes can then be used to train any standard object detector which requires bounding-boxes for training, e.g., [Felzenszwalb *et al.* 2009, Harzallah *et al.* 2009, Vedaldi *et al.* 2009] (sec. 4.4.2).

4.4.1 Sampling positive bounding-boxes

The tubes selected in sec. 4.3.3 offer a very large number of bounding-boxes that could be used as positive samples for training a detector. In our YouTube-Objects dataset, this number is about 10k-70k, depending on the class (tab. 4.1). This is too much data to handle for the training procedures of most modern detectors, which input about 1k positive samples [Felzenszwalb *et al.* 2009, Harzallah *et al.* 2009]. However, not all of these bounding-boxes contain the object of interest. Even with perfect tube selection, the

best available tube might contain bounding-boxes covering other image elements. This happens when tubes mostly on the object start or end on something else, e.g., when the object moves out of the field of view, or when the underlying motion segment drifts to the background. Moreover, some shots might not even contain the target class as they are automatically collected from YouTube (recall we only assume annotation at the video level, not at the shot level). Using such bounding-boxes as positive samples can confuse the training of the detector.

We introduce here a sampling technique to (i) reduce the number of positive samples to a manageable quantity; (ii) select samples more likely to contain relevant objects. The first step is to quantify the quality of each bounding-box in the pool. For this we use a linear combination of its objectness probability and the percentage of perimeter touching the border of the image, exactly as the Ω and Γ terms in eq. (4.3) (but applied to a single frame). The second step is to sample a fixed number S of bounding-boxes according to this quality measure (treating the set of probabilities for all samples as a multinomial distribution). In all our experiments we use $S = 500$.

4.4.2 Training the object detector

The bounding-boxes sampled as described in the previous subsection can be used to train any object detector. As negative training set we randomly sample 2400 video frames from other classes. From this data we train two popular and complementary detectors:

DPM: the part-based model [2] of Felzenszwalb et al. [Felzenszwalb *et al.* 2009]. It consists of a root filter and deformable part filters. The method has been demonstrated to yield highly competitive results on the PASCAL07 [Everingham *et al.* 2007b] dataset.

SPM: We employ SURF [Bay *et al.* 2008] features quantized into a 500-entries codebook learned on 500 frames randomly sampled from the training videos. A bounding-box is described by a 3-level spatial pyramid [Lazebnik *et al.* 2006]. At training time we collect an initial set of negative samples by sampling 10 objectness [Alexe *et al.* 2010] windows in every negative training image. We, then, train a preliminary SVM classifier using Intersection Kernel [Maji *et al.* 2008], and search exhaustively for false positives in the negative images [Dalal and Triggs 2005]. The classifier is retrained using these additional hard negatives. At test time the detector operates on 100000 windows uniformly sampled for every test image, followed by non-maxima suppression.

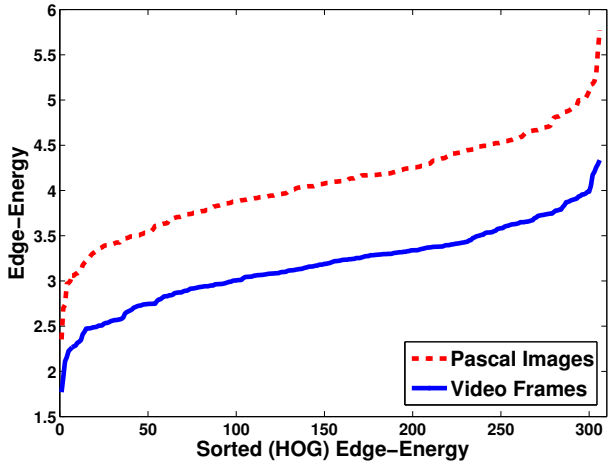

Figure 4.6: **Images vs videos**. *Left: the sum of the magnitude of the HOG features in an object bounding-box, normalized by its size (computed using the implementation of [Felzenszwalb* et al. *2009]). The 300 samples are sorted by gradient energy along the x-axis.*

4.5 Domain adaptation: from videos to images

Training a detector on videos and applying it to images corresponds to having training and test datasets from different domains. Recently, [Torralba and Efros 2011] highlighted the problems of visual classifiers if training and test set differ. [Zanetti *et al.* 2008] showed that this is especially valid for videos from the web that suffer from compression artifacts, low resolution, motion blur and low color contrast. Thus, it is hard to train a detector from video that yields good results on images.

To illustrate that video and still images can be seen as two different domains, we conducted two experiments. The first is illustrated in fig. 4.6 and compares the HOG representation for 300 aeroplanes on frames from our YouTube-Objects dataset to 300 similar aeroplane images from PASCAL07 [Everingham *et al.* 2007b]. We can observe that the gradient energy of images is significantly larger, i.e., around one third, than that of video frames. In the second experiment, we follow the *Name that Dataset* protocol of [Torralba and Efros 2011]. We trained an SVM on GIST features [Oliva and Torralba 2001] to distinguish video from images. This achieves a classification accuracy of 83%, confirming that images and videos are indeed different domains. Another difference between images and videos is the different distribution of viewpoints in which an object typically appears.

[2] www.cs.brown.edu/ pff/latent/

4.5.1 Domain adaptation

Domain Adaptation (DA) approaches try to improve classification accuracy in scenarios where training and test distributions differ. Let $\mathcal{X} = \mathcal{R}^F$ be the input space with F being the number of feature dimensions, and let $\mathcal{Y} = \{-1, 1\}$ be the output space (e.g. aeroplane or not). In DA, we have access to N samples from the source domain $\mathcal{X}^s$ (e.g. video) together with their labels $\mathcal{Y}^s$, and M samples from the target domain $\mathcal{X}^t$ (e.g. images) along with their labels $\mathcal{Y}^t$. Usually, $N \gg M$. The task of domain adaptation is to train a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that performs well on the target domain. Note that the case where the performance decreases for the target domain is referred to as *negative transfer* [Pan and Yang 2010]. There exists a number of approaches for domain adaptation [Pan and Yang 2010]. For our application we explore a few simple methods that have been reported to perform surprisingly well [Daume III 2007]. We summarize below the three approaches we experiment with, following the nomenclature and notation of [Daume III 2007].

All. The simplest possible approach is to ignore the domain shift and directly train a single classifier using the union of all available training data $\{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^N$ and $\{(\mathbf{x}_j^t, y_j^t)\}_{j=1}^M$. This is a simple baseline that should be beaten by any real domain adaptation technique.

Pred. A popular approach is to use the output of the source classifier as an additional feature for training the target classifier. More precisely, we first train a classifier $f_s(\mathbf{x})$ on the source data $\{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^N$. We, then, expand the feature vector of each target sample $\mathbf{x}^t$ to $[\mathbf{x}^t \;\; f_s(\mathbf{x}^t)]$. Finally, we train the target classifier using the expanded target training data $\{([\mathbf{x}_j^t \;\; f_s(\mathbf{x}_j^t)], y_j^t)\}_{j=1}^M$.

Prior. One of the most popular DA methods in Computer Vision is *Prior*, where the parameters of the source classifier are used as a prior when learning the target classifier [Tommasi *et al.* 2010, Aytar and Zisserman 2011]. For instance, an SVM can be regularized in form of $||\mathbf{w}_s - \mathbf{w}_t||^2$, where $\mathbf{w}$ is a learned weight vector.

LinInt. Another technique is to first train two separate classifiers $f_s(\mathbf{x}), f_t(\mathbf{x})$ from the source and the target data, and then linearly interpolate their predictions on new target data at test time. Thus, *LinInt* forms a new classifier $f_{st}(\mathbf{x}) = \lambda f_s(\mathbf{x}) + (1 - \lambda) f_t(\mathbf{x})$, where λ is the interpolation weight (it can be set so as to minimize the expected loss on the target domain).

Beside these four approaches, there exist many others. One advantage of Pred and LinInt over other techniques is that they can combine heterogeneous classifiers, where the dimensionality or even the kind of features differ between the source and target domains.

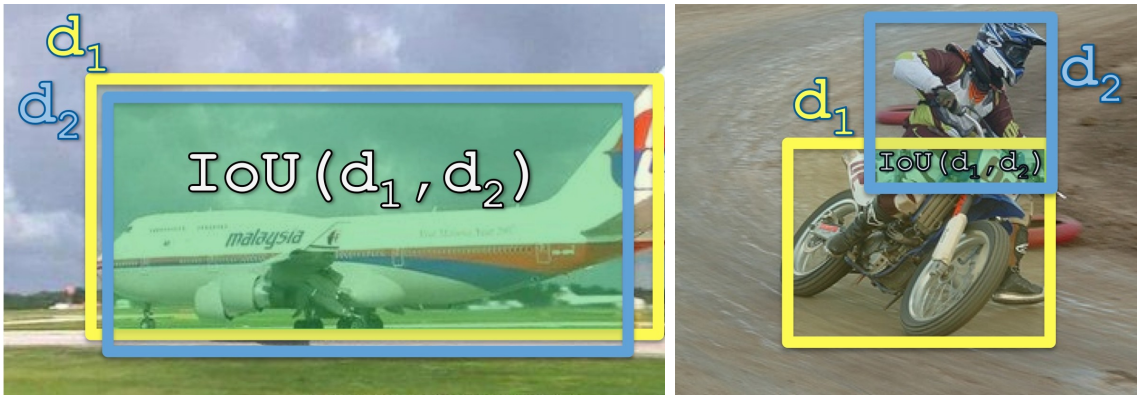

Figure 4.7: **LinInt for object detection**. *Left: there is a strong overlap between* $\mathbf{d_1}$ *and the best overlapping detection* $\mathbf{d_2}$. *Right: the best overlapping detection* $\mathbf{d_2}$ *has only a minimal overlap with* $\mathbf{d_1}$. *LinInt automatically reduces the combination weight through the* IoU *factor.*

4.5.2 LinInt for object detection

In the context of sliding-window object detection, as opposed to classification, *LinInt* cannot directly be applied. Different detectors might not only operate on different kinds of features, but even on different sets of windows in an image. For example, most detectors [Felzenszwalb *et al.* 2009, Pandey and Lazebnik 2011] learn an optimal aspect-ratio from the training data and only score windows of that aspect-ratio on a test image. In our case, the aspect-ratio learned from video data might differ from that learned from images. Other differences might include the sliding-window step and the sampling of the scale-space.

We introduce here a technique for combining arbitrary heterogeneous sliding-window detectors based on *LinInt*. We first let each detector score its own set of windows $\mathcal{D}$ on a test image I. Each window is represented as a 5-D vector $\mathbf{d} = \{x_1, y_1, x_2, y_2, s\}$ composed of the coordinates of the window in the image and its score. The two detectors, each trained separately, produce two separate sets of windows $\mathcal{D}_1$ and $\mathcal{D}_2$ for the same image I. We combine them into a set of windows $\mathcal{D}_c$ with the following algorithm. First, we initialize $\mathcal{D}_c = \emptyset$. Then, for each window $\mathbf{d_1} \in \mathcal{D}_1$ we do

1. Find the most overlapping window in $\mathcal{D}_2$: $\mathbf{d_2} = \arg\max_{\mathbf{d_2} \in \mathcal{D}_2} \mathrm{IoU}(\mathbf{d_1}, \mathbf{d_2})$ with $\mathrm{IoU}(\mathbf{d_1}, \mathbf{d_2}) = \frac{|d_1 \cap d_2|}{|d_1 \cup d_2|}$ the spatial overlap of two windows.
2. Combine the scores of $\mathbf{d_1}, \mathbf{d_2}$ with a modified *LinInt*: $s = \lambda \cdot d_1^s + (1 - \lambda) \cdot \mathrm{IoU}(\mathbf{d_1}, \mathbf{d_2}) \cdot d_2^s$ where $\lambda \in [0, 1]$ weights the two detectors.
3. Add a new window to $\mathcal{D}_c$ with the coordinates of $\mathbf{d_1}$ but with score s.

class	videos	shots	frames	class	videos	shots	frames
aeroplane	13	1097	71327	cow	11	212	29642
bird	16	205	27532	dog	24	982	82432
boat	17	606	74501	horse	15	432	70247
car	9	208	14129	motorbike	14	511	40604
cat	21	220	42785	train	15	1034	117890

***Table 4.1**: Statistics of our YouTube-Objects dataset. In total the dataset is composed of* 155 *videos, divided into* 5507 *shots and amounting to* 571089 *frames.*

This procedure is flexible as it can combine detectors defined on arbitrary sets of windows. It matches windows between the two sets based on their overlap and combines scores of matched pairs of windows. In practice two matched windows often have very high overlap and are almost identical. However, in the case a window from $\mathcal{D}_1$ has no good match in $\mathcal{D}_2$, our technique automatically reduces the combination weight. Fig. 4.7 illustrates two different scenarios for LinInt.

4.6 Experiments

4.6.1 Dataset

Our YouTube-Objects dataset is composed of videos collected from YouTube. For each of 10 object classes, we collected between 9 and 24 videos, whose duration varies between 30 seconds and 3 minutes (tab. 4.1). The videos are weakly annotated, i.e. we only ensure that at least one object of the relevant class is present in each video. For evaluating our automatic localization technique, we annotated bounding-boxes on a few frames containing the object of interest. For each class we annotated one frame per shot on $100 - 290$ different shots. Importantly, these annotations are used exclusively to evaluate our technique (sec. 4.6.2). They are not input at any point in our fully automatic pipeline.

4.6.2 Localizing objects in the training videos

We evaluate here the quality of object localization in the training videos. We adopt the CorLoc performance measure used in [Deselaers *et al.* 2010, Pandey and Lazebnik 2011], i.e., the percentage of ground-truth bounding boxes which are correctly localized up to the PASCAL criterion (intersection-over-union $>= 0.50$).

Both the Class-Generic and the Class-Specific variant of our localization method (sec. 4.3.2) are considered. Moreover, we evaluate separately the quality of (i) the tube extraction process (sec. 4.3.2) and of (ii) the tube selection process (sec. 4.3.3). To evaluate the tube

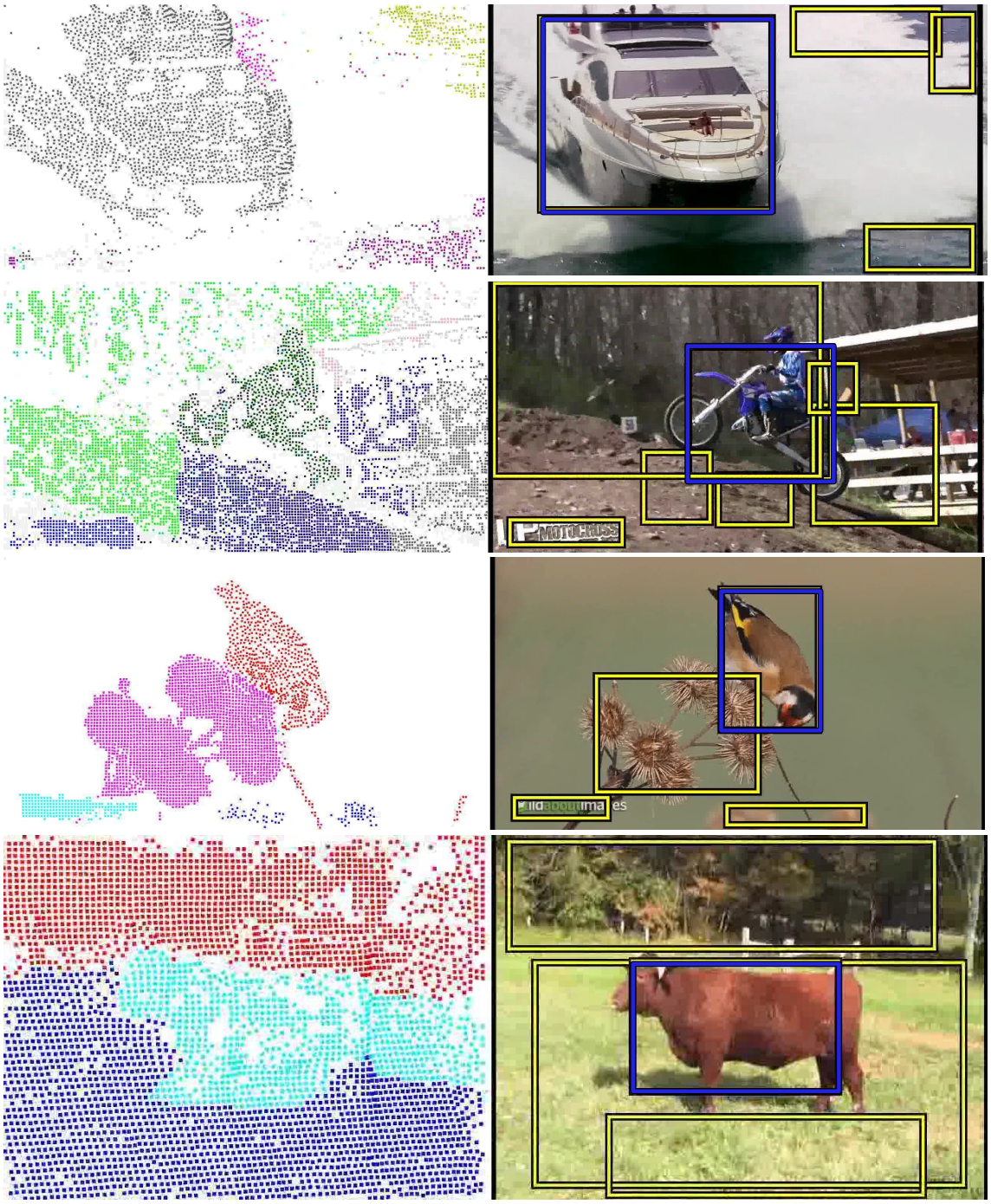

Figure 4.8: **Localizing objects in videos (success cases)**. *We show results of our Class-Generic variant for object localization in video (sec. 4.3). The first column illustrates the result of the motion segmentation while in the second the tubes fit to the motion segments are displayed (the one selected by our approach is in blue). In most cases when a motion segment covering the object of interest is avaialable, our method makes the right selection.*

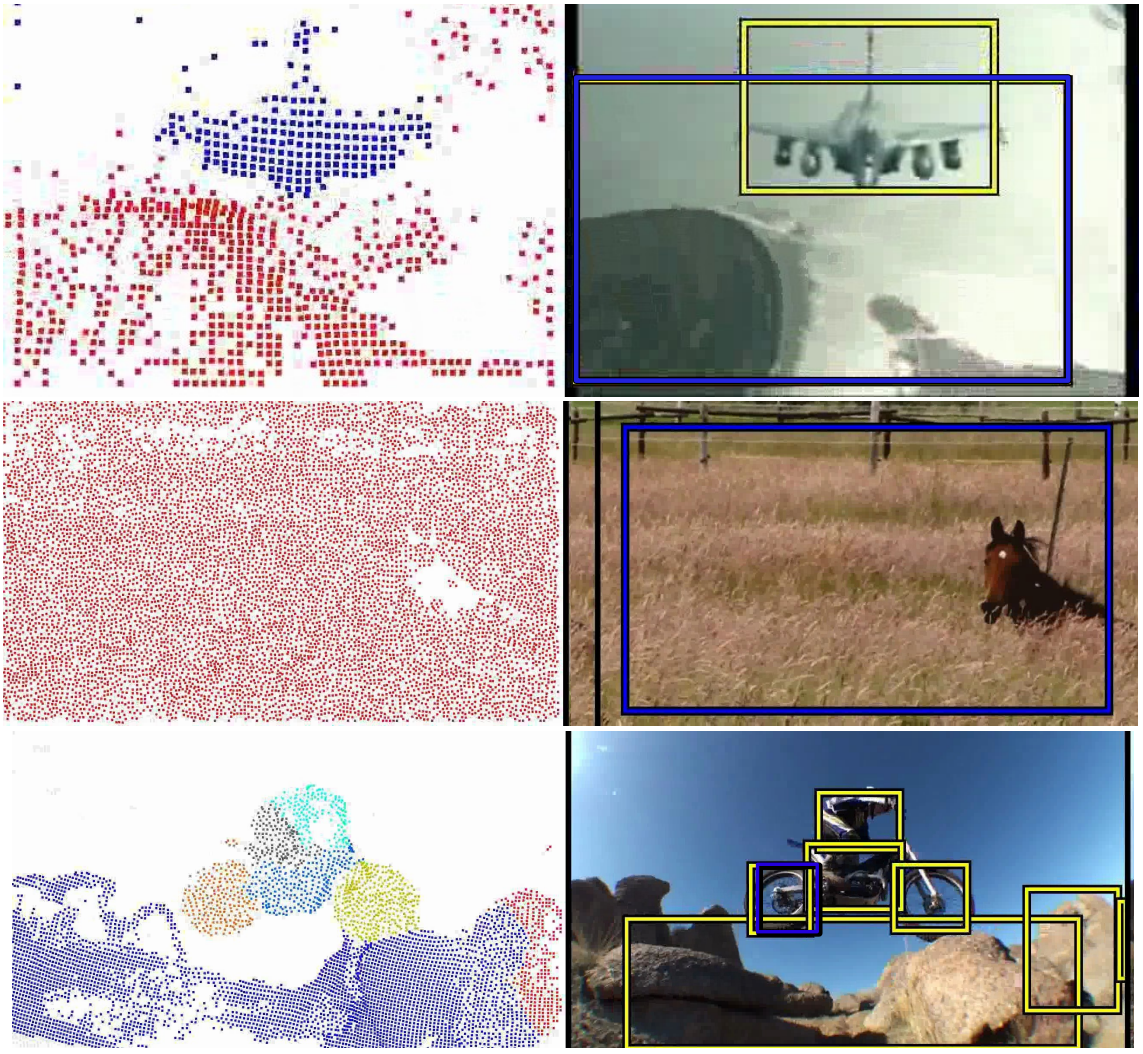

Figure 4.9: **Localizing objects in videos (failure cases)**. *We analyze failure cases of our Class-Generic variant for object localization in video (sec. 4.3). We show results of the motion segmentation (first column) and tubes fit to the motion segments (the one selected by our approach is in blue). In the first row, besides the presence of an appropriate candidate tube the automatic selection selected the wrong one. The second row shows how lack of distinctive motion and heavy occlusion lead to a single motion segment covering the whole image. In the third row, due to the perspective and irregular motion of the stunting motorbike, a single object is over-segmented into multiple parts.*

		aero	bird	boat	car	cat	cow	dog	horse	mbike	train	AVG
CG	Extraction	53.9	19.6	38.2	37.8	32.2	21.8	27.0	34.7	45.4	37.5	34.8
	Selection	51.7	17.5	34.4	34.7	22.3	17.9	13.5	26.7	41.2	25.0	28.5
CS	Extraction	60.1	53.6	58.8	84.7	28.9	82.1	50.4	76.0	78.4	60.7	63.4
	Selection	45.8	40.2	42.7	59.2	14.0	52.6	5.0	34.7	57.7	37.5	38.0

***Table 4.2**: Evaluation of the object localization performance of our method. We show results for the Class-Generic (CG) and Class-Specific (CS) variants separately (see sec. 4.3.2). Performance is the percentage of correctly localized objects for (i) the best available extracted tubes (row "Extraction") and (ii) the automatically selected tubes (row "Selection").*

extraction process, we select the tube with the maximum overlap with the ground-truth bounding box. The CorLoc of this best available tube can be seen as an upper-bound on the CorLoc that can be achieved by the automatic tube selection process.

The results are shown in tab. 4.2. Our automatic tube selection technique is very effective for the Class-Generic variant: it performs close to the upper-bound the majority of classes, indicating it selects the best available tube most of the time. Cat, dog and train present higher intra-class variability which makes it harder to leverage on recurrent appearance patterns to select tubes covering the target objects. The performance of tube extraction varies substantially from class to class and is in general higher for rigid objects. The overall performance of the Class-Generic variant is satisfactory, but could be improved as it is currently missing many objects. The main reason for this is the difficulty of motion segmentation in low quality, uncontrolled YouTube videos (see fig. 4.9).

Localization in the Class-Specific variant offers instead a very high upper-bound thanks to the pre-trained model of PASCAL07 that drives the localization. The automatic selection performs at more than 60% the upper-bound. We interpret this as a good result given the much larger candidate space (about 20 times more) compared to the Class-Specific case.

4.6.3 Training from video

Here we test on the 4952 images from the PASCAL07 test set using the official protocol [Everingham *et al.* 2007b]. This is a very challenging dataset, with large variations in pose, appearance and scale. We report results on a subset of 10 classes, corresponding to objects that move and for which enough video data is available online. Tab. 4.3 presents results for four different training regimes:

VOC: models trained on manual bounding-box annotations from the PASCAL07 $\mathrm{Train}+\mathrm{Val}$ image set.

VMA: models trained on the manually annotated frames from YouTube-Objects (sec. 4.6.1).

		aero	bird	boat	car	cat	cow	dog	horse	mbike	train	mAP
SPM	VOC	22.5	10.0	10.4	27.4	22.1	10.8	13.4	21.1	25.8	27.3	19.1
	VMA	16.8	3.9	2.6	18.7	16.5	10.1	6.0	8.5	15.9	12.0	11.1
	VID-CG	14.0	9.4	0.6	15.0	14.9	9.4	7.5	11.9	14.6	5.6	10.3
	VID-CS	18.1	9.2	0.4	17.2	19.1	19.5	9.6	18.3	18.2	17.2	14.7
DPM	VOC	29.6	10.1	17.1	55.0	18.4	24.7	11.3	57.7	47.8	44.5	31.6
	VMA	23.6	9.9	8.4	40.4	2.0	18.6	9.7	28.3	29.5	15.7	18.6
	VID-CG	17.4	9.3	9.2	35.7	9.4	9.7	3.3	16.2	27.3	15.0	15.2
	VID-CS	22.2	9.6	9.4	41.9	12.3	19.8	9.6	28.6	32.1	19.4	20.5

***Table 4.3**: Evaluation of different detectors and different training regimes on the* PASCAL07 *test set, in Average Precision computed as the area under the precision-recall curve. The mAP column reports the mean AP over all classes.*

VID-CG: models trained from YouTube-Objects on the output of our automatic selection method (sec. 4.3.3). Candidate tubes were obtained with the Class-Generic variant of sec. 4.3.2.

VID-CS: models trained from YouTube-Objects on the output of our automatic selection method (sec. 4.3.3). Candidate tubes were obtained with the Class-Specific variant of sec. 4.3.2.

Tab. 4.3 reports performance for the SPM as well as DPM detectors. The VOC training regime serves as a high quality reference. First we observe that the VID-CG model performs close to the manually annotated VMA, confirming the quality of our automatic learning approach. Moreover, compared to the VOC models, the VID-CG models offer reasonable performance, especially considering they take no manual intervention to train. As expected, the VID-CS models perform better than VID-CG thanks to a more precise localization of the target object (see tab. 4.2), but with the additional cost of pre-training class-specific models in a fully-supervised way.

Interestingly, the results confirm the domain shift discussed in sec. 4.5. There is a significant performance gap between VOC and VMA, although they are trained with a similar number of samples and level of supervision. In sec. 4.6.5 we will improve over these results by training jointly from both images and video with domain adaptation.

4.6.4 Comparison to WS learning from images [Pandey and Lazebnik 2011]

We compare our approach to a recent weakly supervised method for learning from images [Pandey and Lazebnik 2011]. It learns each model from all PASCAL07 training images corresponding to a class, given also its ground-truth average aspect-ratio. This

	aero	boat	horse	mbike	train	mAP
DPM-VID-CG	17.4	9.2	16.2	27.3	15.0	17.0
[Pandey and Lazebnik 2011]	11.5	3.0	20.3	9.1	13.2	11.4

Table 4.4: *Comparison of our approach to weakly supervised learning from images [Pandey and Lazebnik 2011].*

		aero	bird	boat	car	cat	cow	dog	horse	mbike	train
SPM	All(VOC,VID-CG)	1.7	−4.7	−8.4	−1.7	1.5	−2.2	0.2	−2.1	−2.2	−4.2
	Pred(VOC,VID-CG)	1.3	−4.6	0.3	−4.3	2.9	−0.3	−5.2	−1.6	−0.3	−0.1
	Prior(VOC,VID-CG)	−1.1	0.5	−1.1	0.6	−2.2	−5.8	−6.8	0.7	−1.3	−0.3
	LinInt(VOC,VID-CG)	2.9	0.0	0.2	0.6	0.0	0.0	0.0	0.0	1.9	0.0
DPM	All(VOC,VID-CG)	1.1	−0.4	−7.5	−2.7	−1.0	−9.2	−0.7	−9.0	−2.6	−12.2
	LinInt(VOC,VID-CG)	4.5	1.4	2.4	0.0	1.5	0.4	0.0	0.0	0.0	0.0
	LinInt(VOC,VID-CS)	5.4	1.9	1.8	0.9	3.4	1.3	0.5	0.0	0.2	4.6

Table 4.5: *We show relative improvement in Average-Precision wrt VOC models.*

information is required as their approach iteratively refines the DPM detector and does not converge if initialized with a square window. This is a little more supervision than used normally in weakly supervised learning, where only class labels for the image/video are given (as is our case).

Results [Pandey and Lazebnik 2011] are presented for a subset of 14 classes of PASCAL07. We obtained the learned DPM detectors [Felzenszwalb *et al.* 2009] from the authors and evaluated them on the test set. Tab. 4.4 presents results on the 5 classes that have been evaluated in both our work and theirs. Our models learned on videos outperform their models learned from images. We believe this result validates our approach as an effective alternative to weakly supervised learning from images.

4.6.5 Training from video and images

Here we train detectors from a combination of weakly supervised video data *and* PASCAL07 fully supervised images using domain adaptation. Tab. 4.5 presents results for various adaptation methods (sec. 4.5). We report differential results wrt learning only on the target domain (still images, VOC rows in tab. 4.3). We train the λ parameter of LinInt by maximizing performance on the Val set of PASCAL07.

The results show that the All method, which combines training data at the earliest stage, degrades performance (negative transfer [Pan and Yang 2010]). The problem is only partially alleviated when the combination happens at the feature level (Pred) or the parameter level (Prior). The LinInt method instead is immune to negative transfer and prevents a weaker model to harm the combined performance (by automatically setting λ to 0).

Figure 4.10: **LinInt qualitative improvements**. *We illustrate cases from the test set of* PASCAL07. *Each image shows the highest-scoring detection obtained with the respective model indicated at the top of the figure. We see how the VID-CG provides complementary knowledge that is incorporated successfully with LinInt. Even if the highest-scoring detections from both VOC and VID-CG are wrong (see fourth row) the highest-scoring combined detection is accurately placed on the object of interest.*

Moreover, for DPM in 5 out of 10 classes adding VID-CG knowledge with LinInt improves over VOC, demonstrating that knowledge can be transferred from the video to the image domain, leading to a *better detector* than one trained from images alone ($+2.0\%$ AP on average on the 5 classes where $\lambda > 0$). In fig. 4.10 we show qualitative examples of improved detections using LinInt. A combination of LinInt involving VID-CS models leads to additional improvement thanks to the higher accuracy of these models. In particular we notice how a higher number of classes (9 out of 10) benefit from additional knowledge from the video domain.

5

Conclusions

5.1 Weakly supervised learning of interactions between humans and objects in still images

In chapter 2 we introduced a novel approach for learning human-object interactions automatically from weakly labeled images and has been published in 2011 in [Prest *et al.* 2011].

Our approach automatically determines objects relevant for the action and their spatial relations to the human. The performance of our method is comparable to state-of-the-art fully supervised approaches [Gupta *et al.* 2009, Yao and Fei-Fei 2010b] on the Sport dataset of [Gupta *et al.* 2009]. Moreover, on the PASCAL Action Challenge 2010 [Everingham and others 2010], it outperforms the best contestant (Koniusz et al. [Everingham and others 2010]) on classes involving humans and objects.

5.1.1 Outlook

Possible extensions of this work include combining recent state-of-the-art methods for pose estimation [Yang and Ramanan 2011] to further enrich our human-object interaction model. Other action recognition works use pose information [Yao and Fei-Fei 2010b], but their approach learn action-specific pose models in a fully-supervised way, requiring a substantial amount of additional annotations (i.e. limbs location at training time). In a weakly-supervised setting instead we could utilize the recent method of [Yang and Ramanan 2011], trained from generic images, to provide a set of strong candidates for the human pose in each training image. For each human, this model generates thousands of pose hypothesis, each of which with a score reflecting how confident the model is about that particular configuration. Analogously to the spatial and object model (see sec. 2.5.1), we follow the intuition that pose should be similar in images of the same class. In this way the different pose hypothesis computed by [Yang and Ramanan 2011] could become

states in a CRF defined over all images of the same class. A pairwise potential (eq. 2.6) capturing the distance between poses could then drive the selection of recurrent class-specific poses. Moreover the confidence score of each pose would influence the selection as a unary term, penalizing pose configurations with low evidence from the pose estimation engine. Inference on the model [Kolmogorov 2006] would then select one pose per image such that the pose similarity is maximized, allowing to learn class-specific pose models without using any additional supervision.

Such an extended model would be beneficial to both the discriminative power of our model and to the accuracy of the action object localization. Concerning the latter, pose provides a strong cue as to where the action object could be located and, combined with our human-object spatial model, it could improve substantially the localization of the object at test time (see fig. 2.11, 2.12, 2.10).

This extended model could be powerful enough to tap into large-scale dataset such as ImageNet [Deng *et al.* 2009]. In particular the hierarchical structure of ImageNet could be exploited for learning human action grammars, allowing to learn new action classes by leveraging on their semantics. For example interactions such as "playing volleyball" and "playing football" are related as they share the same action object. This information could be of great use in modeling actions in a more powerful way and exploiting knowledge learned previously for other related actions.

5.2 Explicit modeling of human-object interactions in realistic videos

Chapter 3 introduced an approach for learning human-object interactions in videos and has been published in [Prest *et al.* 2012a]. It explicitly tracks both the human and the action-object and represents the interaction as the relative position and motion of the object wrt the human. Experimental results confirm that human-object interactions, when explicitly captured by our method, are a rich source of information for action recognition and localization in video. Furthermore, we show that the proposed interaction model captures information complementary to existing low-level descriptors. Moreover, when combining the two, our approach improves over the state-of-the-art [Kläser *et al.* 2010] on *Coffee & Cigarettes*, as well as on the Rochester Daily Activities dataset [Matikainen *et al.* 2010, Messing *et al.* 2009], and achieves the same results of [Gupta *et al.* 2009] on Gupta video, despite using substantially less supervision for training.

5.2.1 Outlook

A valuable future extension would consist in an automated way to localize action objects at training time without the need for manual annotations. Recent motion segmentation

methods such as [Brox and Malik 2010] allow to aggregate point tracks that move similarly into motion segments, providing spatio-temporal regions of consistent motion (see sec. 4.3). Applying this procedure on the positive training clips would lead to a set of candidate spatio-temporal regions likely to contain the action-object. While most of these motion segments will cover irrelevant motion happening in the background, some of them will capture the location of the action-object over time. In the spirit of chapter 2, it would be then possible to jointly select one motion segment per clip so that the overall motion similarity wrt to the human location is maximized across all positive training clips of the same action. In other words we would mine for similar and discriminative motion for a particular action, therefore obtaining a weakly-supervised localization of the action object. This information can be seen as an initialization for iteratively refining an action object model but also for building an interaction model as in sec. 3.5.

Another interesting direction would be to model the scene structure. By exploiting global movement and spatial relations between multiple people one could obtain a prior on which actions they might be about to do. The actual action is meant to be resolved by more sophisticated methods in a later processing stage. As an example, if two people are approaching each other the model should expect a human-human interactions where they might hug, kiss, punch,... but not ride a bicycle of make a phone call. This reasoning could be modeled as a function that inputs human tracks and outputs a multinomial distribution over the known verbs, representing the probability of each action. Such a model could be applied on a broad range of datasets including crowded scenes and surveillance cameras.

5.3 Learning Object Class Detectors from Weakly Annotated Video

In chapter 4, which is an extension of a previous work published in [Prest *et al.* 2012b], we demonstrated that learning an high-quality object detector from real-world videos collected from the web is possible.

We proposed a novel dataset comprising video sequences downloaded from YouTube and an auto-localization method of objects that allows for learning with a minimum amount of supervision. Additionally, we demonstrated that with our approach it is possible to get good detection performance on still images when training on videos only.

More in detail, we discussed two auto-localization variants for learning from video indicating that: (i) a Class-Generic approach is already able to learn object models from video using only the video label as manual input. Despite using drastically less supervision compared to models trained on static images, these models already obtain good performance on the challenging PASCAL07 dataset. (ii) a Class-Specific approach mak-

ing use of models pre-trained on the PASCAL07 dataset allows for better localization of object instances in video and leads to improved results.

Furthermore, we formulated the problem of learning from both videos and still images jointly as a domain adaptation task. On PASCAL07 this resulted in increased performance compared to training from images alone.

5.3.1 Outlook

We see this work as an important step towards learning visual models of categories from large-scale video sources. Future extension include improving localization in the Class-Generic case, which right now appears to be the bottleneck for building more accurate models (see fig. 4.9). Hence, it would be profitable to look into more advanced motion segmentation methods [Ochs and Brox 2011, Ochs and Brox 2012]. Motion could also become a discriminative feature in localizing the object of interest in video. One can imagine learning class-specific motion models, where the motion pattern of a horse will be very different from that of an aeroplane. Both global motion (i.e. trajectory of a motion segment wrt the image reference) and relative motion (i.e. motion patterns within the same motion segment) can become discriminative features for localizing objects in videos.

One limitation of the present work is that we only use a small subset (500) of the available frames extracted from videos to learn object models. This is due to the limitations of current method for learning object detectors: they become prohibitively slow when dealing with more than a couple of thousands positive training examples. It would be therefore profitable to intervene on the learning side and devise method that can handle a larger amount of unlabeled data to train an object detector. In this context a fast learning approach is needed, as an example a new detector based on locally-linear SVMs (LL-SVM) [Ladicky and Torr 2011]. LL-SVMs have practically the same prediction power as non-linear kernel-based learners but their speed, both at training and testing, is comparable to linear SVMs. Thus, they are perfect candidates to learn from both labeled data and large amounts of weakly-labeled data.

Another potential improvement on the learning side comes from the observation that the hinge-loss used is learning traditional object detectors [Felzenszwalb *et al.* 2009, Vedaldi *et al.* 2009] assumes correctly labeled samples. This is hard to achieve in our method since it is weakly-supervised. Even after having improved our pipeline substantially, noise will not be avoidable. To address this problem we propose two ideas: first, to use a non-convex loss functions in the SVM and, second, to penalize the learner when it predicts different labels in similar sub-regions from two successive frames. While the first solution should be able to considerably absorb noise, the second should help to improve the generalization power of the SVM by using the space-time-constraint given by the fact that we use videos.

Another high-potential improvement could come from domain adaptation techniques that operate at the feature level [Kulis *et al.* 2011] and learn a non-linear mapping between

features from different domains. This could lead to a more effective combination of video frames and static images as a training source, alleviating the difference in the feature space as shown in fig. 4.6.

Video has also great potential for multiple view models. Unlike static images, videos depict a single object from a wide range of viewpoints (see first row in fig. 4.3) and one could learn 3D models of the object using structure from motion [Crandall *et al.* 2011]. Moreover one could extract more detailed models that capture the structure of an object at parts-level. The motorbike in fig. 4.9 provides a good example: while representing a failure case in the current system, it inspires a new setting where an object is recognized as a collection of sub-parts in video (i.e. based on motion and appearance). This would provide a richer and more descriptive output and has the potential of being more discriminative compared to learning from bounding-boxes.

We conclude by pointing out that the YouTube-Objects dataset is available for download on our project page [1] together with additional videos showing intermediate processing stages of our approach. We hope this work will spark more intensive research in exploiting realistic videos for tasks, such as, detection, categorization, etc.

[1] http://groups.inf.ed.ac.uk/calvin/learnfromvideo

Bibliography

[Alexe *et al.* 2010] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, 2010. 2.5.1, 2.5.2, 2.5.3, 2.6.1, 2.7.3, 4.2, 4.3.3, 4.4.2

[Ali *et al.* 2011] K. Ali, D. Hasler, and F. Fleuret. Flowboost - appearance learning from sparsly labeled video. In *CVPR*, 2011. 1.3.2, 4.1, 4.2

[Arora *et al.* 2007] H. Arora, N. Loeff, D. Forsyth, and N. Ahuja. Unsupervised segmentation of objects using efficient learning. In *CVPR*, 2007. 1.3.2, 4.1, 4.2

[Aytar and Zisserman 2011] Y. Aytar and A. Zisserman. Tabula rasa: Model transfer for object category detection. In *ICCV*, 2011. 4.5.1

[Bagon *et al.* 2010] S. Bagon, O. Brostovski, M. Galun, and M. Irani. Detecting and sketching the common. In *CVPR*, 2010. 4.2

[Bay *et al.* 2008] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool. SURF: Speeded up robust features. *CVIU*, 110(3):346–359, 2008. 2.5.3, 2.5.4, 4.3.3, 4.4.2

[Blaschko *et al.* 2010] M. Blaschko, A. Vedaldi, and A. Zisserman. Simulatenous object detection and ranking with weak supervision. In *NIPS*, 2010. 1.3.2, 4.1, 4.2

[Bobick and Davis 2001] A. Bobick and J. Davis. The recognition of human movement using temporal templates. *PAMI*, 23(3):257–267, 2001. 1.3.1, 3.2

[Borenstein and Ullman 2004] E. Borenstein and S. Ullman. Learning to segment. In *ECCV*, 2004. 1.3.2, 4.2

[Bosch *et al.* 2007] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *CIVR*, 2007. 4.3.3

[Botev 2007] Z. Botev. Nonparametric density estimation via diffusion mixing. *The University of Queensland, Postgraduate Series, Nov*, 2007. 2.5.4

[Breitenstein *et al.* 2009] M. Breitenstein, F. Reichlin, and L. V. Gool. Robust tracking-by-detection using a detector confidence particle filter. In *ICCV*, 2009. 3.4.2

[Brox and Malik 2010] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, 2010. 4.3, 4.3.2, 5.2.1

[Brox and Malik 2011] T. Brox and J. Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *PAMI*, 2011. 3.4.2

[C. Desai 2010] C. F. C. Desai, D. Ramanan. Discriminative models for static human-object interactions. In *Workshop on Structued Models in Computer Vision, Computer Vision and Pattern Recognition (SMiCV) in Conjunction with CVPR*, 2010. 2.2

[Cao and Li 2007] L. Cao and F.-F. Li. Spatially coherent latent topic model for concurrent segmentation and classification of objects and scene. In *ICCV*, 2007. 1.3.2, 4.2

[Chen *et al.* 2010] Y. Chen, L. Zhu, and A. Yuille. Active mask hierarchies for object detection. In *ECCV*, 2010. 1.2

[Chum and Zisserman 2007] O. Chum and A. Zisserman. An exemplar model for learning object classes. In *CVPR*, 2007. 1.3.2, 4.1, 4.2

[Comaniciu *et al.* 2001] D. Comaniciu, V. Ramesh, and P. Meer. The variable bandwidth mean shift and data-driven scale selection. In *ICCV*, pages 438–445, 2001. 2.4.3

[Crandall and Huttenlocher 2006] D. J. Crandall and D. Huttenlocher. Weakly supervised learning of part-based spatial models for visual object recognition. In *ECCV*, 2006. 1.3.2, 4.2

[Crandall *et al.* 2011] D. J. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *CVPR*, 2011. 5.3.1

[Dalal and Triggs 2005] N. Dalal and B. Triggs. Histogram of oriented gradients for human detection. In *CVPR*, 2005. 1.2, 3.4.1, 4.3.3, 4.4.2

[Daume III 2007] H. Daume III. Frustratingly easy domain adaption. In *ICML*, 2007. 4.5.1

[Deng *et al.* 2009] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 5.1.1

[Desai *et al.* 2009] C. Desai, D. Ramanan, and C. Folkess. Discriminative models for multi-class object layout. In *ICCV*, 2009. 2.2

[Deselaers *et al.* 2010] T. Deselaers, B. Alexe, and V. Ferrari. Localizing objects while learning their appearance. In *ECCV*, 2010. 1.2, 1.3.2, 2.3.1, 4.1, 4.2, 4.3.3, 4.6.2

[Dollar *et al.* 2005] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *ICCV VS-PETS*, 2005. 1.3.1, 2.2, 3.2

[Duchenne *et al.* 2009] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of humans actions in video. In *ICCV*, 2009. 1.3.1, 3.2

[Efros *et al.* 2003] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV*, 2003. 1.3.1, 3.2

[Eichner and Ferrari 2009] M. Eichner and V. Ferrari. Better appearance models for pictorial structures. In *BMVC*, 2009. 2.4.5

[Everingham and others 2010] M. Everingham *et al.* The PASCAL Visual Object Classes Challenge 2010 Results, 2010. 1.2, 1.3.2, 2.1, 2.9, 2.2, 2.14, 2.8, 2.8, 2.9, 5.1

[Everingham *et al.*] M. Everingham, L. van Gool, C. Williams, and A. Zisserman. The PASCAL Visual Object Classes Challenge (VOC). 1

[Everingham *et al.* 2007a] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html, 2007. 2.4.1, 2.4.5, 3.4.1, 3.7.1

[Everingham *et al.* 2007b] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 Results, 2007. 4.1, 4.4.2, 4.5, 4.6.3

[Everingham *et al.* 2008] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html, 2008. 2.4.5

[Felzenszwalb *et al.* 2009] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 2009. 1.3.2, 2.4, 2.4.1, 2.4.5, 2.7.2, 2.7.3, 3.1, 3.3.1, 3.4.1, 3.7.1, 4.1, 4.2, 4.3.2, 4.4, 4.4.1, 4.6, 4.4.2, 4.5.2, 4.6.4, 5.3.1

[Felzenszwalb *et al.* 2010] P. F. Felzenszwalb, R. B. Girshick, and D. Mcallester. D.m.: Cascade object detection with deformable part models. In *CVPR*, 2010. 4.3.2

[Fergus and Perona 2003] R. Fergus and P. Perona. Caltech object category datasets. http://www.vision.caltech.edu/html-files/archive.html, 2003. 2.4.5, 3.4.1

[Fergus *et al.* 2003] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003. 1.2, 1.3.2, 2.3.1, 4.2

[Ferrari *et al.* 2008] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *CVPR*, 2008. 2.4.1, 3.4.2, 3.4.2

[Filipovych and Ribeiro 2008] R. Filipovych and E. Ribeiro. Recognizing primitive interactions by exploring actor-object states. In *CVPR*, 2008. 1.3.1, 3.2

[Filipovych and Ribeiro 2010] R. Filipovych and E. Ribeiro. Robust sequence alignment for Actor-Object interaction recognition: Discovering Actor-Object states. *CVIU*, 2010. 1.3.1, 3.2

[Gaidon *et al.* 2011] A. Gaidon, Z. Harchaoui, and C. Schmid. Actom sequence models for efficient action detection. In *CVPR*, 2011. 1.3.1, 3.2

[Galleguillos *et al.* 2008] C. Galleguillos, B. Babenko, A. Rabinovich, and S. Belongie. Weakly supervised object localization with stable segmentations. In *ECCV*, 2008. 4.2

[Gehler and Nowozin 2009] P. V. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *ICCV*, 2009. 2.6.4

[Gorelick *et al.* 2007] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *PAMI*, 2007. 1.1, 1.3.1, 3.2

[Grabner and Bischof 2006] H. Grabner and H. Bischof. On-line boosting and vision. In *CVPR*, 2006. 3.7.1, 4.2

[Grabner *et al.* 2008] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV*, 2008. 3.4.2, 3.7.1

[Grubinger *et al.* 2006] M. Grubinger, P. D. Clough, H. Müller, and T. Deselaers. The iapr benchmark: A new evaluation resource for visual information systems. In *International Conference on Language Resources and Evaluation*, Genoa, Italy, 24/05/2006 2006. 2.7.1

[Gupta *et al.* 2008] A. Gupta, T. Chen, F. Chen, D. Kimber, and L. Davis. Context and observation driven latent variable model for human pose estimation. In *CVPR*, 2008. 2.2

[Gupta *et al.* 2009] A. Gupta, A. Kembhavi, and L. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. In *PAMI*, 2009. (document), 1.1, 1.2, 1.3, 1.3.1, 2.1, 2.2, 2.6.2, 2.6.3, 2.6.4, 2.9, 2.7, 2.11, 2.7.1, 2.7.1, 2.7.2, 2.3, 2.7.3, 3.1, 3.2, 3.5.2, 3.7, 3.7.2, 3.7.2, 5.1, 5.2

[Harzallah *et al.* 2009] H. Harzallah, F. Jurie, and C. Schmid. Combining efficient object localization and image classification. In *ICCV*, 2009. 4.4, 4.4.1

[Heusch *et al.* 2006] G. Heusch, Y. Rodriguez, and S. Marcel. Local binary patterns as an image preprocessing for face authentication. In *Automatic Face and Gesture Recognition*, 2006. 2.4.1

[Ikizler and Forsyth 2008] N. Ikizler and D. A. Forsyth. Searching for complex human activities with no visual examples. *IJCV*, 2008. 1.3.1, 3.2

[Ikizler *et al.* 2008] N. Ikizler, R. G. Cinbis, S. Pehlivan, and P. Duygulu. Recognizing actions from still images. In *ICPR*, 2008. 2.2

[Ikizler-Cinbis and Sclaroff 2010] N. Ikizler-Cinbis and S. Sclaroff. Object, scene and actions: Combining multiple features for human action recognition. In *ECCV*, 2010. 2.2

[Ikizler-Cinbis *et al.* 2009] N. Ikizler-Cinbis, G. Cinbis, and S. , Sclaroff. Learning actions from the web. In *ICCV*, 2009. 1.3.1, 2.2, 3.2

[Johansson and Nugues 2008] R. Johansson and P. Nugues. Dependency-based syntactic-semantic analysis with propbank and nombank. In *CoNLL '08: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 183–187, Morristown, NJ, USA, 2008. Association for Computational Linguistics. 2.7.1

[Kalal *et al.* 2010] Y. Kalal, J. Matas, and K. Mikolajcyzk. P-N learning: Bootstrapping binary classifiers from unlabeled data by structural constraints. In *CVPR*, 2010. 3.7.1, 4.2

[Kim and Kim 2009] W.-H. Kim and J.-N. Kim. An adaptive shot change detection algorithm using an average of absolute difference histogram within extension sliding window. In *ISCE*, 2009. 4.3.1

[Kim and Torralba 2009] G. Kim and A. Torralba. Unsupervised detection of regions of interest using iterative link analysis. In *NIPS*, 2009. 1.3.2, 4.2

[Kläser *et al.* 2010] 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. 1.3.1, 3.1, 3.2, 3.3.1, 3.4.1, 3.4.2, 3.4.2, 3.6, 3.7.1, 3.7.1, 3.7.1, 3.7.1, 3.7.1, 2, 5.2

[Kolmogorov 2006] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 28(10):1568–1583, 2006. 2.5.1, 4.3.3, 5.1.1

[Kulis *et al.* 2011] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*, 2011. 5.3.1

[Ladicky and Torr 2011] L. Ladicky and P. H. S. Torr. Locally linear support vector machines. In *ICML*, 2011. 5.3.1

[Laptev and Perez 2007] I. Laptev and P. Perez. Retrieving actions in movies. In *ICCV*, 2007. 1.1, 1.3, 1.3.1, 2.2, 3.1, 3.2, 3.6, 3.7, 3.7.1, 3.7.1, 3.7.1, 3.7.1, 3.3

[Laptev *et al.* 2008] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 1.3.1, 2.2, 3.2

[Lazebnik *et al.* 2006] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 4.3.3, 4.4.2

[Lee and Grauman 2009] Y. J. Lee and K. Grauman. Shape discovery from unlabeled image collections. In *CVPR*, 2009. 4.2

[Lee and Grauman 2011] Y. J. Lee and K. Grauman. Learning the easy things first: Self-paced visual category discovery. In *CVPR*, 2011. 4.1, 4.2

[Leistner *et al.* 2011] C. Leistner, M. Godec, S. Schulter, A. Saffari, and H. Bischof. Improving classifiers with weakly-related videos. In *CVPR*, 2011. 1.3.2, 4.1, 4.2

[Li and Fei-Fei 2007] L.-J. Li and L. Fei-Fei. What, where and who? classifying event by scene and object recognition. In *ICCV*, 2007. 2.6.2

[Liu *et al.* 2009] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos in the wild. In *CVPR*, 2009. 1.3.1, 3.2

[Lowe 1999] D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, Sep 1999. 1.2

[Maji *et al.* 2008] S. Maji, A. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR*, 2008. 3.6, 4.4.2

[Matikainen *et al.* 2010] P. Matikainen, M. Hebert, and R. Sukthankar. Representing pairwise spatial and temporal relations for action recognition. In *ECCV*, 2010. 1.3.1, 3.1, 3.2, 3.7.3, 3.7.3, 5.2

[Messing *et al.* 2009] R. Messing, C. Pal, and H. Kautz. Activity recognition using the velocity histories of tracked keypoints. In *ICCV*, 2009. 1.1, 1.3.1, 3.1, 3.2, 3.7, 3.7.3, 3.7.3, 5.2

[Mikolajczyk and Uemura 2008] K. Mikolajczyk and H. Uemura. Action recognition with motion-appearance vocabulary forest. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008. 1.3.1, 2.2, 3.2

[Nguyen *et al.* 2009] M. H. Nguyen, L. Torresani, F. de la Torre, and C. Rother. Weakly supervised discriminative localization and classification: a joint learning process. In *ICCV*, 2009. 4.2

[Niebles *et al.* 2010] J. C. Niebles, C.-W. Chen, , and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010. 1.3.1, 3.2

[Ochs and Brox 2011] P. Ochs and T. Brox. Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions. In *ICCV*, 2011. 5.3.1

[Ochs and Brox 2012] P. Ochs and T. Brox. Higher order motion models and spectral clustering. In *CVPR*, 2012. 5.3.1

[Oliva and Torralba 2001] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001. 2.6.2, 2.6.3, 4.5

[Ommer *et al.* 2009] B. Ommer, T. Mader, and J. M. Buhmann. Seeing the objects behind the dots: Recognition in videos from a moving camera. *IJCV*, 83(1):57–71, 2009. 1.3.2, 4.2

[Pan and Yang 2010] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. on Knowledge and Data Engineering*, 2010. 4.5.1, 4.6.5

[Pandey and Lazebnik 2011] M. Pandey and S. Lazebnik. Scene recognition and weakly-supervised object localization with deformable part-based models. In *ICCV*, 2011. (document), 1.3.2, 4.1, 4.2, 4.5.2, 4.6.2, 4.6.4, 4.4, 4.6.4

[Prest *et al.* 2011] A. Prest, C. Schmid, and V. Ferrari. Weakly supervised learning of interactions between humans and objects. *TPAMI (accepted for publication, to appear)*, 2011. 1.1, 3.1, 3.3.1, 3.4.1, 3.5.2, 3.7.1, 5.1

[Prest *et al.* 2012a] A. Prest, V. Ferrari, and C. Schmid. Explicit modeling of human-object interactions in realistic videos. *TPAMI (accepted for publication, to appear)*, 2012. 5.2

[Prest *et al.* 2012b] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012. 5.3

[Ramanan *et al.* 2006] D. Ramanan, D. A. Forsyth, and K. Barnard. Building models of animals from video. *PAMI*, 2006. 1.3.2, 4.1, 4.2

[Ramanan *et al.* 2007] D. Ramanan, D. A. Forsyth, and A. Zisserman. Tracking people by learning their appearance. *PAMI*, 29(1):65–81, Jan 2007. 3.4.2

[Rodriguez *et al.* 2008] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach: a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008. 1.3.1, 3.2

[Rodriguez 2006] Y. Rodriguez. *Face Detection and Verification using Local Binary Patterns*. PhD thesis, EPF Lausanne, 2006. 2.4.1

[Russell *et al.* 2006] B. Russell, A. Efros, J. Sivic, W. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006. 4.2

[Satkin and Hebert 2010] S. Satkin and M. Hebert. Modeling the temporal extent of actions. In *ECCV*, 2010. 1.3.1, 3.2

[Schuldt *et al.* 2004] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. *Pattern Recognition, International Conference on*, 3:32–36, 2004. 1.1, 1.3, 1.3.1, 2.2, 3.2

[Sharma *et al.* 2012] G. Sharma, F. Jurie, and C. Schmid. Discriminative spatial saliency for image classification. In *CVPR*, 2012. 2.9, 2.4, 2.9

[Siva and Xiang 2011] P. Siva and T. Xiang. Weakly supervised object detector learning with model drift detection. In *ICCV*, 2011. 1.3.2, 4.2

[Sivic *et al.* 2005] J. Sivic, M. Everingham, and A. Zisserman. Person spotting: video shot retrieval for face sets. In *CIVR*, 2005. 3.4.2

[Sivic *et al.* 2009] J. Sivic, M. Everingham, and A. Zisserman. "Who are you?" – Learning person specific classifiers from video. In *CVPR*, 2009. 3.4.2

[Sullivan and Carlsson 2002] J. Sullivan and S. Carlsson. Recognizing and tracking human action. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 629–644, London, UK, 2002. Springer-Verlag. 2.2

[Sundaram *et al.* 2010] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by GPU-accelerated large displacement optical flow. In *ECCV*, 2010. 1.5, 1.3.2, 3.1, 3.4.2, 1, 3.4.2, 3.7.1

[Thurau and Hlavac 2008] C. Thurau and V. Hlavac. Pose primitive based human action recognition in videos or still images. In *CVPR*, 2008. 2.2

[Todorovic and Ahuja 2006] S. Todorovic and N. Ahuja. Extracting subimages of an unknown category from a set of images. In *CVPR*, 2006. 1.2, 4.2

[Tommasi *et al.* 2010] T. Tommasi, F. Orabona, and B. Caputo. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *CVPR*, 2010. 4.5.1

[Torralba and Efros 2011] A. Torralba and A. Efros. An unbiased look on dataset bias. In *CVPR*, 2011. 4.5

[Tsochantaridis *et al.* 2005] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6:1453–1484, 2005. 4.3.3

[Vedaldi *et al.* 2009] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009. 1.2, 1.3.2, 4.1, 4.2, 4.4, 5.3.1

[Vijayanarasimhan and Grauman 2011] S. Vijayanarasimhan and K. Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. In *CVPR*, 2011. 1.3.2, 4.1

[Viola and Jones 2001] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, pages 511–518, 2001. 1.3.2, 2.4.1, 4.1, 4.2

[Wang *et al.* 2006] Y. Wang, H. Jiang, M. S. Drew, Z. nian Li, and G. Mori. Unsupervised discovery of action classes. In *CVPR*, 2006. 2.2

[Willems *et al.* 2009] G. Willems, J. H. Becker, T. Tuytelaars, and L. van Gool. Exemplar-based action recognition in video. In *BMVC*, 2009. 2.2, 3.6, 3.7.1, 3.3, 3.7.1

[Winn and Jojic 2005] J. Winn and N. Jojic. LOCUS: learning object classes with unsupervised segmentation. In *ICCV*, 2005. 1.3.2, 4.1, 4.2

[Winn *et al.* 2005] J. Winn, Criminisi, A., and T. Minka. Object categorization by learned universal visual dictionary. *ICCV*, 2005. 2.3.1

[Wu *et al.* 2008] Z. Wu, M. Betke, J. Wang, and V. Athitsos. Tracking with dynamic hidden-state shape models. In *ECCV*, 2008. 3.4.2

[Wu *et al.* 2010] S. Wu, B. E. Moore, and M. Shah. Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In *CVPR*, 2010. 3.2

[Yang and Ramanan 2011] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011. 5.1.1

[Yang *et al.* 2005] C. Yang, R. Duraiswami, and L. Davis. Efficient mean-shift tracking via a new similarity measure. In *CVPR*, 2005. 3.4.2

[Yang *et al.* 2010] W. Yang, Y. Wang, and G. Mori. Recognizing human actions from still images with latent poses. In *CVPR*, 2010. 2.2

[Yao and Fei-Fei 2010a] B. Yao and L. Fei-Fei. Grouplet: A structured image representation for recognizing human and object interactions. In *CVPR*, 2010. 1.3.1, 2.2, 2.9, 3.5.2

[Yao and Fei-Fei 2010b] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, 2010. 1.3, 1.3.1, 2.1, 2.2, 2.6.3, 2.6.4, 2.7, 2.7.1, 2.7.2, 2.3, 2.7.3, 3.5.2, 5.1, 5.1.1

[Yao *et al.* 2011a] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. J. Guibas, and L. Fei-Fei. Action recognition by learning bases of action attributes and parts. In *ICCV*, 2011. 1.2, 2.2

[Yao *et al.* 2011b] B. Yao, A. Khosla, and L. Fei-Fei. Combining randomization and discrimination for fine-grained image categorization. In *CVPR*, 2011. 1.2

[Yilmaz and Shah 2005] A. Yilmaz and M. Shah. Actions sketch: a novel action representation. In *CVPR*, 2005. 1.3.1, 3.2

[Zanetti *et al.* 2008] S. Zanetti, L. Zelnik-Manor, and P. Perona. A walk through the web's video clips. In *CVPRW*, 2008. 4.2, 4.5

[Zelnik-Manor and Irani 2001] L. Zelnik-Manor and M. Irani. Event-based analysis of video. In *CVPR*, 2001. 1.3.1, 3.2

[Zhang *et al.* 2007] J. Zhang, M. Marszalek, S. Lazebnik, and S. C. Local features and kernels for classification of texture and object categories: a comprehensive study. *IJCV*, 2007. 2.5.4

[Zhang *et al.* 2011] J. Zhang, K. Huang, Y. Yu, and T. Tan. Boosted local structured hog-lbp for object localization. In *CVPR*, 2011. 1.2

Curriculum Vitae

Personal Data

Name	Alessandro Prest
Date of birth	20^{th} June 1983
Place of birth	Belluno, Italy
Citizenship	Italian

Education

2009 – 2012	*ETH Zurich, Computer Vision Laboratory, Switzerland*
	Doctoral studies
2002 – 2007	*University of Udine, Italy*
	Studies of Computer Science and Information Technology
	Graduation cum laude
1999 – 2002	*Istituto Tecnico Commerciale P.F. Calvi, Belluno, Italy*

Work Experience

2009 – 2012	*ETH Zurich, Computer Vision Laboratory, Switzerland*
	Research assistant
2008 – 2009	*University of Udine, Italy*
	Research fellow
2006 – 2008	*Isomorph srl, Italy*
	Software architect
2004 – 2006	*Isomorph srl, Italy*
	Programmer

Awards

2008 *Best applied physics work*, Italian Physical Society, Genova, Italy