Characterizing Objects in Images using Human Context

Dissertation

zur

Erlangung des Doktorgrades (*Dr. rer. nat.*)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich–Wilhelms–Universität, Bonn

vorgelegt von

Abhilash SRIKANTHA

aus

Bengaluru, Indien

Bonn 2016

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich–Wilhelms–Universität Bonn

1. Gutachter: Prof. Dr. Juergen Gall
2. Gutachter: Prof. Dr. Bastian Leibe
Tag der Promotion: 16.06.2017
Erscheinungsjahr: 2017

Abstract

by Abhilash Srikantha

for the degree of

Doctor rerum naturalium

Humans have an unmatched capability of interpreting detailed information about existent objects by just looking at an image. Particularly, they can effortlessly perform the following tasks: 1) Localizing various objects in the image and 2) Assigning functionalities to the parts of localized objects. This dissertation addresses the problem of aiding vision systems accomplish these two goals.

The first part of the dissertation concerns object detection in a Hough-based framework. To this end, the independence assumption between features is addressed by grouping them in a local neighborhood. We study the complementary nature of individual and grouped features and combine them to achieve improved performance. Further, we consider the challenging case of detecting small and medium sized household objects under human-object interactions. We first evaluate appearance based star and tree models. While the tree model is slightly better, appearance based methods continue to suffer due to deficiencies caused by human interactions. To this end, we successfully incorporate automatically extracted human pose as a form of context for object detection.

The second part of the dissertation addresses the tedious process of manually annotating objects to train fully supervised detectors. We observe that videos of human-object interactions with activity labels can serve as weakly annotated examples of household objects. Since such objects cannot be localized only through appearance or motion, we propose a framework that includes human centric functionality to retrieve the common object. Designed to maximize data utility by detecting multiple instances of an object per video, the framework achieves performance comparable to its fully supervised counterpart.

The final part of the dissertation concerns localizing functional regions or affordances within objects by casting the problem as that of semantic image segmentation. To this end, we introduce a dataset involving human-object interactions with strong *i.e.* pixel level and weak *i.e.* clickpoint and image level affordance annotations. We propose a framework that utilizes both forms of weak labels and demonstrate that efforts for weak annotation can be further optimized using human context.

Keywords: Object detection, human pose, context, weak supervision, affordance, semantic segmentation

Acknowledgements

Foremost, I thank my advisor Juergen Gall whose earnest and skillful approach towards research has been a source of confidence, encouragement and inspiration over the past few years. Juergen's guidance in conceiving the big picture, proficiency in resolving minute details and passion for diverse empirical evaluations has shaped my lifestyle in research. Also, his approachable and inclusive nature has given me a rare opportunity to support establishing a new group, here in Bonn.

I am also thankful to the review committee for their interest in this work and taking time to evaluate the dissertation, specially to Bastian Leibe for the detailed feedback.

I also want to thank Michael Black for his support. His artfully presented insights about important but subtle aspects of research have been immensely helpful. Thanks also to colleagues from Tuebingen and Bonn who are ever enthusiastic about brainstorming and discussing papers. I am particulary grateful to Dimitrios Tzionas, Umar Iqbal, Martin Garbade, Alexander Richard, Johann Sawatzky, Aura Munoz and Varun Jampani for their collaboration, discussions, troubleshooting, technical assistance and feedback.

My heartfelt thanks to administative staff at both sites. Special thanks to Melanie Feldhofer for her excellent support and making my transition into a relatively new country memorable. Many thanks to the cluster computing facilities in Aachen and Tuebingen. Experiments presented in this dissertation wouldn't have been possible without your prompt and diligent support.

I also want to thank my teachers from school: T. V. Uma, K. Jayashree, Uday K. Thakur, Jayamathy Sharma, Mala R. Denny, K. S. Shrish, Leena Mary, Alka Sehgal, Deepak Rajendraprasad, Lillykutty Jacob, Desire Sidibe and Markus Harz. Thank you for making learning an enjoyable experience.

I thank my parents Savitha and Srikantha for constantly supporting me through online presence. And, to Priyanka for backing me with genuine companionship and giving me confidence. I very much enjoy her curiosity and help with data collection and annotation tools, and with machine learning in general.

This research has been supported by a scholarship from the Max Planck society and a grant from the DFG Emmy Noether program.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

2d/3d	Two-dimensional/Three-dimensional
AP	Average precision
AUC	Area under the curve
BOW	Bag of words
BP	Belief propagation
CRF	Conditional random field
CNN	Convolutional neural network
DCNN	Deep convolutional neural network
DPM	Deformable part model
DTW	Dynamic time warping
EM	Expectation maximization
FPPI	False positives per image
GMM	Gaussian mixture model
HMM	Hidden markov model
HMP	Hierarchical matching pursuit
HOG	Histogram of gradients
HOL	Histogram of leaves
IOU	Intersection over union
ISM	Implicit shape model
LbP	Local binary pattern
LBP	Loopy belief propagation
MRF	Markov random field
PHOG	Pyramid HOG
PS	Pictorial structure
px	Pixel
RBF	Radial basis function
RGBD	RGB-Depth
SGD	Stochastic gradient descent
SIFT	Scale invariant feature transform
SRF	Structured random forest
SVM	Support vector machine

Frequently Used Symbols

$\setminus$	Set subtraction operation
$\{P_i\}$	Set of entities P_i
$\mathcal{C}$	Discrete label space
D_c^L	Set of displacements $\{\mathbf{d}_i\}$ of class c in leaf L

$\mathbf{d}_i$	Displacement from center of patch to centroid of object
$\mathcal{E}$	Edges of graph $\mathcal{G}$
f	Feature channel index
$\mathcal{G}$	Graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$
$\mathcal{H}$	Hough space, $\mathcal{H} \subseteq \mathbb{R}^H$
$\mathbf{h}$	Hypothesis in Hough space, $\mathbf{h} \in \mathcal{H}$
$\mathbf{I}$	Image
$\mathcal{I}_i$	Features associated with patch $\mathcal{P}_i$
$\mathcal{K}$	Keypoints configuration
K	Gaussian kernel
λ	Parameter for linear combination
M_{uv}	Message along the directed edge $(u \rightarrow v)$
Ω	Set of all pixel locations in an image, $\Omega \subseteq \mathbb{R}^2$
$\phi(\cdot)$	Unary potential
$\psi(\cdot, \cdot)$	Pairwise potential
$\mathcal{P}_i$	Image patch, $\mathcal{P}_i = (\mathcal{I}_i, c_i, \mathbf{d}_i)$
$p\left(c\|L\right)$	Class probability distribution in leaf L
$\mathbb{R}^D$	D-dimensional Real space
s, s_u	Scale, unit scale
$\mathcal{T}$	Set of trees $\{T_1, \ldots, T_{\|\mathcal{T}\|}\}$
θ	Parameters
t	Binary test for non-leaf node
$U(\cdot)$	Function to measure uncertainty
$(u \rightarrow v)$	Directed edge in $\mathcal{E}$
$\mathcal{V}$	Vertices of graph $\mathcal{G}$

CHAPTER 1

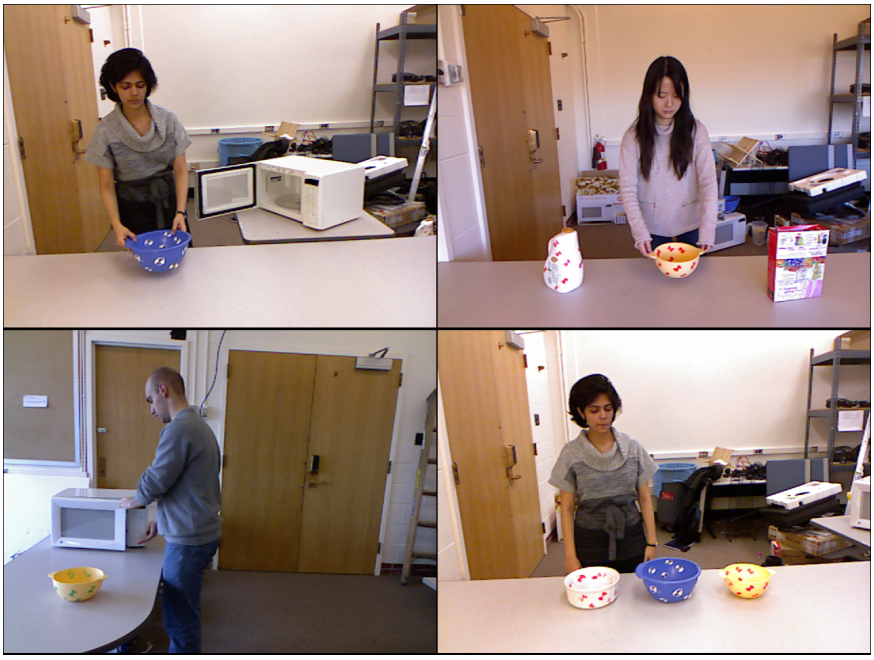
Introduction

Contents

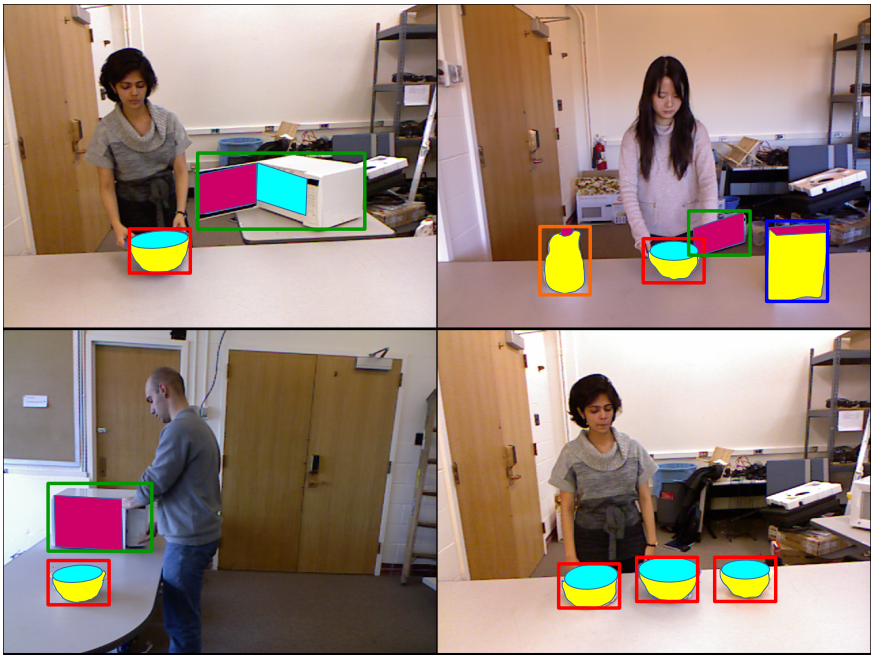
1.1. MOTIVATION

Breakthroughs in emerging sectors such as energy, computing, materials science and robotics have created an infrastructure that collects an unprecedented amount of data from billions of people. Artificial intelligence enables extracting information from this bulk of unstructured knowledge thereby shaping our approach to consumption: from media recommendations to connecting professionals, from language translation to virtual assistants and from manufacturing cars to ordering a taxi. Given the scope and rate at which it influences contemporary social and economic factors, we are expecting to enter a new phase of industrial revolution [1]. A precondition to this convergence between the physical and digital worlds is for automated systems to be able to deduce high level semantic information from visual data, a goal that is shared with computer vision.

As examples of such applications, robots can assist humans in performing routine chores like in the 1985–89 television series *Small Wonder* or virtual reality can make games like *Pokémon Go* more indulging and interactive or self driving vehicles can ease navigation. In such scenarios, it is imperative for computers to be able to assess images like in Fig 1.1(a) and answer the following questions: *What objects* are present? *Where* are they located? *What functionalities* do object parts serve? The goal of this dissertation is to develop data-driven techniques that help realize this goal.



(a)



(b)

Fig. 1.1: Object detection at work. (a) Images illustrating humans performing daily activities. (b) Ideal outputs of the system: drawing tight bounding boxes around objects and labeling object part functionalities. See text for details.

Assuming that the input is an RGB or RGBD image or video stream, we formulate the above problems as a set of well-defined simpler tasks. They are: a) Mark each instance of a predefined object by a tight bounding box around its visual extent. b) Label regions within each bounding box by their functionality or affordance. Following the popular convention of [2], the problems are formulated as *category level object detection* and *semantic segmentation* tasks respectively. Fig 1.1(b) shows the ground truth responses for these tasks. Here, all instances of objects *bowl*, *microwave* and *box* are bounded in red, green and blue respectively and functional regions within these objects are highlighted in cyan for *containable*, magenta for *openable* and yellow for *holdable*.

Significant progress has been achieved in object detection in the recent past. In categories where objects cover large image regions, the problem is well addressed. For instance, Average precision (AP) for large objects (size above 96^2 px) using a baseline method [3] is 0.369, whereas the performance for medium (size between 36^2 and 96^2 px) and small objects (size below 36^2 px) is 0.264 and 0.072, respectively, leaving a large scope for improvement. This leads us to ponder over why this problem is so challenging.

1.2. CHALLENGES

Inverse problems such as computer vision are required to cope with many challenges. This is because such systems are purported to deduce accurate inferences from noisy and lossy 2d projections of a 3d scene. In case of object detection, class membership must be interpreted despite inherent inconsistency while digitizing visual concepts into language representations. Apart from this semantic fuzziness, change in viewpoint, deformation, illumination, occlusion and motion results in high intra-class variations. The interplay between these factors produces a rich diversity in visual appearance making object detection challenging. These factors are illustrated in Fig 1.2 and are described as follows.

Discrete category labels: While humans have an unmatched capability of associating objects and categories, the underlying mechanisms are not clearly understood. It is in fact unclear if categorization is a suitable representation for computer vision [4, Chapter 1]. E.g., categorizations emerge from languages which are in turn continuously evolving, giving rise to visual polysemy [5] where visually disparate instances are grouped together *e.g. mouse* pertains to both rodents and computer accessories. Another source of disparity is due to categorization based on functionality *e.g.* a *mobile phone* and a traditional *phone* might not look similar but cater the same purpose. Also, subtle semantic differences contribute to inconsistency across examples of the same class *e.g.* between *house* and *building*.

Intra-Class Variability: Generic object detection tasks are required to handle variation in appearance intrinsic to the object category. For instance, variations in color and shape can cause significant variability within *mugs* and variations in texture and number of parts can alter the appearance of *chairs* drastically. The problem is intensified for functionality driven definitions that encapsulate a variety of objects resulting in a specific action *e.g. whisker* for "beating an egg".

Articulation and Viewpoint: Articulated objects or part wise rigid objects can cause

Fig. 1.2: Why is object detection challenging? (a) discrete category labels merging bathing and coffee mugs together, (b) intra-class variations within coffee mugs, (c) viewpoint variations and articulation, (d) illumination, (e) background clutter, (f) occlusion, (g) alternative functionalities and (h) motion blur. Images courtesy of websites: magicemart, qualitylogoproducts, toxel, ikea, wayfair, foodspotting, alicdn, financialexpress, tinydeal, terapeak, designsponge, herpeculiarlife, thisiswhyimbroke, thisiswhyimbroke, wordpress, craftychica, netdna-cdn.

substantial variation in appearance. Many daily life objects *e.g.* *microwave*, *scissors* etc. fall into this category. Also, large relative changes in the object-camera viewpoint result in substantial changes in 2d shape and appearance for many objects, *e.g.*, front and side views of *car*, *bicycle*, etc. However, the advantage is that starkly different viewpoints can provide information about parts hidden in other views but are visually highly diverse. On the other hand, small object distortions or viewpoint variations can be approximated as deformations of a template.

Motion Blur: This is a common artifact which is introduced due to digitized temporal domain. This occurs because relative motion of objects involved is far too high in comparison with the rate of image capture. The blurring obscures perceived appearance and position of the object resulting in more challenging detection scenarios. Typically, deconvolution techniques and/or optical flow are employed to either improve image quality or pool information from neighboring frames, respectively.

Illumination: Variation in object appearance can also be caused by extrinsic factors such as

illumination, *i.e.* the position and the number of light sources. This has an immediate impact on brightness, contrast and hue of the 3d scene captured in an image. Varying illumination also causes shadows, reflections and transparencies in the 3d scene which can create artifacts, *e.g.* false boundaries, double reflections, etc. in an image.

Truncation and Occlusion: Understanding from image data must also handle truncation when the object lies partially within the image's field of view. This results in incomplete image evidence frequently dooming object detection based on low and mid level image representations. Partial loss of image evidence also occurs due to occlusion when parts of objects are obscured by themselves or others. Occlusion reasoning can either be data driven, *i.e.* by learning occlusion patterns as features or driven by geometry, where depth ordered reasoning using additional imaging modalities can be useful.

Background clutter: Most real world images contain objects in their natural environments. In some cases, background can implicitly improve the confidence of object sighting, *e.g. sheep on grass* or can explicitly guide object localization, *e.g. mouse* detection can gain from prior localization of *monitor*, *keyboard*, etc. In other cases, background can be highly unstructured resulting in low level and scale space ambiguities thereby making object detection challenging.

1.3. PRIOR WORK

The tryst with object detection started with *Project Mac*, a summer assignment in MIT in 1966. In the past decades, basic principles of addressing this problem have evolved in tandem with those in related areas, *e.g.* neuroscience, computation, materials science etc. Early work until the 1980s posed object detection as an alignment between a predefined (3d) shape and image evidence as a whole. As a result, considerable attention was devoted to representing shapes as a (hierarchical) combination of primitives as well as edge/line based techniques to establish correspondences. Improved representations (snakes) and models (scale space, pictorial structures (PS)) in combination with optimization methods (variational approaches, graphical models, *e.g.* Markov random field (MRF)) enhanced robustness against intra-class variations and noise.

To further accommodate the challenges of object detection, the theme of *invariance* gained popularity. Numerous approaches designed features invariant to geometry, *e.g.* under homographies or more specifically, under affinities, similarities, rotations; or invariant to appearance, *e.g.* scale invariant feature transform (SIFT), local binary patterns (LbP) [6], histogram of gradients (HOG) [7] etc. Starting with statistical methods, such as Eigenfaces [8] for object recognition, usage of more sophisticated machine learning techniques paved way for feature based learning, which was further strengthened by advances in convex optimization techniques (support vector machine (SVM), graphcuts). Increased robustness to background clutter was achieved by sliding window classifiers that operated in two major paradigms: *bag of words* (BOW) model which ignored spatial ordering between object parts and Pictorial Structures (PS) model which did the opposite.

In the past few years, feature based learning has gained immensely due to large annotated datasets, efficient hardware and high capacity models, *e.g.* deep convolutional neural networks

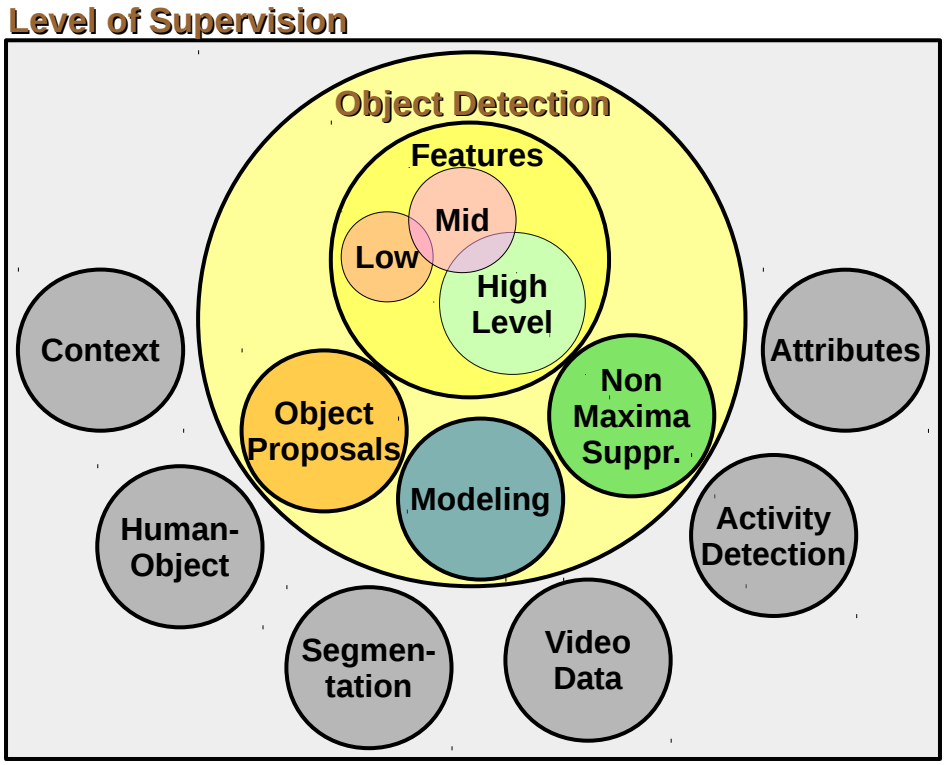


Fig. 1.3: An overview of research themes for addressing the object detection problem.

(DCNN) which has enabled end-to-end learning. Simultaneously, efficient search mechanisms have evolved from exhaustive sliding window to data driven techniques, *e.g.* object proposals where search is restricted to image subregions. In this regard, Fig 1.3 illustrates various research themes within object detection which are briefly discussed below. Furthermore, a more detailed prior work of each subproblem is discussed in the corresponding chapters.

1.3.1. Object Modeling

Template matching using sliding window search [8, 9] is a primitive method of object modeling. Improved accuracy is obtained by boosting where subsequent weak classifiers are tuned by previously misclassified examples [9], specifically using random forests in [10]. Further, [11] performs a non-linear mapping of original samples based on basis samples before learning a weak classifier, where basis samples are composed of hard negative examples. As discussed in Section 1.2, high variability in object appearance is a challenging problem. One possible solution is to use an ensemble of exemplar-SVMs [12] which are shown to generalize well. Another choice is to learn mixture models where each component represents a subset of examples. Clustering examples is either predefined *e.g.* based on viewpoints [13] or performed automatically [14]. Alternatively, a boosted classifier based on soft binning is proposed in [15]. Here, clustering is based on various features pooled from regionlets, which are base feature extraction regions with fixed relative positions within the object.

Conversely, the *BOW* model [16] neglects spatial ordering and defines objects as a col-

lection of local features. Although this model is intuitive and generative, the lack of rigorous treatment of spatial relations between object components holds it back from localizing problems such as segmentation. This problem is partially dealt with by extending the model using a multi-resolution approach [17].

A more robust extension is the *Implicit Shape Model* (ISM) or *Star Model* [18] which combines object recognition and segmentation within a generalized Hough transform. To this end, a codebook that maps an image patch appearance to the probabilistic vote for the possible location of the object centroid is learned. A discriminative interpretation is proposed in [19] which learns class specific weights for votes from training data. This is further generalized by efficiently replacing generative with discriminative, class-specific codebooks using a random forest framework in [20, 21]. Boosting is incorporated into the model during classification [22] or regression [23] or both [24] and a multi-class extension is proposed in [25].

In order to reason about object parts instead of only its centroid, a *Deformable Part Model* (DPM) or *Tree Model* is proposed in [26, 27]. Testing in a sliding window procedure, the model maximizes the linear combination of part visibility scores and inter-part deformations, which can be efficiently solved if the latter graph is a tree. During training, linear weights are learned in a convex procedure by pre-determining the otherwise hidden part locations greedily. More principled approaches to discovering parts are proposed in [28, 29, 30, 31] and in [32] where parts between various object models are shared. Variations due to deformation are parameterized at the object level [33] or part level [34] and part-occlusion is addressed in [35, 36, 37, 38]. While inter-part deformation is realized as a predefined tree in [27], it can also be learned from training data as in [39]. We compare the tree and star model for object detection in Chapter 4.

New paradigms replacing computationally expensive sliding window protocol is presently an active field of research. While an efficient search strategy is proposed in [40], a majority of approaches formulate object detection as classifying object proposals [41, 42]. Further, standalone techniques that directly regress object locations are proposed in [43, 44]. Combining large annotated datasets with high capacity models, the latter approaches currently are state-of-the-art. A study indicating the strengths and limitations of such models is presented in [45] and [46] discusses possible future directions for research on this topic.

1.3.2. Designing Features

Recent trends in feature learning have surpassed most handcrafted features. However, a historical perspective is presented here for completeness. *Low level features* based on pixel intensities within a local neighborhood are computed as corners [47], interest points (SIFT) [48, 49, 50], gradient histograms (HOG) [7], textures (LbP) [6] and their combinations [51]. HOG, being highly popular, is generalized to 3d in [52] and its computation is sped up using lookup tables in [53, 54]. An extension from gradients to textures is presented in the histogram of sparse codes [55]. With increasing ubiquity of depth cameras, low level features based on new modalities are proposed in [56, 57].

Mid level features are defined over a larger spatial support and are often defined by grouping local low level information. Popular features in this category are contours which are obtained by grouping boundary [58] or edge [59] information. Contour based object detection is proposed in [60, 61, 62, 63, 64]. Adaptive pooling of appearance based features is proposed

in [65, 66] and an approach with similar motivation is investigated in Chapter 3.

High level features have a wide spatial support and often span the entire image. Fisher vectors [67] which are composed of average-pooled zero and first order moments of local features have been applied to object detection [68]. GIST [69] is also composed of aggregated mid level features extracted from predefined partitions of the image. Classifier scores pooled over the entire image are also used as features as in [70].

1.3.3. Object Proposals

As discussed in Section 1.3.1, there is increasing interest towards designing class agnostic object proposal techniques. *Bottom-up* methods rely on grouping low level (*e.g.* pixel level) information to achieve the end goal. In [71], a sequential keypoint proposal scheme is presented where a keypoint is localized in the context of previously proposed keypoints and their appearance. As for region proposals, [72] is based on saliency computation and [73, 74] are based on related ad-hoc measures. In [75, 76], region proposals are obtained by grouping superpixels, which are in turn obtained by image segmentation. Casting entirely as a regression problem, [77] learns a function using deep neural networks to output object bounding boxes given an image as input. On the other hand, [78, 79] design cascaded classifiers to perform an exhaustive search for image subregions containing objects.

Top down methods rely on high level (*e.g.* image level) information to achieve the end goal. In [80], object proposals are adopted from K-nearest annotated training images which are retrieved based on similarity between global image descriptors. An approach based on guided partitioning of image subregions to search for those containing objects is designed in [81]. The issue of finding small objects in images due to low resolutions is dealt with using a multi-resolution framework in [82]. Finally, a study towards understanding the characteristics and limitations of various object proposal techniques is presented in [83].

1.3.4. Non Maximal Suppression

It is desirable to have object detection systems performing at high recall and precision. Following standard evaluation [2], duplicate detections of an object are considered as false positives thereby deteriorating performance. This is often the case with sliding window approaches which can have multiple highly overlapping windows classified as the same instance of an object. To this end, a class specific approach to combine multiple such detections into a single bounding box is proposed in [84, 85, 86]. This is generalized in [87, 88] where co-occurrence statistics between multiple classes is utilized to reconcile multiple detections within and between classes. A different strategy is adopted in [89, 90] where coarse bounding box estimates are iteratively refined into those that tightly fit objects.

1.3.5. Context

There is a broad agreement about the ability of contextual cues to offset challenges posed by object modeling [91]. For instance, given the context of larger, more easily detectable objects like *keyboard* and *monitor*, detecting a *mouse* could be more accurate. Typically, context has been employed to improve classifier performance. In [92], each window is scored in the context of all other windows of the image considering their appearance as well as spatial

relations. Similar approaches are proposed in [88, 93, 94, 95, 96]. This strategy is generalized in [51, 97, 98] where all windows are jointly labeled in the context of each other. This is realized as a conditional random field (CRF) where unary potentials are independent window classifier scores and pairwise potentials impose the contextual constraints.

Further, recurring image regions unrelated to objects and occurring outside them are discovered as parts in [99]. High level information has also been used as contextual cues to improve performance. E.g., in [100, 101] high level facial pose and bodypart priors are used to improve localizing facial and bodyparts respectively. We investigate the utility of high level cues in form of human pose for object detection in Chapter 4.

Context can also been used to efficiently search image regions for objects. An active search strategy that sequentially chooses the next window for classification based on previously evaluated windows is proposed in [102, 103, 104, 105].

1.3.6. Video data

Videos provide not only temporal continuity constraints but also rich information regarding artifacts caused by objects undergoing motion, *e.g.* deformations, occlusions and motion blur. In this regard, video data is routinely used to adapt detectors pretrained on image data to the video domain. Inspite of ignoring temporal modeling, performance gains are obtained from incorporating motion incurred appearance variation in [106, 107, 108, 109]. Enforcing temporal consistency within a video to improve object detector performance is demonstrated in [110] and is generalized to joint object discovery and segmentation within a single video in [111]. Learning object models from a collection of videos is also popular and an unsupervised approach is presented in [112]. Weakly supervised approaches typically operate on a labeled collection of videos and localize objects based on appearance or motion similarities across the collection [113, 114, 115, 116]. We explore this aspect in the context of small and medium sized household objects in Chapter 5.

1.3.7. Unified Problems

Approaching traditionally disparate problems in a unified framework can be advantageous in that redundant processing can be coupled and each subtask can gain complementary information from the rest. This also resonates with the long term goal of realizing a computationally efficient multi-tasking agent.

To this end, detection and segmentation are closely associated problems and are often coupled together. Instance segmentation has been explored as preprocessing for object detection in [68, 117, 118] and for 3d object detection in [119]. An opposite strategy is explored in [56]. Several approaches explore bidirectional interaction between object detection and instance segmentation [120, 121] and semantic segmentation in images [122, 123, 124, 125] and video [111].

Object detection has also been combined with object pose estimation [126, 127] and other higher level tasks such as action detection [128, 129] and human pose estimation [35]. We demonstrate unified object detection and human pose estimation under human-object interactions in Chapter 4.

1.3.8. Attributes and Affordances

Attributes are descriptive physical quantities *e.g.* weight, size etc. As these are more semantic, gains are typically obtained from complementary modalities such as text or web-data. On the other hand, most object classes are designed for predefined functionality. Affordances *i.e.* physically grounded regions in objects that serve a specific functionality, can be used as an alternate representation to offset the complex variety in appearance. Pixelwise affordance reasoning of objects is explored in [130, 131] and in Chapter 6.

1.3.9. Reduced supervision

Increasing diversity in training data is primordial for improvement in object detection performance [32]. However, collecting ground truth annotations over ever increasing data can be expensive and intractable. This provides an apt motivation towards exploring techniques that can perform well despite of reduced supervision levels. Several approaches optimize the extent of human intervention for annotation during model learning [132, 133, 134].

Other approaches investigate incorporation of unlabeled examples. *Semi supervised* approaches initialize models using a few examples which are then improved by augmented unlabeled image data [135] or unlabeled videos [136, 137]. Strong human intervention is also reduced by using motion cues in videos to improve object detectors in [107, 138]. Data augmentation through the use of generative approaches *e.g.* projecting 3d models of objects on the image plane is explored in [139, 140]. In a different strategy, hard negative examples are mined [26, 141] to augment the set of negative examples.

Weakly supervised approaches utilize ground truth labels acquired at a higher level of abstraction. A popular scheme is to utilize ground truth labels at an image level thereby lacking object localization information. Approaches in [93, 142, 143, 144, 145] explore such a scenario. A further generalization is explored in [146, 147, 148] where ground truth labels are available only on collections of images called bags. A positive bag is where the object of interest occurs in at least one image, and negative bags are where the object occurs in none of them. An exactly opposite strategy is employed in [149, 150] where localization is inferred by the responses of a pretrained classifier on various subregions of an image. In the context of weakly supervised learning, we address learning object detectors from weakly labeled videos in Chapter 5 and weakly supervised segmentation in Chapter 6.

Unsupervised approaches renounce the support of data annotation and learn object models by exploring data for recurring patterns. A survey is presented in [151]. While the approach in [152] proposes to represent novel object classes as a combination of pretrained detectors, a co-detection and segmentation approach for dominant objects is adopted in [28, 125, 153]. In a different approach, assuming human-object interactions are indicative of object class, high level joint trajectory information is used to localize objects in [154].

1.4. THESIS LAYOUT

The dissertation segregates the broad problem of object detection into smaller sub-problems and dedicates a chapter to studying each of them in detail. The underlying theme of the thesis is for each sub-problem to detect or describe small and medium sized objects within the purview

of human-object interactions. Fig 1.4 illustrates the problems addressed by each chapter. The thesis is organized as follows:

- Chapter 2 briefly discusses fundamental modules that are used in this dissertation. This includes Hough forest for object detection [20, 21], TRW-S algorithm [155] for inference over loopy MRFs, superpixel representation [156] of RGB/RGBD images and optical flow [157] between video frames in order to extract dense correspondences.

- Chapter 3 addresses the independence assumption of image patches in the Hough based object detection framework [20]. To this end, mid level representations derived from grouping low level features in a local neighborhood are shown to improve object detection performance in RGB and RGBD image datasets. Considering inherent sparsity of the mid level features, oblique forests which are shown to perform better than axis-aligned forests are proposed. The sparsity is controlled by adjusting the support of the grouping neighborhood. Further, the benefit of combining hypotheses from low level and mid level features is demonstrated, indicating that both features encode complementary information. The details of this chapter was published in [158].

- Chapter 4 explores two paradigms for appearance based object modeling, namely star and tree model. It is shown that while the tree model achieves slightly better perfor-

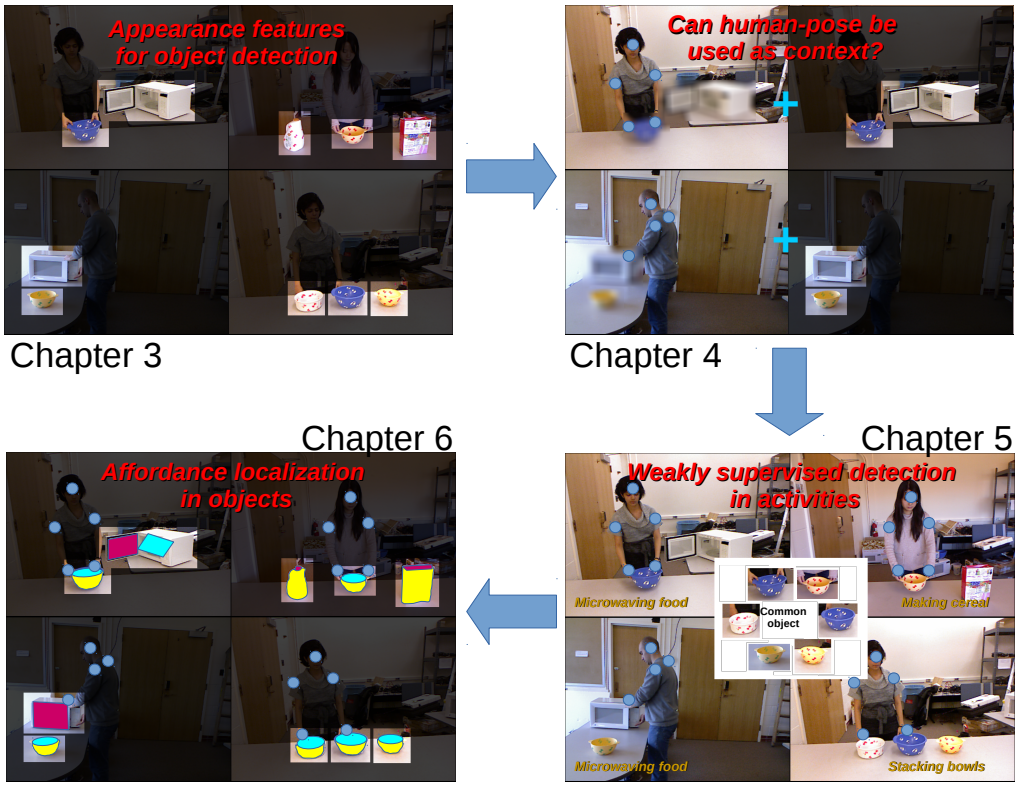


Fig. 1.4: Sequence of subproblems addressed in this dissertation to achieve the goal discussed in Fig 1.1.

mance, appearance features are of limited reliability for small and medium sized objects. To this end, the chapter explores an interaction centric perspective by introducing human context to modeling objects. This is formulated as an approach that combines two modalities, namely image appearance and human pose, for object detection. Evaluation is performed on three challenging video datasets that contain small objects that are often occluded during human-object interactions. It is shown that while human pose alone is insufficient to accurately localize objects, combining it with an independently learned appearance based detector results in improved performance, irrespective of the underlying pose estimation technique. The details of this chapter was published in [159].

- Chapter 5 addresses weakly supervised detection of small and medium sized objects from a collection of activity videos. This is formulated as a two stage framework where the first stage generates a set of spatio-temporal object proposals and a subsequent stage selects a subset that best describes the object common in all videos. The framework greedily infers multiple object instances from each video to maximize the utility of data. We show that approaches that rely entirely on object motion or appearance fail for this task. Further, we show that low level object appearance and high level human pose based functionality are complementary and coupling them greatly improves performance. This is demonstrated on three challenging datasets where performance comparable to fully supervised methods are obtained despite reduced supervision. The details of this chapter was published in [160].

- Chapter 6 addresses the problem of localizing functional regions within objects. Assuming objects are pre-localized, this is formulated as a weakly supervised semantic affordance segmentation problem. To this end, an expectation-maximization approach that can be trained on image level and/or clickpoint annotations is proposed. Further, we explore the possibility of using contextual information from human-object interactions to transfer clickpoint annotations to images with only image level labels. The approach is evaluated on two datasets, including a custom dataset containing 3090 images and 9916 object instances with rich contextual information with pixel wise affordance annotations. The details of this chapter can be found in [161].

- Finally, Chapter 7 presents conclusion and future work.

Related work of each sub-problem will be discussed in further detail in the corresponding chapter that addresses it.

List of Publications

- A. Srikantha and J. Gall.
  Hough-based Object Detection with Grouped Features.
  In International Conference on Image Processing (ICIP), 2014.
  https://ps.is.tuebingen.mpg.de/publications/srikantha-icip-2014

- A. Srikantha and J. Gall.
  Discovering Object Classes from Activities.
  In European Conference on Computer Vision (ECCV), 2014.
  https://ps.is.tuebingen.mpg.de/publications/srik-eccv-2014

- A. Srikantha and J. Gall.
  Human Pose as Context for Object Detection.
  In British Machine Vision Conference (BMVC), 2015.
  https://ps.is.tuebingen.mpg.de/publications/srik-bmvc-2015

- A. Srikantha and J. Gall.
  Weak Supervision for Detecting Object Classes from Activities.
  In Computer Vision and Image Understanding (CVIU), 2016.
  https://pages.iai.uni-bonn.de/gall_juergen/

- J. Sawatzky*, A. Srikantha* and J. Gall.
  Weakly Supervised Affordance Detection.
  In Computer Vision and Pattern Recognition (CVPR), 2017.
  https://pages.iai.uni-bonn.de/gall_juergen/

CHAPTER 2

Preliminaries

Contents

2.1. HOUGH FORESTS FOR OBJECT DETECTION

Hough forests are random forests adapted to efficiently implement a generalized Hough transform. We now briefly discuss the approach proposed in [20, 21].

For training, it is assumed that for each class $c \in \mathcal{C}$, a set of training images is available. For the positive classes, D-dimensional bounding box annotations to determine the center and the size of the positive examples are also provided. Here, $D = 2$. Each tree T in the Hough forest $\mathcal{T} = \{T_t\}$ is then constructed from a set of patches $\{\mathcal{P}_i = (\mathcal{I}_i, c_i, \mathbf{d}_i)\}$ that are sampled from the examples where $\mathcal{I}_i$ are the extracted features associated with a patch of fixed size in $\mathbb{R}^D$, c_i is the class label of the exemplar the patch is sampled from and $\mathbf{d}_i$ is the displacement vector from the patch center to the centroid of the training exemplar. Patches from negative instances have a class label $c_i = 0$ and a pseudo displacement $\mathbf{d}_i = \mathbf{0}$. The positive examples are scaled to unit size, so that the longest spatial dimension is about $s_u = 100$. Without loss of generality, it is assumed that the per class aspect ratio is fixed and that the size of the object can be represented by a scale factor s/s_u.

A tree is composed of leaf and non-leaf nodes. To construct a leaf node L, information from incoming patches is used to store the class probability $p(c|L)$ and a list $D_c^L = \{\mathbf{d}_i\}_{c_i=c}$ of valid offset vectors. Non-leaf nodes are assigned binary tests whose domain is the feature descriptor $\mathcal{I}_i = (I_i^1, \ldots, I_i^F)$, where each I_i^j is a fixed sized feature matrix *e.g.* intensity, gradient and/or associated human pose and F is the number of channels. A binary test on a patch $t_\theta(\mathcal{I}) \rightarrow \{0, 1\}$ parameterized by $\theta = \{f, \mathbf{p}, \mathbf{q}, \tau\}$ is defined using $f \in \{1, 2, \ldots, F\}$, two positions $\mathbf{p} \in \mathbb{R}^D$ and $\mathbf{q} \in \mathbb{R}^D$ within the feature matrix and a real valued threshold τ. The test is defined as:

$$t_\theta(\mathcal{I}_i) = \begin{cases} 0 & \text{if } I_i^f(\mathbf{p}) < I_i^f(\mathbf{q}) + \tau, \\ 1 & \text{otherwise.} \end{cases} \tag{2.1}$$

A Hough tree is constructed recursively starting from the root. During construction, a node receives a set of training patches $P = \{\mathcal{P}_i\}$. If the depth of the node is maximum $d_{max} = 25$

or the number of patches is below a threshold $N_{min} = 20$, the constructed node is declared as a leaf and corresponding $\left(p(c|L), D_c^L\right)_{c\in\mathcal{C}}$ information is computed and stored. Otherwise, a non-leaf node is created and an optimal binary test $\hat{\theta}$ is chosen from a large pool of randomly generated binary tests such that

$$\begin{aligned} \hat{\theta} &= \underset{\theta^k}{\text{argmax}}\ \Delta U_*(P, \theta^k) \\ \Delta U_*(P, \theta) &= U_*(P) - \sum_{b\in\{0,1\}} \frac{|P_b(\theta)|}{|P|} U_*\left(P_b(\theta)\right), \end{aligned} \tag{2.2}$$

where children nodes with incoming patches $P_b(\theta) = \{\mathcal{P}_j | \mathcal{P}_j \in P, t_\theta(\mathcal{I}_j) = b\}$ are created by binary test t_θ. The evaluation criterion U_* for a binary test is designed to minimize uncertainties in both discrete and continuous random variables *i.e.* class labels and offset vectors, respectively. To this end, the class-label uncertainty measuring the impurity of class labels c_i is defined as:

$$U_1(P) = -|P| \cdot \sum_{c\in\mathcal{C}} p(c|P) \ln\left(p(c|P)\right). \tag{2.3}$$

The second measure is called offset uncertainty and corresponds to the impurity of the offset vectors $\mathbf{d}_i$, defined as:

$$U_2(P) = \sum_{c\in\mathcal{C}\setminus 0} \left(\sum_{\mathbf{d}\in D_c^P} \left\| \mathbf{d} - \frac{1}{|D_c^P|} \sum_{\mathbf{d}'\in D_c^P} \mathbf{d}' \right\|^2 \right), \tag{2.4}$$

where D_c^P is the set of all offsets of patches from class c in set P. This corresponds to the variance of the vote distribution which is assumed to be unimodal in the present case. Further, it is to be noted that background patches corresponding to $c_i = 0$ are ignored here.

As for the pool of binary test parameters, $\{\theta^k\}$ are first generated by sampling parameters $f, \mathbf{p}$ and $\mathbf{q}$ uniformly. Parameter τ is sampled from the interval observed from incoming patches. This is followed by assigning U_* by randomly choosing between Equations (2.3) and (2.4) unless there are too few ($< 5\%$) negative patches in which case Eqn (2.3) is selected.

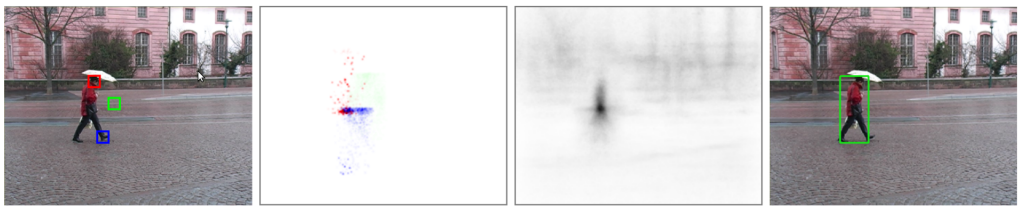


Fig. 2.1: For each of the three patches emphasized in (a), the pedestrian class-specific Hough forest casts weighted votes about the possible location of a pedestrian (b) (each color channel corresponds to the vote of a sample patch). Note the weakness of the vote from the background patch (green). After the votes from all patches are aggregated into a Hough space (c), the pedestrian can be detected (d) as a peak in this image. Figure adapted from [21].

It is to be noted that $* = 1$ or 2, depending on the random choice. Interleaving the nodes that penalize both class label and offset uncertainties ensures low variations on both fronts within patches incoming in a leaf.

During detection, as shown in Fig 2.1, image patches are passed through each tree in the Hough forest and the resulting leaves are used to cast votes in the Hough space $\mathcal{H} \subseteq \mathbb{R}^H$. The Hough space encodes the hypothesis $\mathbf{h}$ for an object position in scale-space and class, *i.e.* here, $H = 4$. Let a patch $\mathcal{P}_\mathbf{y} = \left(\mathcal{I}_\mathbf{y}, c_\mathbf{y}, \mathbf{d}_{c(\mathbf{y})}\right)$ be centered at position $\mathbf{y} \in \Omega \subseteq \mathbb{R}^D$ in the test image. Here Ω is set of all pixel locations, $\mathcal{I}_\mathbf{y}$ is the set of observed features of the patch, $c_\mathbf{y}$ is the hidden class label and $\mathbf{d}_{c(\mathbf{y})}$ is the hidden displacement from the patch to the unknown object's center. Based on the appearance $\mathcal{I}_\mathbf{y}$, patch $\mathcal{P}_\mathbf{y}$ ends in a leaf $L(\mathbf{y})$. Let $\mathbf{h}(c, \mathbf{x}, s)$ be the hypothesis for the object belonging to class c with size s and centered at $\mathbf{x} \in \Omega$. The conditional probability $p(\mathbf{h}|L)$[1] can be computed as

$$\begin{aligned} p\left(\mathbf{h}(c,\mathbf{x},s)|L(\mathbf{y})\right) &= \sum_{l\in\mathcal{C}} p\left(\mathbf{h}(c,\mathbf{x},s)\middle|c(\mathbf{y})=l, L(\mathbf{y})\right)\cdot p\left(c(\mathbf{y})=l\middle|L(\mathbf{y})\right), \\ &= p\left(\mathbf{h}(c,\mathbf{x},s)\middle|c(\mathbf{y})=c, L(\mathbf{y})\right)\cdot p\left(c(\mathbf{y})=c\middle|L(\mathbf{y})\right), \\ &= p\left(\mathbf{x}=\mathbf{y}-\frac{s}{s_u}\mathbf{d}(c)\middle|c(\mathbf{y})=c, L(\mathbf{y})\right)\cdot p\left(c(\mathbf{y})=c\middle|L(\mathbf{y})\right), \end{aligned} \tag{2.5}$$

where s_u is the unit size from the training data. Both factors in Eqn (2.5) are estimated during training. While $p(c|L)$ is estimated by the proportion of patches per class label reaching the leaf after training, the distribution $p(\mathbf{h}|c, L)$ can be approximated by a sum of Dirac measures $\delta_\mathbf{d}$ for the displacement vectors $\mathbf{d} \in D_c^L$:

$$p\left(\mathbf{h}(c,\mathbf{x},s)|L(\mathbf{y})\right) = \frac{p\left(c(\mathbf{y})=c|L(\mathbf{y})\right)}{|D_c^{L(\mathbf{y})}|}\left(\sum_{\mathbf{d}\in D_c^{L(\mathbf{y})}}\delta_\mathbf{d}\left(\frac{s_u(\mathbf{y}-\mathbf{x})}{s}\right)\right). \tag{2.6}$$

For the entire forest $\mathcal{T}$, we pass the features of the patch $\mathcal{I}_\mathbf{y}$ through all the trained trees and average the probabilities from Eqn (2.6) from different leaves:

$$p\left(\mathbf{h}|\mathcal{I}_\mathbf{y}\right) = \frac{1}{T}\sum_{t=1}^{|\mathcal{T}|} p\left(\mathbf{h}|L_t(\mathbf{y})\right), \tag{2.7}$$

where $L_t(\mathbf{y})$ is the corresponding leaf for tree T_t. The votes from all patches of the image are accumulated into the Hough space $\mathcal{H}$:

$$p\left(\mathbf{h}|\mathcal{I}\right) = \sum_{\mathbf{y}\in\Omega} p\left(\mathbf{h}|\mathcal{I}_\mathbf{y}\right). \tag{2.8}$$

The modes of $p\left(\mathbf{h}|\mathcal{I}\right)$ can be obtained by searching for local maxima using a Parzen estimator with a Gaussian kernel K:

[1] In the text, abbreviated forms $p(\mathbf{h}|L)$, $p(\mathbf{h}|c, L)$ and $p(c|L)$ are used for $p\left(\mathbf{h}(c,\mathbf{x},s)|L(\mathbf{y})\right)$, $p\left(\mathbf{h}(c,\mathbf{x},s)|c(\mathbf{y})=c, L(\mathbf{y})\right)$ and $p\left(c(\mathbf{y})=c|L(\mathbf{y})\right)$, respectively.

$$\begin{aligned} \hat{p}\left(\mathbf{h}|\mathcal{I}\right) &= \sum_{\mathbf{h}' \in \mathcal{N}(\mathbf{h})} w_{\mathbf{h}'} \cdot K(\mathbf{h} - \mathbf{h}'), \text{where} \\ w_{\mathbf{h}'} &= \sum_{\mathbf{y} \in \Omega} \sum_{t=1}^{|\mathcal{T}|} \sum_{\mathbf{d} \in D_c^{L_t(\mathbf{y})}} \frac{p\left(c(\mathbf{y}) = c | L_t(\mathbf{y})\right)}{|D_c^{L_t(\mathbf{y})}|} \delta_d \left(\frac{s_u(\mathbf{y} - \mathbf{x})}{s} \right). \end{aligned} \tag{2.9}$$

The weight of a hypothesis $w_{\mathbf{h}'}$ accumulates votes that support similar hypotheses $\mathbf{h}'(c, \mathbf{x}, s) \in \mathcal{H}$. After all votes are cast, $\hat{p}(\mathbf{h}|\mathcal{I})$ represents the sum of the weights of the hypotheses in the neighborhood of $\mathbf{h}$ weighted by a Gaussian kernel K. While the location of a local maximum $\hat{\mathbf{h}}(c, \mathbf{x}, s)$ encodes class, position and size of the object, the value of $\hat{p}(\hat{\mathbf{h}}|\mathcal{I})$ is not a probability but serves as a confidence measure for each hypothesis.

2.2. CONVERGENT TREE-REWEIGHTED MESSAGE PASSING FOR ENERGY MINIMIZATION

Algorithms for minimizing discrete energy are of fundamental importance in computer vision. We now briefly discuss a technique proposed in [155] which is guaranteed to yield a locally optimal solution. Many problems can be formulated in terms of minimizing an energy defined over an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ given by

$$E\left(\mathbf{x}|\theta\right) = \theta_{const} + \sum_{u \in \mathcal{V}} \theta_s\left(x_s\right) + \sum_{(u,v) \in \mathcal{E}} \theta_{uv}\left(x_u, x_v\right), \tag{2.10}$$

where $\mathcal{V}$ corresponds to a set of vertices and $\mathcal{E}$ corresponds to a set of edges. For each $u \in \mathcal{V}$, let x_u be a variable taking values in some discrete label space $\mathcal{C}_u$. Concatenating the variables at each node, we obtain a vector $\mathbf{x}$ with $n = |\mathcal{V}|$ elements. The vector takes values in the space $\mathcal{C} = \mathcal{C}_1 \times \mathcal{C}_2 \times \cdots \times \mathcal{C}_n$. Let symbols u and v denote nodes in $\mathcal{V}$, (u, v) an edge in $\mathcal{E}$, and j and k be values in $\mathcal{C}_u$ and $\mathcal{C}_v$, respectively.

Parameters θ of the energy are specified by the constant term θ_{const}, the unary term $\theta_u\left(j\right)$ and the pairwise term $\theta_{uv}\left(j, k\right)$. The last two terms are further denoted as $\theta_{u;j}$ and $\theta_{uv;jk}$, respectively. Then θ can be viewed as a vector $\theta = \{\theta_\alpha | \alpha \in \mathcal{J}\} \in \mathbb{R}^d$ where the index set $\mathcal{J}$ is

$$\mathcal{J} = \{const\} \cup \{(u; j)\} \cup \{(uv; jk)\}.$$

Note that $(uv; jk) \equiv (vu; kj)$, so $\theta_{uv;jk}$ and $\theta_{vu;kj}$ are the same element. Also, θ_u and θ_{uv} are used to denote vectors of size $|\mathcal{C}_s|$ and $|\mathcal{C}_u \times \mathcal{C}_v|$ respectively. The energy in Eqn (2.10) can then be written as the Euclidean product depending on two vectors θ and $\mathbf{x}$ as

$$E\left(\mathbf{x}|\theta\right) = \langle \theta, \phi\left(\mathbf{x}\right) \rangle = \sum_{\alpha \in \mathcal{J}} \theta_\alpha \phi_\alpha\left(\mathbf{x}\right), \tag{2.11}$$

where the different spaces of θ and $\mathbf{x}$ are reconciled by the mapping $\phi : \mathcal{C} \rightarrow \mathbb{R}^d$, which is in turn defined by $\phi_\alpha : \mathcal{C} \rightarrow \mathbb{R}$

$$\begin{aligned}\phi_{const}(\mathbf{x}) &= \mathbb{I}(\text{true})\\ \phi_{u;j}(\mathbf{x}) &= \mathbb{I}(x_u = j)\\ \phi_{uv;jk}(\mathbf{x}) &= \mathbb{I}(x_u = j)\cdot\mathbb{I}(x_v = k),\end{aligned}$$

where $\mathbb{I}(\cdot)$ is an indicator function returning one if its argument is true, and zero otherwise.

Belief Propagation (BP) algorithms approximate minimization of the energy $E(\mathbf{x}|\theta)$ as in Eqn (2.10). Typically, they maintain a message $M_{uv} = \{M_{uv:k}|k \in \mathcal{C}_t\}$ for each directed edge $(u \to v) \in \mathcal{E}$, which is a vector of $|\mathcal{C}_t|$ components. The vector of all messages is denoted as $M = \{M_{uv}\}$. The basic operation of BP is to pass a message from node u to node v for the directed edge $(u \to v) \in \mathcal{E}$. It consists of updating the vector M_{uv} as follows:

$$M_{uv;k} := \min_{j\in\mathcal{C}_u}\left\{(\bar{\theta}_{u;j} + \sum_{(w\to u)\in\mathcal{E}, w\neq v} M_{wu;j}) + \bar{\theta}_{uv;jk}\right\} + const_v, \tag{2.12}$$

where $const_v$ is a constant independent of k. The message for a directed edge $(u \to v)$ is said to be valid if this update does not change M_{uv}. The BP algorithm keeps passing messages from edges in some order until convergence, *i.e.* until all messages are valid. While it provides exact solutions if the graph is a tree, general convergence is not guaranteed for graphs containing loops. Significant speedups in the latter case have been obtained for special classes of pairwise potentials using distance transforms [162]. Alternatively, instead of storing the original parameter vector $\bar{\theta}$ and messages M, a single parameter vector after reparameterization $\theta = \bar{\theta}(M)$ can be stored, where θ is called a reparameterization of $\bar{\theta}$ *i.e.* $\theta \equiv \bar{\theta}$ if they define the same energy function (*i.e.* $E(\mathbf{x}|\theta) = E\left(\mathbf{x}|\bar{\theta}\right)\ \ \forall\ \mathbf{x} \in \mathcal{C}$).

Further, let $\mathcal{T}$ be a collection of trees in graph $\mathcal{G}$ and $\rho^T, T \in \mathcal{T}$ be some distribution on $\mathcal{T}$. It is assumed that each tree has a non zero probability and each edge in $\mathcal{E}$ is covered by at least one tree. For a given tree $T = \left(\mathcal{V}^T, \mathcal{E}^T\right)$ a set corresponding to the indices associated with vertices and edges in the tree is defined as

$$\mathcal{J}^T = \{const\} \cup \{(u;j)\,|u \in \mathcal{V}^T\} \cup \{(uv;jk)\,|\,(u,v) \in \mathcal{E}^T\}.$$

To each tree $T \in \mathcal{T}$, an energy parameter θ^T is associated that respects the structure of T. More precisely, the parameter θ^T must belong to the following linear constraint set:

$$\mathcal{A}^T = \{\theta^T \in \mathbb{R}^d \mid \theta^T_\alpha = 0\ \ \forall\ \alpha \in \mathcal{J}\backslash\mathcal{J}^T\} \tag{2.13}$$

By concatenating all of the tree vectors, a larger vector $\boldsymbol{\theta} = \{\theta^T|T \in \mathcal{T}\}$ is created, which is an element of $\mathbb{R}^{d\times|\mathcal{T}|}$. The constraint set of vector $\boldsymbol{\theta}$ is given by

$$\mathcal{A} = \{\boldsymbol{\theta} \in \mathbb{R}^{d\times|\mathcal{T}|} \mid \theta^T \in \mathcal{A}^T\ \ \forall\, T \in \mathcal{T}\} \tag{2.14}$$

Considering the function $\boldsymbol{\Phi}_\rho : \mathcal{A} \to \mathbb{R}$ defined as follows:

$$\boldsymbol{\Phi}_\rho = \sum_T \rho^T \min_{\mathbf{x}\in\mathcal{C}} \langle\theta^T, \phi(\mathbf{x})\rangle, \tag{2.15}$$

Algorithm 1 TRW-S algorithm for a graph with monotonic chains

1: Set all messages to zero
2: Set E_{bound} to constant
For nodes $u \in \mathcal{V}$ do the following operations in the order of increasing $i(u)$:
– Compute $\hat{\theta}_u = \bar{\theta}_u + \sum_{(w,u)\in\mathcal{E}} M_{wu}$. Normalize vector $\hat{\theta}_u$ as follows:
$\delta := \min_j \hat{\theta}_{u;j} \qquad \hat{\theta}_{u;j} := \hat{\theta}_{u;j} - \delta \qquad E_{bound} := E_{bound} + \delta$
– For all edges $(u,v) \in \mathcal{E}$ with $i(u) < i(v)$ update and normalize message M_{uv} as follows:
$M_{uv;k} := \min_j \{(\gamma_{uv}\hat{\theta}_{u;j} - M_{vu;j}) + \bar{\theta}_{uv;jk}\}$
$\delta := \min_j M_{uv;k} \qquad M_{uv;k} := M_{uv;k} - \delta \qquad E_{bound} := E_{bound} + \delta$
3: Reverse the ordering: set $i(u) := |\mathcal{V}| + 1 - i(u)$
4: Check whether a stopping criterion is satisfied; if yes, terminate, otherwise go to step 2.

it can be shown that if $\sum_T \rho^T \theta^T = \bar{\theta}$ then $\mathbf{\Phi}_\rho(\boldsymbol{\theta})$ is a lower bound on the optimal value of the energy for vector $\bar{\theta}$ as per Eqn (2.10), following from Jensen's inequality. To get the tightest bound, the following maximization problem is considered

$$\max_{\boldsymbol{\theta}\in\mathcal{A}, \sum_T \rho^T\theta^T \equiv \bar{\theta}} \mathbf{\Phi}_\rho(\boldsymbol{\theta}). \tag{2.16}$$

While several versions of maximizing Eqn (2.16) are possible, a practical algorithm is described in Alg 1. The input to the algorithm is an energy function specified by parameter vector $\bar{\theta}$. The method works by passing messages; for each directed edge $(u \to v) \in \mathcal{E}$ there is a message M_{uv} which is a vector of $\mathcal{C}_t$ components. The algorithm is initialized by (1) selecting an ordering of nodes $i(\cdot)$, *i.e.* a mapping of nodes in $\mathcal{V}$ onto the set $\{1, 2, \ldots, |\mathcal{V}|\}$; (2) selecting chains $T \in \mathcal{T}$ which are monotonic with respect to $i(\cdot)$, [2] where each edge is covered by at least one chain; and (3) choosing a probability distribution ρ over chains $T \in \mathcal{T}$ such that $\rho^T > 0, \sum_T \rho^T = 1$. These choices define coefficients γ_{uv} in Alg 3 by $\gamma_{uv} = \rho_{uv}/\rho_u$ where ρ_{uv} and ρ_u are edge and node appearance probabilities, respectively.

The stopping criterion is defined using a heuristic where the procedure is terminated if the value of the lower energy bound E_{bound} has not increased during the last N iterations. Finally, constructing a solution $\mathbf{x}$ given reparameterization $\hat{\theta} = \sum_T \rho^T \theta^T$ involves greedily choosing label x_u that minimizes $\hat{\theta}(x_u) + \sum_{i(w)<i(u)} \hat{\theta}_{wu}(x_w, x_u)$ in the order defined by $i(\cdot)$.

2.3. IMAGE SUPERPIXEL REPRESENTATION

The problem of segmenting an image into regions plays a powerful role in computational vision problems. We briefly discuss the approach from [156] that efficiently captures perceptually important groupings or regions in an image using a graph based approach. Let the image be represented as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with vertices $u \in \mathcal{V}$ corresponding to pixels and edges $(u,v) \in \mathcal{E}$ corresponding to pairs of neighboring pixels. Each edge $(u,v) \in \mathcal{E}$

[2] Graph $\mathcal{G}$ and chains $T \in \mathcal{T}$ are said to be monotonic iff the ordering of nodes $i(u), u \in \mathcal{V}$ exists such that each chain T satisfies the following property: if $u_1^T, \ldots, u_{n(T)}^T$ are the consecutive nodes in the chain, then the sequence $i(u_1^T), \ldots, i(u_{n(T)}^T)$ is monotonic.

Algorithm 2 Graph based segmentation algorithm

1: Input is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with n vertices and m edges. The output is a segmentation of $\mathcal{V}$ into components $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_r)$.
2: Sort $\mathcal{E}$ into $E = (o_1, \ldots, o_m)$, by non-decreasing edge weight.
3: Start with a segmentation $\mathcal{S}^0$, where each vertex u is its own component.
4: **for** $q = 1, \ldots, m$ **do**
5: To construct $\mathcal{S}^q$ given $\mathcal{S}^{q-1}$, let u and v denote the vertices connected by the q^{th} edge in the ordering *i.e.* $o_q = (u, v)$. Let components of $\mathcal{S}^{q-1}$ be such that $\mathcal{S}_i^{q-1}$ contains u and $\mathcal{S}_j^{q-1}$ contains v. If $\mathcal{S}_i^{q-1} \neq \mathcal{S}_j^{q-1}$ and $\theta_{st} \leqslant MInt(\mathcal{S}_i^{q-1}, \mathcal{S}_j^{q-1})$ then $\mathcal{S}^q$ is obtained by merging $\mathcal{S}_i^{q-1}$ and $\mathcal{S}_j^{q-1}$. Otherwise $\mathcal{S}^q = \mathcal{S}^{q-1}$.
6: Return $\mathcal{S} = \mathcal{S}^q$.

has a corresponding weight θ_{uv}, which is a non-negative measure of dissimilarity between neighboring elements u and v based on low level attributes such as color, depth, optical flow etc.

In the graph based approach, a segmentation $\mathcal{S}$ is a partition of $\mathcal{V}$ into components such that each component $\mathcal{S}_i \in \mathcal{S}$ corresponds to a connected component in a graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ where $\mathcal{V}_i \subseteq \mathcal{V}$ and $\mathcal{E}_i \subseteq \mathcal{E}$. The partitioning is subject to the constraint that vertices in a component are similar, whereas those in different components are dissimilar.

To this end, a predicate $\mathcal{D}$ is defined to evaluate whether or not there is evidence for a boundary between two components in a segmentation. The internal difference of a component $\mathcal{V}_i \subseteq \mathcal{V}$ is defined to be the largest weight in the minimum spanning tree of the component $MST(\mathcal{V}_i, \mathcal{E}_i)$. That is,

$$Int(\mathcal{V}_i) = \max_{(u,v) \in MST(\mathcal{V}_i, \mathcal{E}_i)} \theta_{uv}. \tag{2.17}$$

The difference between two components $\mathcal{V}_i, \mathcal{V}_j \subseteq \mathcal{V}$ is defined as the minimum weight of the edge connecting the two components. That is,

$$Dif(\mathcal{S}_i, \mathcal{S}_j) = \max_{u \in \mathcal{V}_i, v \in \mathcal{V}_j, (u,v) \in \mathcal{E}} \theta_{uv}. \tag{2.18}$$

If there is no edge connecting $\mathcal{S}_i$ and $\mathcal{S}_j$ then $Dif(\mathcal{S}_i, \mathcal{S}_j) = \infty$. The evidence for a boundary between a pair of components is given by checking the difference between them as:

$$\mathcal{D}(\mathcal{S}_i, \mathcal{S}_j) = \begin{cases} \text{true} & \text{if } Dif(\mathcal{S}_i, \mathcal{S}_j) > MInt(\mathcal{S}_i, \mathcal{S}_j), \\ \text{false} & \text{otherwise} \end{cases} \tag{2.19}$$

where the minimum internal difference, $MInt$, is defined by

$$MInt(\mathcal{S}_i, \mathcal{S}_j) = \min \left(Int(\mathcal{S}_i) + \frac{k}{|\mathcal{S}_i|} \ , \ Int(\mathcal{S}_j) + \frac{k}{|\mathcal{S}_j|} \right), \tag{2.20}$$

where k is a parameter that sets the scale of observation in that larger k causes a preference for larger components. The segmentation algorithm is shown in Alg 2. It can be shown that the final segmentation produced is independent of which non-decreasing order of edges is used and is neither too coarse nor too fine in spite of the inherently greedy procedure.

2.4. LARGE DISPLACEMENT OPTICAL FLOW

Motion in the form of optical flow is one of the fundamental bottom-up cues for segmentation and tracking. We briefly discuss the large displacement optical flow proposed in [157]. Let $\mathbf{I}_1$ and $\mathbf{I}_2 : (\Omega \subseteq \mathbb{R}^2) \rightarrow \mathbb{R}^d$ be the first and the second frame to be aligned. A gray scale image is represented with $d = 1$, whereas $d = 3$ for color images. Further, $\mathbf{x} := (x, y)^\top$ denotes a point in the image domain Ω, and $\mathbf{w} := (u, v)^\top$ is the optical flow field, *i.e.*, a function $\mathbf{w} : \Omega \rightarrow \mathbb{R}^2$. A common assumption is that of color constancy, *i.e.* corresponding points should have the same gray or color value. This can be expressed by the energy

$$E_{color}(\mathbf{w}) = \int_\Omega \Psi \left(|\mathbf{I}_2(\mathbf{x} + \mathbf{w}(\mathbf{x})) - \mathbf{I}_1(\mathbf{x})|^2 \right) d\mathbf{x}, \tag{2.21}$$

which penalizes deviations from the assumption of color constancy. A robust function $\Psi(s^2) = \sqrt{s^2 + \varepsilon^2}, \varepsilon = 0.001$ allows to deal with long displacements, occlusions and other non-Gaussian deviations of the matching criterion. Allowing for illumination variations, the constraint in Eqn (2.21) is supplemented by a constraint on the gradient, which is invariant to additive brightness changes:

$$E_{grad}(\mathbf{w}) = \int_\Omega \Psi \left(|\nabla \mathbf{I}_2(\mathbf{x} + \mathbf{w}(\mathbf{x})) - \nabla \mathbf{I}_1(\mathbf{x})|^2 \right) d\mathbf{x}. \tag{2.22}$$

Complementary to energies based on local descriptors, regularization can be enforced by penalizing the total variation of the flow field as:

$$E_{smooth}(\mathbf{w}) = \int_\Omega \left(|\nabla u(\mathbf{x})|^2 + |\nabla v(\mathbf{x})|^2 \right) d\mathbf{x}. \tag{2.23}$$

As color and gradients are low level, incorporating more descriptive features such as point correspondences can help emphasize small structures and is formulated as:

$$E_{match}(\mathbf{w}, \mathbf{w}_1) = \int \delta(\mathbf{x})\, \rho(\mathbf{x})\, \Psi \left(|\mathbf{w}(\mathbf{x}) - \mathbf{w}_1(\mathbf{x})|^2 \right) d\mathbf{x}, \tag{2.24}$$

where $\mathbf{w}_1(\mathbf{x})$ denotes the correspondence vectors obtained by descriptor matching at some points $\mathbf{x}$. $\delta(\mathbf{x})$ is an indicator function returning one if a descriptor is available at point $\mathbf{x}$ and zero otherwise. $\rho(\mathbf{x})$ is the matching score and serves as weighting of the point correspondence. The matching task $\mathbf{w}_1 : \Omega \rightarrow \Omega$ is formulated as another energy term to be minimized:

$$E_{desc}(\mathbf{w}_1) = \int \delta(\mathbf{x})\, |\mathbf{f}_2(\mathbf{x} + \mathbf{w}_1(\mathbf{x})) - \mathbf{f}_1(\mathbf{x})|^2 d\mathbf{x}, \tag{2.25}$$

where $\mathbf{f}_1(\mathbf{x})$ and $\mathbf{f}_2(\mathbf{x})$ denote the fields of feature vectors in frame 1 and frame 2, respectively. Gathering all terms together into a single optimization problem; we arrive at the following formulation:

$$E(\mathbf{w}) = E_{color}(\mathbf{w}) + \gamma E_{grad}(\mathbf{w}) + \alpha E_{smooth}(\mathbf{w}) + \beta E_{match}(\mathbf{w}, \mathbf{w}_1) + E_{desc}(\mathbf{w}_1). \tag{2.26}$$

Here, α, β and γ are parameters. The goal is to solve Eqn (2.26) using a continuation method [163] which solves the original problem by solving a sequence of subproblems at

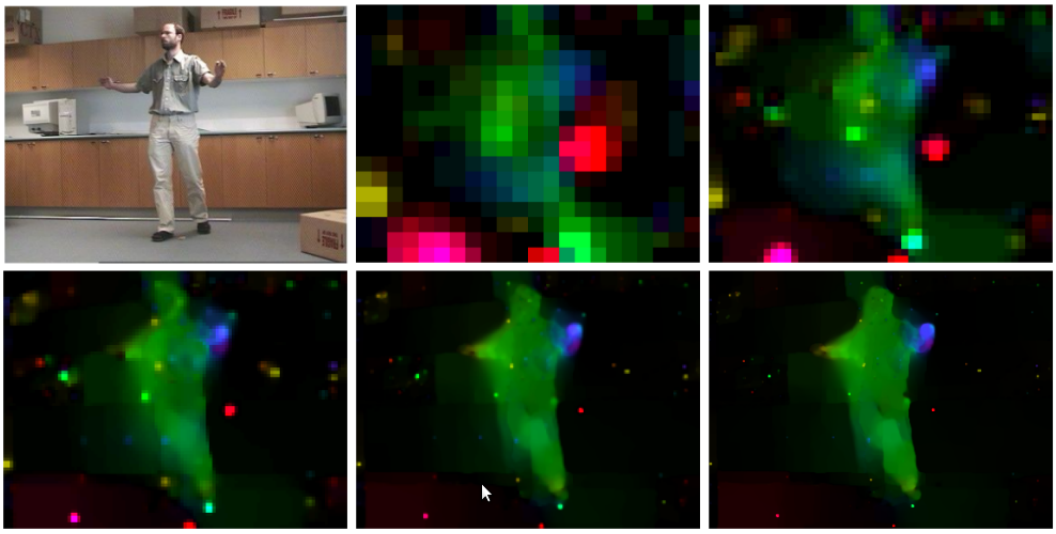


Fig. 2.2: Image of a sequence where the person is stepping forward and moving his hands followed be a sequence of evolving flow field from coarse to fine. The region correspondences dominate the estimate at the beginning. Outliers are removed over time as more and more data from the image are taken into account. Figure adapted from [157].

different levels of resolution by smoothing the input images. As a preprocessing step, keypoint matching between the two images is computed. The confidence of the match is used as the score $\rho(\mathbf{x}_i)$.

It is worth noting that point correspondences are integrated into the continuation method, giving them high impact at the beginning of the process, where the image resolution is very small and the correspondences dominate the color and gradient constancy terms, thereby guiding the solution towards large displacement solutions. As the resolution increases, the ratio between the fixed number of point correspondences and the number of pixels in the image drops, thereby tuning down the impact of point correspondences and relying more on low level color and gradient cues. The evolving flow field is illustrated in Fig 2.2.

CHAPTER 3

Grouped Features for Object Detection

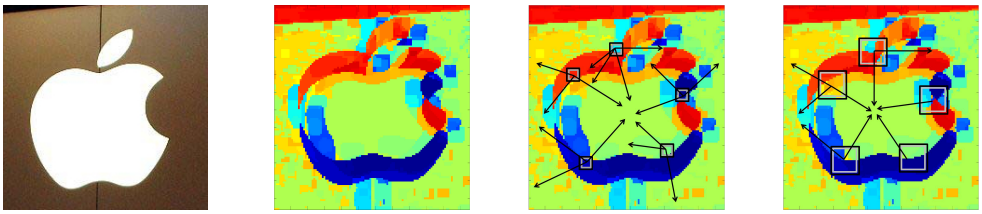
Contents

3.1. INTRODUCTION

As discussed in Section 2.1, Hough-based voting approaches or implicit shape models (ISM) [18, 164, 165] model an object by a codebook of features and their spatial offsets to the center or root of the object. This is also known as a star model. For object detection, features of the test image are matched to the codebook, where each codebook entry models a distribution over the space of object hypotheses. Based on these distributions, each feature votes for an object hypothesis that usually encodes the class and bounding box of the object, but it can also provide additional information like depth [166].

In the past few years, several improvements have been introduced. For instance, a max-margin framework has been proposed to learn the voting discriminatively [19, 167]. In [168], the voting is not performed by points in the scale-space but by lines in order to resolve ambiguities in location and scale. The learning of a codebook has been addressed in [21, 169], where random forests [170, 171] that solve classification and regression problems simultaneously have been introduced. The performance of these methods has been further improved by using self-similarity features [172], enforcing voting consistency by learning several models per object class [14], which is in the spirit of [27], or by training the trees with a global loss function instead of a local one [22].

In this chapter, the independence assumption of the features is addressed. While modeling distributions for each codebook entry independently makes the above methods very efficient, it often yields a low recall if a high precision is required, *i.e.* if the number of false positives needs to be very low. It is therefore beneficial to model the probability for an object hypothesis not conditioned on a single feature but for groups of features in a local neighborhood, as



(a) (b) (c) (d)

Fig. 3.1: Illustration of Hough-based voting with independent (or low level) features. Each pixel of an image (a) is assigned to a codeword. Each color in (b) corresponds to a codeword. For independent features, each pixel votes according to its assigned codebook entry (c). For grouped (or mid level) features, the voting depends also on the assigned codebook entries in its neighborhood (d).

illustrated in Fig 3.1. To this end, a discriminative Hough forest model of an object is initially learned on independent features and a second classification-regression forest is used to learn the probability of a hypothesis for a group of features. This is inspired by the work of Shotton *et al.* [173] that introduces semantic texton forests to solve classification problems like segmentation and image-categorization tasks very efficiently. In the experiments, it is shown that the split functions used in segmentation forests [173] are too weak for the task of object detection. To this end, oblique classification-regression forests are proposed that combine features from different trees and outperform features from a single tree. Also, the benefit of combining independent and grouped features is investigated and the approach is evaluated on four RGB and RGBD datasets.

3.2. RELATED WORK

The grouping of sparse contour features during test time has been addressed in [62]. The approach iteratively estimates the group assignments and optimal affine transformation of the detected groups for voting. Instead of learning groupings from training data, the approach tries to resolve ambiguities in the test image. While this makes the approach suitable for datasets with only a handful of training examples, the bottom-up grouping used is very specific to contour features and not suitable for densely extracted image features that are more generic. Contour features have also been an avenue for attention in other works. In [61], a codebook of recurring contour-pairs is learned from positive examples and an active appearance model for the object boundary is used for detection. During testing, Hough voting is performed by contour-pairs to identify hypotheses. However, the lack of a strong global model requires an additional verification stage [174]. Similarly, [60] ranks contours against a predefined boundary model using a partial matching scheme and allows most promising contours to individually vote for the hypothesis in parametric space. This approach also uses an additional classifier for verification [168]. A similar partial match based approach is used in [175], where the location of individual contours is jointly optimized for generating object hypotheses. More recently, a max-margin framework has been proposed to learn a bag of jointly placed contours that represent the object [176]. During testing, individual contour matches are considered and a joint

placement of all other expected contours is verified against the hypothesis. In [63] an object shape is modeled as a sparse linear combination of contours found in the positive training examples. Furthermore, an explicit part based model is used to facilitate discriminative learning for object detection. A mixed approach of combining contour and appearance information is adapted in [177], where objects are detected by grouping superpixels and jointly optimizing for boundary and appearance terms. What all these methods have in common is that they rely on implicit or explicit shape matching procedures. While these methods perform very well on datasets where the contours of objects can be relatively easily extracted and do not need many training examples, we focus on an approach that is not limited to a very specific set of sparse features.

3.3. HOUGH-BASED OBJECT DETECTION

As discussed in Section 2.1, implicit shape models [18] or Hough-based voting approaches represent an object by features like image patches or contour fragments that appear at certain locations with respect to the object center. These object features are often treated as independent entities during testing and the probability for an object hypothesis $\mathbf{h}$, which encodes the label, position, scale and aspect ratio of an object, is approximated by the sum of the probabilities of the hypothesis for each feature $\mathcal{I}(\mathbf{y})$:

$$p(\mathbf{h}|\mathcal{I}) \approx \sum_{\mathbf{y}\in\Omega} p(\mathbf{h}|\mathcal{I}(\mathbf{y})) \tag{3.1}$$

where $\mathcal{I}(\mathbf{y})$ denotes a feature extracted from image $\mathcal{I}$ at location $\mathbf{y}$. This chapter relaxes the independence assumption of the features by computing the probabilities of the hypothesis for groups of features instead of single features:

$$p(\mathbf{h}|\mathcal{I}) \approx \sum_{\mathbf{y}\in\Omega} p(\mathbf{h}|\mathcal{I}(\mathcal{N}(\mathbf{y}))) \tag{3.2}$$

where $\mathcal{I}(\mathcal{N}(\mathbf{y}))$ denotes the features within a neighborhood of $\mathbf{y}$. This is illustrated in Fig 3.1. Eqn (3.1) is briefly described before discussing Eqn (3.2) in Section 3.3.2.

3.3.1. Independent Features

A Hough forest consists of an ensemble of decision trees. While training a tree, each non-leaf node is assigned a binary test that is applicable to all data samples encountering the node. The sample is directed to either the left or the right child depending on the result of this test. Consequently, each leaf node holds data samples that have been grouped together according to the intent of the preceding binary tests. In order to attain better generalization, each tree is trained randomly, which is achieved by (1) training each tree with a random set of samples from the training data and (2) considering a random subset of possible binary tests for each non-leaf node and choosing the one that results in an optimal split of the incoming data points. Implementation details relevant for Section 3.3.2 are described below.

Training data. Each tree T of a forest is trained on a set of patches $\{\mathcal{P}_i = (\mathcal{I}_i, c_i, \mathbf{d}_i)\}$ that are randomly sampled from the positive and negative example images. Each patch has the same size and $\mathcal{I}_i$ denotes the appearance of the patch, represented by low level features like

color and histograms of gradients. $\mathbf{d}_i$ is the offset vector from the patch center to the object center given by the center of the bounding box if the class label c_i is positive. If the patch has been sampled from a negative example image or from the background of a positive example image, the class label c_i is zero and $\mathbf{d}_i$ is not used.

Splitting function. The splitting functions $t_\theta(\mathcal{I}_i) \rightarrow \{0, 1\}$ at each non-leaf node of the tree split the training data arriving at the node into two sets, where θ is the parameter vector that defines the binary test. In [21], the functions are defined by

$$t_\theta(\mathcal{I}_i) = \begin{cases} 0 & \text{if} \quad I_i^f(\mathbf{p}) - I_i^f(\mathbf{q}) < \tau \,, \\ 1 & \text{otherwise} \end{cases} \tag{3.3}$$

where $I_i^f(\mathbf{p})$ and $I_i^f(\mathbf{q})$ are the values of low level feature f at pixel locations $\mathbf{p}$ and $\mathbf{q}$ of patch $\mathcal{P}_i$. The family of binary tests is therefore defined by $\theta = (f, \mathbf{p}, \mathbf{q}, \tau)$.

Training. The trees are constructed recursively starting from the root node. For a given set of training patches P arriving at a node, the best split function is selected from a random set of generated split functions. The goodness of a split function t_θ, which splits the training patches into two sets $P_0(\theta)$ and $P_1(\theta)$, is measured either by the classification objective

$$\Delta U_1(\theta) = U_1(P) - \sum_{l \in \{0,1\}} \frac{|P_l|}{|P|} U_1(P_l(\theta)), \tag{3.4}$$

where $U_1(\cdot)$ is the entropy measuring class uncertainty, or by a regression objective that minimizes the variance of the offset vectors:

$$\Delta U_2(\theta) = U_2(P) - \sum_{l \in \{0,1\}} \frac{|P_l|}{|P|} U_2(P_l(\theta)), \tag{3.5}$$

where $U_2(\cdot)$ is the variance of the offset distances. As in [21], one of the objectives is randomly selected for each node. The training continues until the maximal depth, 25, is reached or if $|P_0|$ or $|P_1|$ of the best split is below a threshold, 20. At each leaf L_T of tree T, the class probability $p(c|L_T)$ and the distribution of the offset vectors $p(\mathbf{d}|c, L_T)$ for the positive class are stored.

Testing. For object detection, the peaks of $p(\mathbf{h}|\mathcal{I})$ in Eqn (3.1) are detected for different scales s and aspect ratios a. To this end, the image is resized according to scale and aspect ratio and each image patch $\mathcal{I}_\mathbf{y},\ \mathbf{y} \in \Omega$ is passed through all trees $\mathcal{T}$. The probability of a hypothesis for class c and location $\mathbf{x}$ is then given by

$$p(\mathbf{h}(c, \mathbf{x}, s, a)|\mathcal{I}_\mathbf{y}) = \frac{1}{|\mathcal{T}|} \sum_{T \in \mathcal{T}} p(\mathbf{h}(c, \mathbf{x}, s, a)|L_T(\mathbf{y})), \tag{3.6}$$

$$\text{with} \quad p(\mathbf{h}(c, \mathbf{x}, s, a)|L_T(\mathbf{y})) = p(\mathbf{d}(\mathbf{y}, \mathbf{x}, s, a)|c, L_T(\mathbf{y})) \cdot p(c|L_T(\mathbf{y})), \tag{3.7}$$

where $\mathbf{d}(\mathbf{y}, \mathbf{x}, s, a)$ is the scale and aspect ratio normalized offset vector between $\mathbf{y}$ and $\mathbf{x}$.

3.3.2. Grouped Features

The approach to model $p(\mathbf{h}|\mathcal{I}_{\mathcal{N}(\mathbf{y})})$ in Eqn (3.2) instead of independent features in Eqn (3.1) is inspired by semantic texton forests [173]. Texton forests convert pixels into a set of semantic

textons and a second classifier trained on the textons is used to segment or classify an image. In the present context, the probability of a hypothesis based on all leaves within a neighborhood of $\mathbf{y}$ is learned instead of simply averaging the probabilities of all leaves as in Eqn (3.6) as:

$$p(\mathbf{h}|\mathcal{I}_{\mathcal{N}(\mathbf{y})}) = p(\mathbf{h}|\{L_T(\mathcal{N}(\mathbf{y}))\}_{T\in\mathcal{T}}). \tag{3.8}$$

In order to learn the probability, a second forest that is trained on the patch-to-leaf assignments obtained from the original forest is employed. To this end, the approach discussed in Section 3.3.1 is modified suitably:

Training data. Instead of training a forest on patches of low level image features, the forests are now trained on histograms of leaves (HOL). This mid level representation pools features originally designed to serve a high level task, as learned by the first classifier, over a local neighborhood. While the class label c_i and the offset vector $\mathbf{d}_i$ of a group of features $\mathcal{G}_i = (\text{HOL}_i, c_i, \mathbf{d}_i)$ are the same as in Section 3.3.1, histograms of leaves consist of a histogram for each tree HOL_T where the entries are given by $L_T(\mathcal{N}(\mathbf{y}))$, *i.e.* by the leaves within a rectangular region around image location $\mathbf{y}$. The histograms are normalized such that $\sum_{L\in T} \text{HOL}_T(L) = 1$.

Splitting function. As splitting functions, we investigate two families. In [173], the value of a single bin of a histogram is used as test function:

$$f_\theta(HOL) = \begin{cases} 0 & \text{if} \quad \text{HOL}_T(L_T) < \tau\,, \\ 1 & \text{otherwise} \end{cases} \tag{3.9}$$

where $\theta = (T, L_T, \tau)$. While T selects the histogram HOL_T, L_T is the index of one bin in the histogram. Because this family of splitting functions is shown to be not powerful enough for the task at hand, a larger family of split functions is proposed:

$$f_\theta(HOL) = \begin{cases} 0 & \text{if} \quad \sum_{T\in\mathcal{T}} w_T \cdot \text{HOL}_T(L_T) < \tau\,, \\ 1 & \text{otherwise} \end{cases} \tag{3.10}$$

with $\theta = (\{w_T\}_{T\in\mathcal{T}}, \{L_T\}_{T\in\mathcal{T}}, \tau)$. The real-valued weights w_T combine the features from different trees resulting in so-called oblique forests [171].

Training and Testing. The training is performed as in Section 3.3.1. For testing, the first forest is applied to all scales and aspect ratios in order to assign each image patch to some leaves $\{L_T\}_{T\in\mathcal{T}}$. The probability of an object hypothesis $\mathbf{h}$ is then given by the two forests stacked together:

$$p(\mathbf{h}(c,\mathbf{x},s,a)|\mathcal{I}_{\mathcal{N}(\mathbf{y})}) = p(\mathbf{h}(c,\mathbf{x},s,a)|\{L_T(\mathcal{N}(\mathbf{y}))\}_{T\in\mathcal{T}}) \tag{3.11}$$

$$= \frac{1}{|\mathcal{T}_{gr}|} \sum_{T_{gr}\in\mathcal{T}_{gr}} p(\mathbf{h}(c,\mathbf{x},s,a)|L_{T_{gr}}(\{L_T(\mathcal{N}(\mathbf{y}))\}_{T\in\mathcal{T}})), \tag{3.12}$$

where $\mathcal{T}$ is the first forest with independent features and $\mathcal{T}_{gr}$ is the second forest. The approach is illustrated in Fig 3.1 for a single tree.

Further, a version where the probabilities for the hypotheses of independent features as in Eqn (3.6) and the grouped features as in Eqn (3.11) are combined is also investigated:

$$p(\mathbf{h}(c,\mathbf{x},s,a)|\mathcal{I}_{\mathbf{y}},\mathcal{I}_{\mathcal{N}(\mathbf{y})},\lambda) \propto p(\mathbf{h}(c,\mathbf{x},s,a)|\mathcal{I}_{\mathcal{N}(\mathbf{y})})^{\lambda} \cdot p(\mathbf{h}(c,\mathbf{x},s,a)|\mathcal{I}_{\mathbf{y}})^{1-\lambda} \tag{3.13}$$

where the parameter $\lambda \in [0,1]$ steers the impact of the two probabilities.

support:	7×7			13×13		
depth:	5	10	16	5	10	16
Applelogos	55.0/60.0	**90.0/90.0**	75.0/75.0	15.0/15.0	75.0/75.0	10.0/10.0
Bottles	**92.8/92.8**	89.2/89.2	75.0/75.0	57.1/57.1	71.4/71.4	32.2/32.2
Giraffes	72.3/74.5	78.7/80.8	83.0/83.0	61.7/61.7	**83.0/85.1**	74.5/74.5
Mugs	61.3/61.3	**67.7/74.2**	61.3/61.3	45.2/45.2	61.3/61.3	51.6/51.6
Swans	70.6/76.5	70.6/70.6	58.8/58.8	58.8/58.8	**82.3/88.2**	41.2/58.8

Table 3.1: Recall for ETHZ dataset at 0.3/0.4 FPPI (%) for given spatial support $\mathcal{N}$ and depth of the first forest to generate histograms of leaves (HOL).

3.4. EXPERIMENTS

The effectiveness of the proposed grouped features is evaluated on four different datasets of increasing difficulty and its performance is compared with previously published results. In each case, the evaluation protocols of previous works is adopted to facilitate comparison. Hypotheses are classified according to the PASCAL-VOC criterion [2] and performance is quantified using false positives per image [178] (FPPI) and average precision [2] (AP) measures.

ETHZ Dataset. The dataset consists of 255 images classified into five categories: *Applelogos, Bottles, Giraffes, Mugs, Swans*. *Giraffes* is the largest class consisting of 87 examples and *Swans* is the smallest consisting of 32 examples. There are two protocols [19] and [61] for this dataset. We use the protocol as in [19] where half of the images of a class are taken as positive examples for training and an equal number of negative examples is taken from the other classes. The other images are used for testing. The baseline with independent features is trained using 5 trees with maximal depth of 25. After normalizing positive examples to unit scale and aspect ratio, all patches have a fixed size of 16×16 pixels and each tree is trained on 20,000 positive and 20,000 negative patches. Half of the negative patches are drawn from the background of positive examples and the other half from the negative examples. Patch features consist of 15 feature channels [21]: 6 color channels obtained by the Lab color space processed by a 5×5 min- and max- filter and 9 gradient features obtained by 9 HOG bins

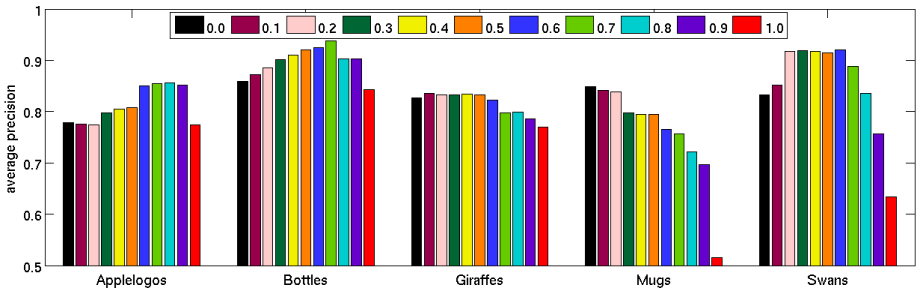


Fig. 3.2: Average Precision plots for varying parameter λ as in Eqn (3.13) for each class of the ETHZ dataset. The *black* and *red* bars correspond to independent and group features, respectively. While for Applelogos, Bottles and Swans the combination gives a clear improvement for a value around 0.6, the combination does not work well for Giraffes and Mugs.

using a 5×5 cell and soft binning. Testing is performed by spanning three aspect ratios and five scales.

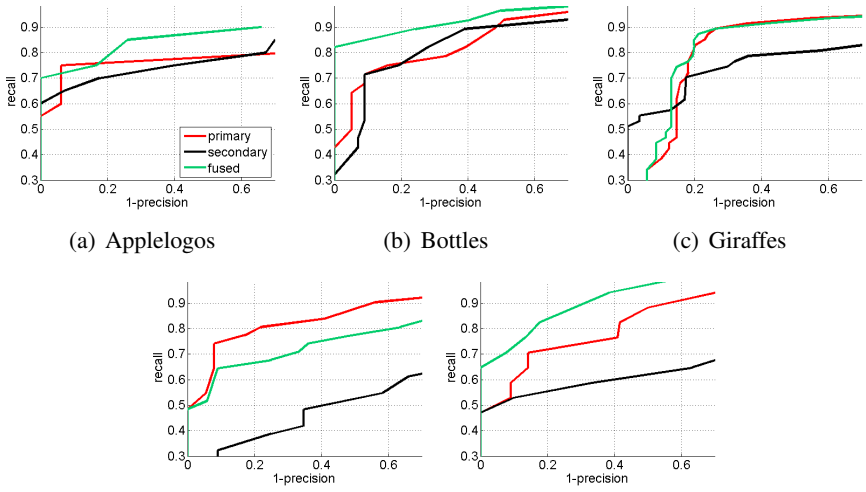
Object detection with grouped features depends on the spatial support of the neighborhood $\mathcal{N}$, but also on the depth of the first forest to generate the HOL. To study the effect of both parameters, we used tree depth $\in \{5, 10, 16\}$ and spatial support $\in \{7 \times 7, 13 \times 13\}$ for computing HOL. The resulting performance for each case is shown in Table 3.1. The tree depth is very important since for very deep trees the leaves become highly specific causing a drop in performance. The neighborhood size also has an impact on the performance. Since the configuration pair $\{10, 7 \times 7\}$ performs reasonably for all classes, it is used for all experiments henceforth.

Further, the combination according to Eqn (3.13) is studied. Fig 3.2 presents results for all classes as a function of parameter λ. The combination performs well only for three out of five classes. This is mainly due to the small size of the dataset (cf. VOCB3DO dataset below). Fig 3.3 and Table 3.2 also compare independent features, grouped features, and the best combination using different measures.

In addition, comparison with state-of-the-art methods is presented in Tables 3.3 and 3.4. Although the proposed method does not perform optimally for all classes due to the very small amount of training data, the performance is comparable to state-of-the-art methods, which either involve an additional verification step or are specifically tailored to contour features suited for this shape dataset.

INRIA Horse Dataset. The dataset consists of 170 positive and 170 negative examples; of which the first 50 in each case are used for training. The baseline is made up of 5 trees, each trained with 40,000 positive patches and 40,000 negative patches. Furthermore, hard-negative training is performed by mining the 50 hardest negative examples in the training data. The hard-negatives contributes to an additional gain of 8% recall at 0.3 FPPI.

Three types of splitting functions are investigated on this dataset. Firstly, axis aligned



(a) Applelogos (b) Bottles (c) Giraffes

(d) Mugs (e) Swans

Fig. 3.3: Precision-recall plots for (red) independent features, (black) group features and (green) the best combination. As seen in Fig 3.2, the performance varies among classes.

	Average Precision			optimal λ	Recall at 0.3/0.4 FPPI		
	(3.6)	(3.11)	(3.13)		(3.6)	(3.11)	(3.13)
Applelogos	77.8	77.4	85.4	0.9	80.0/80.0	90.0/90.0	90.0/90.0
Bottles	85.9	84.3	93.8	0.7	92.9/96.4	89.2/89.2	96.4/96.4
Giraffes	82.6	76.9	83.4	0.1	91.5/93.6	78.7/80.8	91.5/91.5
Mugs	84.9	62.6	84.1	0.1	90.3/90.3	67.7/74.2	87.1/90.1
Swans	83.2	63.3	90.2	0.6	100/100	70.6/70.6	100/100

Table 3.2: Performance comparison for ETHZ dataset wrt independent features Eqn (3.6), grouped features Eqn (3.11) and their combination Eqn (3.13)

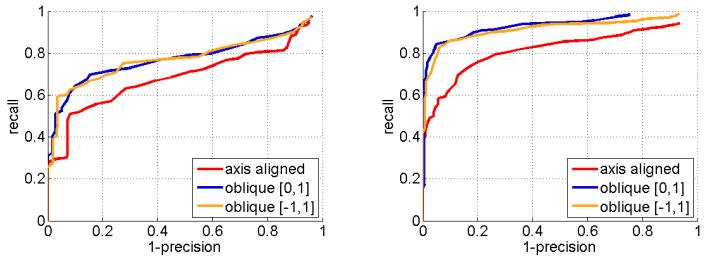
	Proto.	Verific.	Applelogos	Bottles	Giraffes	Mugs	Swans
Ours	[19]	N	90.0/90.0	96.4/96.4	91.5/91.5	90.3/90.3	100/100
[19]	[19]	Y	95.0/95.0	92.9/96.4	89.6/89.6	93.6/96.7	88.2/8.2
[177]	[19]	Y	100/100	96.0/97.0	86.0/91.0	90.1/91.0	98.0/100
[63]	[19]	Y	95.0/95.0	100/100	87.2/89.6	93.6/93.6	100/100
[27]	[61]	N	95.0/95.0	96.3/100	84.7/84.7	96.7/96.7	94.1/94.1
[27]	[61]	N	95.0/95.0	100/100	72.9/72.9	83.9/83.9	58.8/64.7
[176]	[61]	N	95.0/95.0	100/100	91.3/91.3	96.7/96.7	100/100
[175]	[61]	N	92.0/92.0	97.9/97.9	85.4/85.4	87.5/87.5	100/100
[60]	[61]	Y	93.3/93.3	97.0/97.0	79.2/81.9	84.6/86.3	92.6/92.6
[168]	[61]	Y	95.0/95.0	89.3/89.3	70.5/75.4	87.3/90.3	94.1/94.1
[61]	[61]	Y	77.7/83.2	79.8/81.6	39.9/44.5	75.1/80.0	63.2/70.5

Table 3.3: Comparing performance for ETHZ dataset wrt Eqn (3.13) with state-of-the-art methods (recall at 0.3/0.4 FPPI). Note that there are two different protocols [19] and [61]. The proposed approach based is comparable to state-of-the-art without requiring an additional verification stage

	Proto.	Verific.	Applelogos	Bottles	Giraffes	Mugs	Swans	mean
Ours	[19]	N	85.4	93.8	83.4	84.9	90.2	87.5
[63]	[19]	Y	84.5	91.6	78.7	88.8	92.2	87.2
[175]	[61]	N	88.1	92.0	75.6	86.8	95.9	87.7
[168]	[61]	Y	86.9	72.4	74.2	80.6	71.6	71.1
[27]	[61]	N	89.1	95.0	60.8	72.1	39.1	71.2
[176]	[61]	N	-NA-	-NA-	-NA-	-NA-	-NA-	88.2

Table 3.4: Comparison of Eqn (3.13) with state-of-the-art methods (Average precision) for ETHZ dataset

forests that employ splitting functions as Eqn (3.9). Secondly, oblique forests that employ



(a) INRIA Horse Dataset (b) Weizmann Dataset

Fig. 3.4: Precision-recall plots showcasing the performance of three types of splitting functions for grouped features. The oblique forests as per Eqn (3.10) outperform axis aligned forests as per Eqn (3.9).

splitting functions as Eqn (3.10) with weights $w_T \in [0, 1]$ and lastly, oblique forests with weights $w_T \in [-1, 1]$. Performance of each of the three variants is shown in Fig 3.4(a). While the oblique forests outperform Eqn (3.9), also allowing negative weights does not change the performance. Table 3.5 compares combining grouped and individual features ($\lambda = 0.1$) with state-of-the-art methods.

Weizmann Horse Dataset. The dataset comprises 200 and 456 images for training and testing respectively. For the independent features, a forest of 5 trees with 20,000 positive and 20,000 negative patches followed by hard-negative mining is trained. As for the INRIA horse dataset, performance of the three types of splitting functions is shown in Fig 3.4(b). The oblique forests again outperform the axis-aligned forests. Table 3.6 compares combining grouped and individual features ($\lambda = 0.1$) with state-of-the-art methods.

Berkeley 3-D Object Dataset. The dataset is a collection of real-world images captured with a Kinect sensor comprising RGBD image pairs for over 50 classes. A six-fold split is predefined for 8 of these classes, resulting in 48 splits, over which baseline performances using a part based model [27] for various RGB, D and RGBD features are presented in [57]. Interestingly, the authors report a significant drop in performance upon including depth information.

The forests of the independent features are ensembles of 10 trees, each trained with 100,000 positive patches and an equal number of negative patches without any hard negative training. The utility of depth channel is investigated by comparing forests built on independent RGB-only features ignoring depth information, and independent RGBD features where the depth of a pixel is used as an additional feature. The results are tabulated in Table 3.7. A significant improvement of RGBD over RGB features is observed in contrast to [57].

The forest on grouped features is based on an oblique forest with $w_T \in [0, 1]$. Each forest consists of 10 trees and is trained with the same protocol as for independent features. The performance of grouped features is at most equal to that of independent features with an exception for *bottles*, where a gain of 0.5% in average precision is seen. Further, both forests also combined as in Eqn (3.13) by fixing the parameter λ for each split using a validation dataset, which is obtained by splitting the training data in half. Table 3.7 presents performances of the best possible combination, the value of λ set using validation and the resulting performance. It is to be noted that combining both forests mostly results in an improved performance, indicating that although the grouped features alone do not outperform their individual counterparts,

they contain complementary information.

The approach based on Eqn (3.13) with λ estimated on the validation set achieves an average precision of 0.314 and is comparable or better than the state-of-the-art methods [57] and [119], which achieve 0.280 AP and 0.312 AP, respectively. The approach [38] reports 0.592 AP, but it follows a different evaluation procedure by using custom annotation of the dataset. Qualitative results of detections from individual and grouped features from the various datasets are shown in Fig 3.5.

3.5. SUMMARY

An approach for mid level grouping features for Hough-based object detection in RGB and RGBD images has been presented. Evaluation is based on four datasets of various difficulties, where a performance comparable to state-of-the-art methods has been achieved. Hard negative training for independent features improves performance for two datasets albeit at the cost of doubled training time. It is observed that highly specific independent features and small datasets adversely affect performance in Eqn (3.13). Also, the oblique forests for grouped features outperform axis-aligned forests. While the grouped features do not perform well for all classes of the ETHZ dataset with very few training examples, they outperform independent features on the more realistic VOCB3DO dataset. The experiments also show that a combination of independent and grouped features improves performance, indicating that both feature sets encode complementary information.

	proposed	[176]	[177]	[62]	[19]
recall	88.0	93.7	92.4	87.3	85.3

Table 3.5: Recall at 1.0 FPPI for INRIA Horse dataset

	proposed	[21]	[179]
AP	97.2	98.0	96.0
recall	94.3	95.1	91.5

Table 3.6: Recall at 1.0 FPPI for Weizmann Horse dataset

Class	RGB	RGBD	Group	bestComb.	λ	Combin.	[57]
bowl	0.231	0.402	0.394	0.423	0.5	0.420	**0.430**
cup	0.123	0.346	0.339	0.358	0.5	**0.357**	0.260
monitor	0.282	0.540	0.530	0.547	0.4	0.547	**0.750**
mouse	0.208	0.282	0.275	0.302	0.4	**0.301**	0.190
phone	0.076	0.163	0.129	0.172	0.3	0.163	**0.180**
keyboard	0.085	0.314	0.283	0.321	0.4	**0.321**	0.170
chair	0.028	**0.208**	0.161	0.211	0.4	0.206	0.140
bottle	0.022	0.178	0.183	0.201	0.2	**0.195**	0.120

Table 3.7: Average precision for the VOCB3DO Dataset

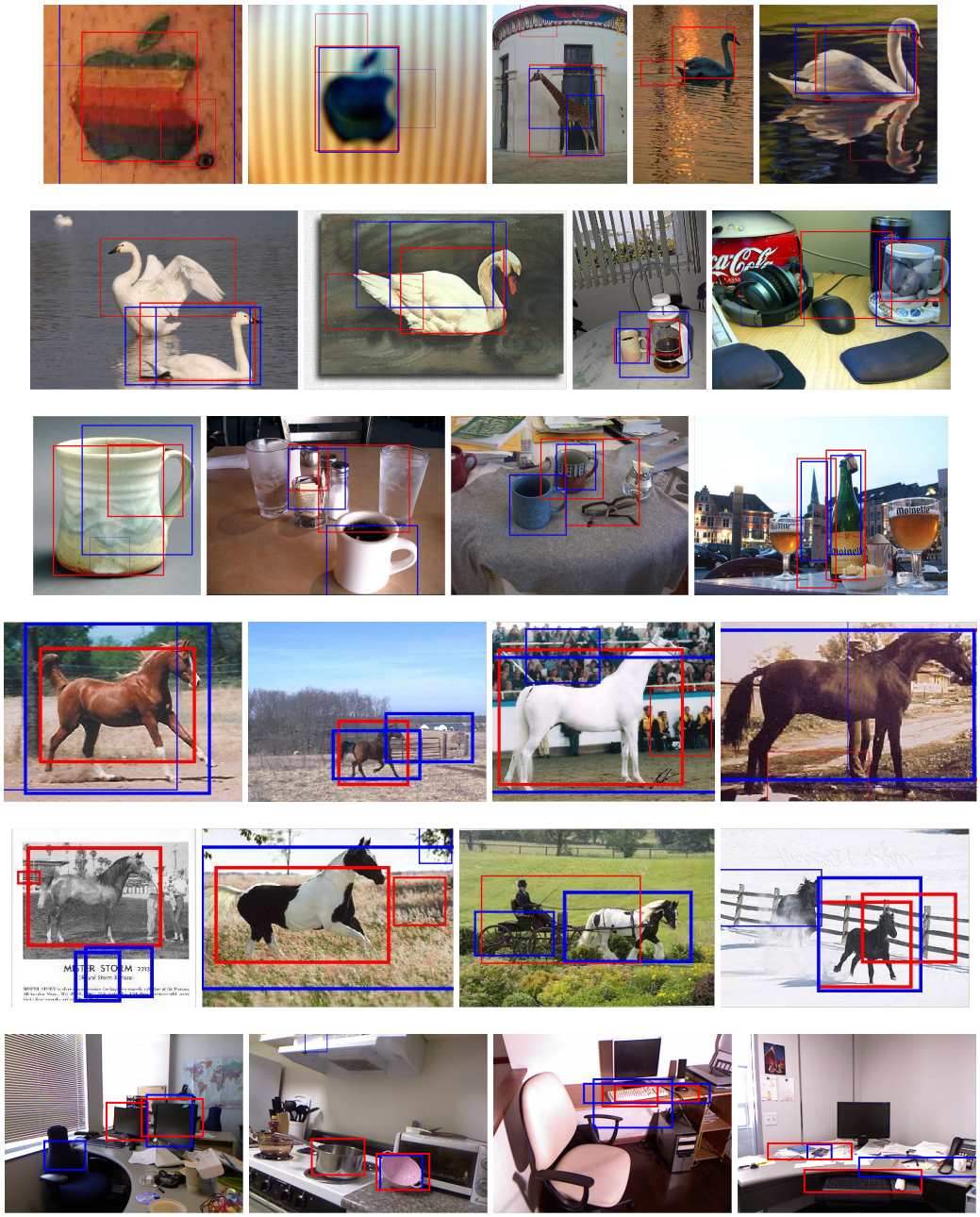


Fig. 3.5: Detections from (blue) individual and (red) group features on various datasets. (rows 1–3:) ETHZ Dataset (rows 4–5:) INRIA and Weizmann Horse (row 6:) VOCB3DO dataset.

CHAPTER 4

Using Human Pose as Context for Object Detection

Contents

4.1. INTRODUCTION

The previous chapter investigated improved appearance based representations using a star model. To this end, mid level features based on grouping low level features in a local neighborhood was proposed. It was shown that information encoded by both features was complementary and that improved object detection performance could be achieved by combining them. Nonetheless, appearance based object detection is still an open problem [3, 82] for small and medium sized objects where visual evidence become unreliable due to poor resolutions and frequently occurring human interactions. This introduces new challenges as objects are heavily occluded and undergo large pose and appearance variations during the process. Therefore, this chapter investigates the advantages of adopting part based models and utilizing high level human pose as a contextual cue for object detection.

The context of human-object interactions has been adopted by numerous recent methods [35, 180, 181, 182, 183, 184]. For instance, [35] extends a deformable part model (DPM) [26] to model spatial relations between body parts and parts of objects. This approach, however, only works well for images showing the instant of human-object interaction, *i.e.*,

when a human is closely in contact with an object. For images without an interaction, pose and objects are independently modeled, e.g., by having several models including either object or pose, or both together. In such cases, the additional information from human context is therefore not exploited.

This chapter investigates an approach that includes human pose as an additional context for object detection. The approach is not limited to images showing explicit human-object interactions, but also works for general images where human pose can be inferred. For instance, a pose related to emptying a tin indicates that a tin opener might be close although the person does not use the tin opener at this moment. To this end, objects are represented by a part based model where location of a part is predicted from both image and human pose data using regression forests.

The experiments show that jointly modeling human and object as in [35] leads to suboptimal performance for object detection. On the other hand, the proposed approach which has flexibility to incorporate potential gains from either modality is successfully demonstrated on three datasets [154, 180, 185] that have varying quality of automatically extracted 2d or 3d human pose. Further, the effect of various human pose estimation techniques on object detection accuracy is investigated. An outline of the approach is presented in Figure 4.1.

4.2. RELATED WORK

Combining humans and objects together to address various problems in computer vision has received considerable attention in the recent past. [86] builds a discriminative model for action classification by reasoning about spatial co-occurrences of body parts and objects. In [186] a weakly supervised approach is proposed for action classification that does not require annotations of objects and humans in training images. Human context has also been used to deduce object functionality either by inferred [187] or by hypothesized human pose [181].

As for methods relating to object detection, [182] proposes a generative model that combines body part trajectories and object appearance. However, it uses strictly handcrafted metrics to tap human motion information which can be difficult to adapt to realistic actions. [183] learns a discriminative random field model by representing body part location priors as nodes and spatial relations between body parts and objects as edges. However, mixture models are treated independently resulting in poor performance for complex data. In this regard, [184] introduces a coarse-to-fine hierarchical grammar for a more concise representation of mixture models. Introducing phraselets, [35] extends a DPM [188] to improve the quality of mixtures by clustering training examples based on their relative locations. The method reports state-of-the-art results for joint pose estimation, action classification and object detection.

4.3. OBJECT DETECTION

As illustrated in Figure 4.1(f), an object is represented through a tree model by a set of descriptive keypoints $\mathcal{K} = \{\mathbf{k}_i\}$ where $\mathbf{k}_i \in \mathbb{R}^2$ encodes the image location of the i^{th} keypoint. As in the pictorial structures model [188], the spatial relations between them are defined by a directed graph $\mathcal{E}$ and the prior on any keypoint configuration is given by

$$p(\mathcal{K}) = \prod_{i,j \in \mathcal{E}} \psi_{ij}(\mathbf{k}_i, \mathbf{k}_j), \tag{4.1}$$

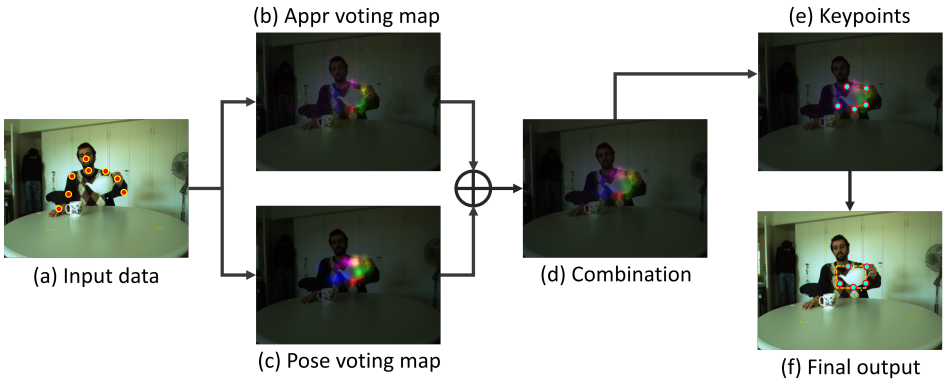


Fig. 4.1: Detecting teapots: (a) Input is an image and automatically extracted human pose. (b) Object keypoint unaries based on appearance features and (c) Keypoint unaries based on human pose features. Note the reduced keypoint localization capability. (d) Linear combination of unaries. (e) Inferring keypoints using the pictorial structures model. (f) Regressing object bounding box using the inferred keypoints.

where the pairwise potentials $\psi_{ij}(\mathbf{k}_i, \mathbf{k}_j)$ model spatial relations between two keypoints $\mathbf{k}_i$ and $\mathbf{k}_j$. Given an observation $\mathcal{D}$, an optimal configuration is estimated by the maximum of the posterior distribution

$$\begin{aligned} p(\mathcal{K}|\mathcal{D}) &\propto p(\mathcal{D}|\mathcal{K}) \cdot p(\mathcal{K}) \\ &\propto \prod_i \phi_i(\mathbf{k}_i) \cdot \prod_{i,j \in \mathcal{E}} \psi_{ij}(\mathbf{k}_i, \mathbf{k}_j) \end{aligned} \tag{4.2}$$

While 3-mixture Gaussians are used as pairwise potentials to model relative keypoint offsets in the star structured graph $\mathcal{E}$ for efficient inference as in [188], this chapter focuses on extracting more discriminative unary potentials $\phi_i(\mathbf{k}_i)$ derived from appearance and human pose features. The unary potentials will be discussed in Section 4.4.

A bounding box (x_1, y_1, x_2, y_2) must be predicted from the inferred keypoint configuration $\mathcal{K}$ since evaluation is based on the PASCAL-VOC [2] criterion. To this end, a mixture of linear least squares regressors is used to predict each parameter of the bounding box independently. For the regression, keypoint locations are normalized such that the mean becomes zero and variance one. A mixture of 3 regressors is used, each of which is trained on a cluster of training data. As for the feature vector for clustering, the aspect ratio to the normalized keypoints is added resulting in a $(2|\mathcal{K}| + 1)$ dimensional vector. The aspect ratio is calculated using the smallest rectangle enclosing all keypoints.

The inference procedure results in multiple overlapping detections for each object instance. Therefore, redundant detections are eliminated using a greedy approach. Given an image, a set of detected bounding boxes and their respective scores $p(\mathcal{K}|\mathcal{D})$ is obtained. The set is sorted according to the score and all bounding boxes that have an intersection-over-union (IOU) ratio over 0.5 with a higher-scoring bounding box are discarded.

The unary potentials $\phi_i(\mathbf{k}_i)$ in Eqn (4.2) are modeled by probabilities over keypoint location $\mathbf{k}_i$. The probabilities are estimated from two modalities, namely the object appearance $\mathcal{D}_A$ and the human pose $\mathcal{D}_P$, *i.e.*

$$\phi_i(\mathbf{k}_i) = p(\mathbf{k}_i|\mathcal{D}_A, \mathcal{D}_P). \tag{4.3}$$

As random forests are used as regressors, they are briefly introduced in Section 4.4.1. Sections 4.4.2, 4.4.3 and 4.4.4 then present unary potentials based on individual features and their combination.

4.4.1. Random Forests

Hough forests, as discussed in Section 2.1, are used for object detection. However, instead of voting for the center of the bounding box, the random forests are used to predict keypoints of an object. These keypoints are then used to infer object bounding box as described in Section 4.3. A tree T in a forest $\mathcal{T}$ is built from a random subset $P = \{\mathcal{P}_i\}$ of the training data. For each training image, features $\mathcal{I}_i$ are extracted. For training a tree, the set P is recursively divided into two subsets P_0 and P_1 using a binary split function $t_{\hat{\theta}}(\mathcal{I}_i) \rightarrow \{0, 1\}$. The split function, which maximizes the information gain $\Delta U_*(P, \theta)$, is chosen from a pool of randomly generated split functions:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, \Delta U_*(P, \theta) \tag{4.4}$$

$$\Delta U_*(P, \theta) = U_*(P) - \sum_{s\in\{0,1\}} \frac{|P_s(\theta)|}{|P|} U_*(P_s(\theta)), \tag{4.5}$$

where U_* is randomly chosen to be the class entropy or the squared error of the predicted mean. The best split function is stored at the node and the training continues recursively until the maximum depth of the tree is reached or the number of samples in a node falls below a threshold. Incoming training data P is stored at the leaves.

4.4.2. Appearance Features

The case when keypoints are predicted from image data is first considered. In this case, an observation $\mathcal{D}_A = \{\mathcal{P}_a\}$ consists of a set of image patches. The image features $\mathcal{I}_a$ are as described in Section 3.4, *i.e.* , they consist of 15 feature channels: 6 color channels obtained by the Lab color space processed by a 5×5 min- and max- filter and 9 gradient features obtained by 9 HOG bins using a 5×5 cell and soft binning.

To train a forest for each keypoint, patches are sampled from training images where patches within a radius of 100 pixels are considered as positive examples and as negative examples otherwise. Each patch is further augmented with a binary class label c and in case of a positive patch the scale s of the object and the offset $\mathbf{d}$ to the keypoint are also stored. The splitting functions used are pixel comparisons as in Section 2.1:

$$t_\theta(\mathcal{I}_a) = \begin{cases} 0 & \text{if} \quad \mathcal{I}_a^f(\mathbf{p}) - \mathcal{I}_a^f(\mathbf{q}) < \tau\,, \\ 1 & \text{otherwise} \end{cases} \tag{4.6}$$

where parameters $\theta = (f, \mathbf{p}, \mathbf{q}, \tau)$ are described by coordinates $\mathbf{p}$ and $\mathbf{q}$ within the patch, the selected feature $f \in \{1, 2, \cdots, 15\}$ and a threshold τ. In Eqn (4.5), the splitting functions are either based on optimizing the classification or regression criterion:

$$U_1(P) = -\sum_c p(c|P) \ \log(p(c|P))$$
$$U_2(P) = \frac{1}{|D_+^P|} \sum_{\mathbf{d} \in D_+^P} \left\| \mathbf{d} - \frac{1}{|D_+^P|} \sum_{\mathbf{d} \in D_+^P} \mathbf{d} \right\|^2, \tag{4.7}$$

where D_+^P is the set of offsets of positive patches. At the leaves, class probabilities $p(c|L)$, distributions of the offset vectors with respect to a quantized scale $\hat{s}$ and keypoint class c, *i.e.* $p(\mathbf{d}|c, \hat{s}, L)$, are stored. The unary potential based on appearance for a given scale $\hat{s}$ is then defined by

$$\phi_i^A(\mathbf{k}_i, \hat{s}) = \sum_{\mathbf{y} \in \Omega} \frac{1}{|\mathcal{T}_i|} \sum_{T \in \mathcal{T}_i} p(\mathbf{k}_i - \mathbf{y}|c, \hat{s}, L_T) \cdot p(c|L_T), \tag{4.8}$$

where $\mathcal{T}_i$ is the forest trained for the i^{th} keypoint and Ω is a set of locations in the image.

In contrast to [21], training examples are not scaled to a fixed object size since this requires performing object detection over several scaled versions of the test image. Instead, scale of objects in training images is stored in the leaves and a test image is processed at a resolution as is. The unaries $\phi_i^A(\mathbf{k}_i, \hat{s})$ are therefore modeled for pixel location $\mathbf{k}_i$ and scale $\hat{s}$. The keypoint configuration $\mathcal{K}$ is then inferred as per Eqn (4.2) for each scale independently.

4.4.3. Human Pose Features

When the keypoints are predicted from automatically extracted 2d or 3d human pose, the observation $\mathcal{D}_P$ are pose features $\mathcal{I}_p$ are based on joint locations $\mathbf{j}_m$ as in [189], *i.e.*, for all joint combinations the Euclidean distance between two joints is computed and for all quadruples of joints the normal plane feature and the velocity feature are used. The features are illustrated in Fig 4.2.

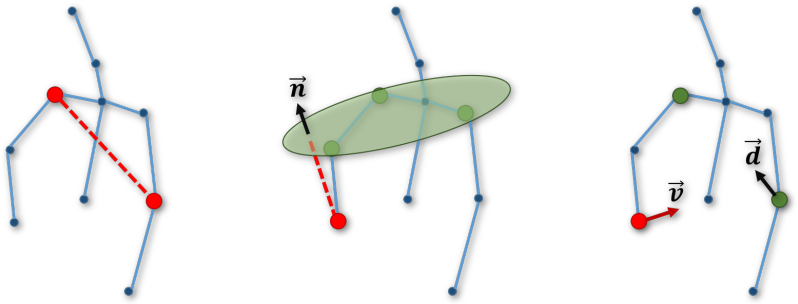


Fig. 4.2: Human pose based features used for object detection: (left) 2d or 3d distance between two randomly chosen joint locations, shown in red (middle) perpendicular distance from a joint, shown in red, to the plane defined by randomly chosen joints, shown in green (right) projection of the velocity of a joint, shown in red, along the displacement between two randomly chosen joints, shown in green.

To train a forest for each keypoint, training images with the object of interest are considered as positive examples and as negative examples otherwise. For each image, pose is augmented with a binary class label c. The positive examples are further augmented with scale s of the object and offsets $\mathbf{d}_m$ from all joints to the keypoint. The splitting functions are defined by

$$t_\theta(\mathcal{I}_p) = \begin{cases} 0 & \text{if} \quad \mathcal{I}_p^f < \tau \,, \\ 1 & \text{otherwise} \end{cases} \tag{4.9}$$

where $\mathcal{I}_j^f$ is a randomly chosen pose feature. The splitting functions are selected as in Eqn (4.7).

Besides class probabilities $p(c|L)$, the distributions of offset vectors with respect to a quantized scale $\hat{s}$ and keypoint class c for each joint m, *i.e.* , $p_m(\mathbf{d}_m|c, \hat{s}, L)$, are stored at the leaves. The unary potential based on pose for a given scale $\hat{s}$ is then defined by

$$\phi_i^P(\mathbf{k}_i, \hat{s}) = \sum_{m=1}^{M} \frac{1}{|\mathcal{T}_i|} \sum_{T \in \mathcal{T}_i} p_m(\mathbf{k}_i - \mathbf{j}_m|c, \hat{s}, L_T) \cdot p(c|L_T), \tag{4.10}$$

where $\mathcal{T}_i$ is the forest trained for the i^{th} keypoint and M is the number of joints.

4.4.4. Combining Appearance and Pose

The unary potential in Eqn (4.2) is a linear combination of the filtered unaries discussed in Sections 4.4.2 and 4.4.3:

$$\phi_i(\mathbf{k}_i, \hat{s}) = \left(K(\sigma_A) * \phi_i^A(\mathbf{k}_i, \hat{s})\right) + \lambda \left(K(\sigma_P) * \phi_i^P(\mathbf{k}_i, \hat{s})\right), \tag{4.11}$$

where $*$ represents the convolution operation and σ is the standard deviation for the Gaussian blur kernel K. Since the human pose can only provide a rough prior for the location of an object class but is insufficient for accurate object localization, $\sigma_P > \sigma_A$. The parameters λ, σ_A and σ_P are estimated by cross-validation.

4.5. EXPERIMENTS

The proposed approach is evaluated on three datasets: ETHZ-Activity [154], CAD-120 [180] and MPII-Cooking [185]. Human pose is inferred in all three datasets using different methods. ETHZ-Activity uses a model based method to extract 3d joint locations of the upper body, CAD-120 uses the OpenNI SDK to extract 3d full body joint locations and MPII-Cooking uses a PS model to extract the 2d joint locations for the arms.

The datasets collectively represent a rich variety of human-object interactions. *E.g.*, elementary interactions are captured in ETHZ-Activity, multi-object interactions in MPII-Cooking and CAD-120 also captures varying viewpoints. There is also a diversity in objects ranging from large to small and from rigid to deformable. The amount of occlusion also varies and the objects are sometimes barely visible. Figure 4.3 shows some cropped representative images. The ground truth of 5 keypoints for each object in the three datasets was manually labeled for every 10^{th} frame of the training data[1].

[1] Annotations can be found at `http://ps.is.tue.mpg.de/person/srikantha`

Evaluation is based on the PASCAL-VOC measure [2] that considers a detected bounding box as true positive when the IOU ratio with the ground truth bounding box exceeds 0.5. Multiple detections overlapping with a true positive are counted as false positives. Performance is reported as area under the precision-recall curve (AUC or AP) where the precision at any recall level r is replaced by the maximum precision measured at recall levels exceeding r as in [2].

Implementation details are presented in Section 4.5.1 followed by the evaluation on the three datasets in Sections 4.5.2– 4.5.4.

4.5.1. Implementation Details

Random Forests: A forest consists of 4 trees with a maximum depth of 25. A tree based on appearance features is trained with 100,000 positive and negative 16×16 sized image patches each and contains at least 20 samples in a leaf. At each node, a pool for splitting functions is generated by randomly choosing 10 thresholds τ and 100 combinations for other parameters in θ. A tree based on human pose features is trained with all positive and negative examples and contains at least 10 samples in a leaf. The pool of splitting functions is generated by randomly choosing 80 parameters and 8 thresholds. The pairwise potentials in Eqn (4.2) are modeled by a mixture of 3 Gaussians.

Setting parameters: The proposed method has three parameters as per Eqn (4.11). These parameters are set by grid search on the validation dataset which is obtained by splitting the training data in half. The search was σ_A and $\sigma_P \in \{5, 11, 41, 161\}$ and $\lambda \in \{0, 100, 250, 500\}$. Generally, the parameters are found to be stable across several splits of a dataset. In such cases, parameters are therefore estimated only using the first split.

4.5.2. MPII-Cooking Dataset

The dataset contains two cooking activities performed by 12 actors. The object classes and annotations are adopted from [160], which is a subset of the dataset [185]. A 7 fold cross validation is performed for evaluation as in [185]. Typically, a split contains 6,000 positive and 4,000 negative training examples and 2,000 testing examples. Regarding running times, while training the proposed method on one split took 40hrs on a 6-core 3.2GHz machine, running [35] took 72hrs on the same setup.

The AP for each object class averaged over all 7 splits are given in Table 4.1. Firstly, the proposed approach based on appearance and pose features (*Comb*), which is described in Section 4.4.4, is compared to only one of the two modalities, namely appearance (*Appr*) and pose (*Pose*), which are described in Section 4.4.2 and Section 4.4.3, respectively. Although the pose features perform worse than the appearance features, the combination results in improved performance. While a separate forest is trained for each modality, the performance of an approach where a single forest is trained on a concatenation of appearance and pose features is indicated as *Concat*. In this case both splitting functions Eqn (4.6) and Eqn (4.9) are used in a single tree. The accuracy of this approach, however, drops sharply in contrast to the appearance features.

The proposed method is also compared to the three most related approaches. In [21], Hough forests are used for object detection. While the proposed approach uses a tree model as

class	Appr.	Pose	RCNN [3]	Gall [21]	Desai [35]	Concat.	PoseObject	Comb.
bowl	0.25	0.15	0.22	0.17	0.07	0.02	0.15	0.27
bread	0.50	0.45	0.22	0.30	0.20	0.13	0.29	0.60
pan	0.20	0.20	0.57	0.34	0.22	0.14	0.21	0.23
plate	0.51	0.48	0.44	0.54	0.22	0.49	0.42	0.51
grater	0.13	0.02	0.11	0.15	0.03	0.03	0.13	0.14
squeezer	0.33	0.22	0.18	0.35	0.07	0.21	0.33	0.35
tin	0.16	0.07	0.14	0.11	0.14	0.05	0.03	0.16
spiceholder	1.00	0.15	1.00	1.00	0.60	0.92	0.15	1.00
average	0.38	0.22	0.36	0.37	0.19	0.25	0.21	0.41

Table 4.1: Average precision for the MPII-Cooking dataset.

described in Section 4.3, [21] uses a star model (using a single keypoint). When comparing it with the proposed approach using only appearance features, we observe that the multi-keypoint is only slightly better than the single-keypoint setup. Also in [3], deep neural networks are used for object detection. The performance using this approach (*RCNN*) is worse in comparison with [21] indicating the limitation of such methods to small and medium sized objects.

The method [35] combines human pose estimation and object detection. The method is trained on the training data with estimated human pose and annotated keypoints for the objects. The approach actually performs worse than the pose features. Therefore, an approach using random forests (*PoseObject*) by using appearance based features and using the joints of the human pose as additional keypoints is implemented. The results are also worse than the pose features. In order to analyze if the reduced accuracy stems from the additional pose estimation, which is not performed by the proposed approach since the estimated human poses provided by the dataset [185] are used, the impact of the chosen pose estimation method for the proposed approach is evaluated. To this end, a pose estimator [101] trained on a separate training set for pose estimation [185] was used to estimate poses on both training and test data. Using the poses estimated by the approach [101] does not change the object detection accuracy, which remains at 0.41. This indicates that it is not the pose estimation that results in a poor performance, but the combination of objects and pose as proposed in [35] is not flexible enough to model object-pose relations that are not limited to the moment of an interaction.

Additionally, the importance of estimating human pose using the same method for the training and testing data is investigated. Hence, the proposed approach is modified by retaining the poses provided by [185] for the training data but replacing poses for the testing data. When using [101] for estimating the human pose on the test data, the accuracy slightly drops from 0.41 to 0.40, showing that the approach can be trained and tested with different methods for human pose estimation. Upon using human poses obtained by [35] on the test data, the accuracy drops slightly to 0.39, but is still better than using the appearance features only.

4.5.3. ETHZ-Activity Dataset

The ETHZ-Activity dataset contains 143 sequences where 6 subjects interact with 12 different objects. Evaluation is performed through a 6 fold cross validation protocol for each of the 12 objects. Typically, a split contains 400 positive and 4,000 negative training examples and 1,500 testing examples. As a preprocessing stage, all images are normalized for lighting conditions

class	Appr.	RCNN [3]	Pose	Gall [21]	Desai [35]	Concat.	Comb.
brush	0.37	0.10	0.10	0.24	0.51	0.20	0.46
calculator	0.98	0.91	0.70	1.00	0.84	0.32	0.98
camera	0.77	0.26	0.80	0.74	0.79	0.72	0.93
headphone	0.42	0.07	0.43	0.25	0.64	0.13	0.47
marker	0.09	0.10	0.02	0.02	0.08	0.06	0.09
mug	0.25	0.25	0.13	0.30	0.54	0.05	0.30
phone	0.33	0.32	0.02	0.05	0.07	0.01	0.33
puncher	0.74	0.09	0.08	0.78	0.64	0.30	0.76
remote	0.24	0.17	0.05	0.33	0.10	0.15	0.29
roller	0.45	0.53	0.08	0.48	0.68	0.14	0.51
teapot	0.42	0.41	0.36	0.51	0.46	0.36	0.42
videogame	0.48	0.48	0.12	0.40	0.63	0.42	0.52
average	0.46	0.29	0.24	0.42	0.50	0.23	0.51

Table 4.2: Average precision for the ETHZ-Activity dataset.

using ACE [190] with parameters $a = 8$ and levels of interpolation set to 12.

The results, reported in Table 4.2, trend similarly as for the MPII-Cooking dataset. The appearance features outperform the pose features except for the classes *camera* and *headphone*. The proposed combination outperforms each of the modalities and the concatenation of both features. The method [21] performs slightly worse than the PS model with appearance features. However, *RCNN* [3] performs considerably worse. As can be seen, the method performs poorly for small classes such as *marker*, *remote* and *phone*. A closer examination revealed low recall of the object proposal stage for such objects. The approach from [35] performs better for this dataset and achieves a higher accuracy than the pose or appearance features, but the proposed combination still performs better on average.

4.5.4. CAD-120 Dataset

The CAD-120 dataset contains 120 sequences of 10 different high level activities performed by 4 subjects. Evaluation follows a 4 fold cross validation protocol for each of the 10 objects. Each split contains between 3,000 and 8,000 training examples and 4,000 testing examples. It must be noted that while most classes have a sufficient amount of training data, this is not the case for the classes $book$ and $remote$, resulting in most object detectors to fail. Also, the human pose extracted from OpenNI SDK not only has noisy joint locations specially for hands and legs, but also consists of missing joints due to low detection confidence or frequent occlusion. Missing joint locations are handled by assigning them to a default value of zero.

The results are reported in Table 4.3. The pose features perform poorly on the dataset due to low quality of estimated human poses. In particular, arms are often wrongly estimated as shown in Figure 4.3. Nevertheless, using pose features in addition to the baseline appearance features improves accuracy. The method [21] performs worse than the keypoint approach with appearance features on average. The accuracy of the approach [35] is similar to the accuracy of the concatenated features, which is lower than the proposed approach with appearance features. However, *RCNN* [3] outperforms all baselines. Significant improvements are found particularly for classes such as *cloth*, *microwave* and *medicinebox* where objects have relatively reliable and expressive appearance based features.

class	Appr.	RCNN [3]	Pose	Gall [21]	Desai [35]	Concat.	Comb.
book	0.00	0.02	0.00	0.00	0.03	0.00	0.00
bowl	0.69	0.76	0.17	0.68	0.17	0.48	0.69
box	0.60	0.74	0.10	0.55	0.03	0.27	0.60
cloth	0.03	0.66	0.00	0.02	0.12	0.00	0.03
cup	0.24	0.71	0.02	0.26	0.12	0.03	0.24
medicinebox	0.35	0.95	0.17	0.32	0.69	0.39	0.40
microwave	0.15	0.55	0.15	0.13	0.30	0.10	0.20
milk	0.75	0.09	0.30	0.71	0.61	0.69	0.75
plate	0.25	0.55	0.02	0.26	0.03	0.03	0.25
remote	0.00	0.09	0.00	0.00	0.00	0.00	0.00
average	0.31	0.63	0.09	0.29	0.21	0.20	0.32

Table 4.3: Average precision for the CAD-120 dataset.

4.6. SUMMARY

This chapter compares two models for object representations, namely star and tree model. While the tree model is found to perform slightly better, the challenges incurred during human-object interaction are addressed with an approach that combines image appearance and human pose for object detection. The approach is evaluated on three challenging datasets that contain small objects that are often occluded during human-object interaction. The experiments not only show that human pose improves an appearance based object detector irrespective of the underlying pose estimation technique, but also that the proposed combination of a separate forest for each modality outperforms the concatenation of features or a joint model for human pose estimation and object detection.

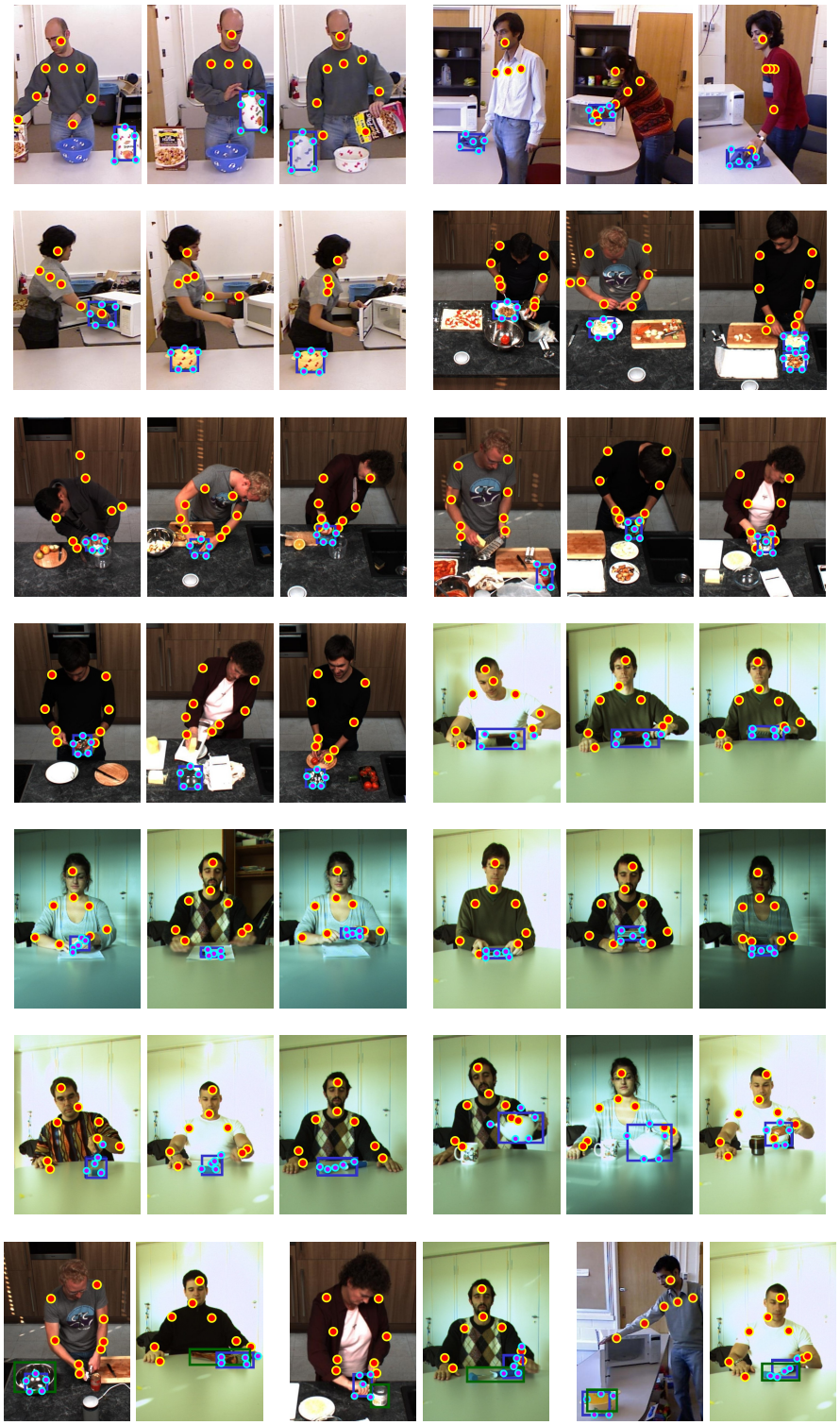


Fig. 4.3: Qualitative results showing input human pose and most confident inferred bounding boxes as per Eqn (4.11). **Top six rows:** Successful detections are shown for classes *Milk-box, Cloth, Bowl* from CAD-120; *Plate, Squeezer, Tin, Bowl* from MPII-Cooking and *Brush, Marker, Videogame, Roller, Teapot* from ETHZ-Activity. **Last row:** Failed detections due to scale problems, occlusions and faulty bounding box regression. Groundtruth bounding boxes are shown in green.

CHAPTER 5

Weakly Supervised Detection of Object Classes from Activities

Contents

5.1. INTRODUCTION

The previous chapter addressed the problem of detecting small and medium sized objects during human-object interactions. A major challenge in such scenarios is that of reduced visual evidence which drastically affects the performance of even state-of-the-art object detection approaches [3, 35]. To this end, an improved method that incorporates both low level appearance features and high level human pose features is successfully demonstrated on three datasets. However, as is innate to all data driven object detection, a vast amount of annotated training examples is crucial for good performance.

This dependence can be a bottleneck in many scenarios either because of efforts involved or inherent ambiguity involved during bounding box annotation. In the future, present-day crowdsourcing solutions will be impractical due to high associated costs and ever increasing amount of data. Moreover, this also ignores the vast amount of freely available weakly structured data. As a result, recent works in object detection have turned towards utilizing weakly

labeled data [125, 144, 145, 146, 191, 192, 193, 194], particularly videos [109, 116, 195]. Critically, these methods assume that motion or appearance of objects are sufficient descriptors for segmenting them with relative ease, which is indeed the case for large active objects such as flying airplanes and walking tigers. The assumption is further strengthened by the abundance of labeled videos on the Internet which are characteristically object- or action-centric. However, modeling daily objects such as markers, remotes or plates is still largely unresolved [45]. Exploiting weakly labeled data for such objects is further complicated by the scarcity of *clean* data because such objects do not form popular subjects for generating and sharing videoclips.

Nonetheless, labeled videos involving human activity *e.g. pouring milk* or *eating cereal* are abundantly available but violate the principal assumption of objects with dominant appearance based features. This is because prevelant themes of videos are now human body parts and background clutter instead of objects of interest. The problem is further complicated by varying appearance and pose of objects undergoing interactions coupled with low resolutions and frequent occlusions. As a result, appearance-only approaches are limited in capacity to detect such objects.

To this end, an approach which addresses the problem of weakly supervised learning for medium or small sized objects from action videos where humans interact with them is proposed. The method is composed of two stages, as shown in Fig 5.1. The first stage tackles the issue of objects of interest, which need not necessarily be dominant motion segments. Instead, we generate seeds by sampling superpixels that are likely to overlap with objects and track them to form spatio-temporal tubes as illustrated in Fig 5.2. To tackle the rich variety of object appearance and motion, tracking is made robust by sampling from a pool of algorithms and parameters. The second stage tackles the issue of appearance features alone being insufficient to describe objects. To this end, an object similarity measure is proposed that depends not only on appearance and size but also on functionality derived from relative motion with respect to the human. Further, the method facilitates extracting (possibly) numerous tubes from each video. This results in increased economy of tapping information from the data. Also, due to inherent clutter and noise, having flexibility to choose no tube from a video can potentially improve homogeneity within inferred tubes. This is realized as a greedy iterative procedure.

As in the previous chapter, the robustness of the proposed approach is demonstrated on three demanding datasets, namely one RGB dataset [185] and two RGBD datasets [154, 180]. Each dataset is recorded with a different type of sensor viz. time of flight [154], color camera [185] and structured light sensor [180]. Automatically extracted human pose in each dataset also varies in the number of detected body parts and in the quality of joint localization.

5.2. RELATED WORK

Object detection encapsulates determining whether an image contains instances of a certain object category and their locations. Optionally, additional information, *e.g.*, about part locations [26], object pose [35, 184] and occlusion [36, 37, 196, 197] has been inferred. The fundamental challenge is to effectively model inter and intra class appearance and shape variation of objects. To this day, this is usually achieved by designing a parametric model.

To this day, the parameters of the model are learned through a set of training instances using statistical machine learning techniques. The various learning methods can therefore be characterized by the extent of supervision involved during learning. At one end of the spec-

trum, fully supervised methods require careful annotation of object locations in the form of bounding boxes [21, 26, 41], segmentations [198] or even object part locations [199, 200], which is costly and can frequently introduce inconsistency and ambiguity. On the other hand, unsupervised learning methods that do not require any supervision aim at finding similar objects in a set of unlabeled images [151, 193] or videos [111]. They are, however, often limited to frequently occuring and visually consistent objects and are easily susceptible to background clutter. The stringent requirements regarding cleanliness of input data has been relaxed by using exemplar samples [201] or by employing pretrained object detectors [152, 202, 203]. On similar lines, cosegmentation [125, 191, 192, 194] approaches identify object instances up to a bounding box or segmentation on a collection of images with an object class label. Further, [204] segments objects in videos by clustering long term point trajectories. However, the method assumes similarity between trajectories from object regions and does not investigate relationships between videos.

Weakly supervised learning lies at the middle of the spectrum by providing annotations at a higher level of abstraction, thereby reducing the annotation effort. This is an important scenario for many practical applications because weak labels are more readily available, *e.g.*, in the form of text tags [205], movie transcripts [206, 207], geographical meta-data [208] or captions [209]. Weakly labeled videos are exploited in [109, 116, 160, 195, 210, 211].

In the context of object detection, the common practice has been to model object location with latent variables while jointly learning an appearance model. Most approaches impose certain assumptions for successful application, *e.g.*, [116, 144, 145, 146, 149] assume a single predominant object in the input data and [116, 195] assume rigid or articulated objects with motion distinctive from its background. These assumptions guide the latent variables such that the solution extracts object instances despite object deformations and background. In practice, however, the quality of a solution depends on the similarity measure used. For instance, [145, 146, 149] obtain a solution set that is most consistent in terms of shape and color, [116, 195] exploit motion and appearance consistency within the input data and [144] exploits symmetry constraints of objects in a multiclass framework. The solution is mostly obtained by multiple instance learning [146] or by minimizing an energy on a fully connected graph [116, 160]. Most methods fail to exploit training data completely as they only select one instance per image or video. This is a suboptimal choice because all other instances of that object in the image are ignored therefore failing to tap its true potential. This limitation is dealt within [144] by introducing a latent SVM formulation that exploits presence of multiple object instances in an image. On similar lines, the proposed method is also designed towards extracting multiple examples within the framework of exploiting human context for building models of small and medium sized objects.

The theme of scene understanding driven by human context has gained recent attention owing to advances in techniques and commercial SDKs for human pose estimation [154, 180, 212, 213, 214, 215, 216, 217, 218, 219, 220]. In [218, 220], image regions are segmented based on observed human trajectories in office and street environments. While several works [182, 221, 222] investigate combining object detection and action recognition, the works [154, 180, 217, 219] employ affordance cues as higher level representation for video understanding. In [217], both object detection and activity recognition are improved by jointly representing objects and their functionality. Unsupervised clustering of objects based on their motion relative to humans is performed in [154]. Further, human activity is recognized based

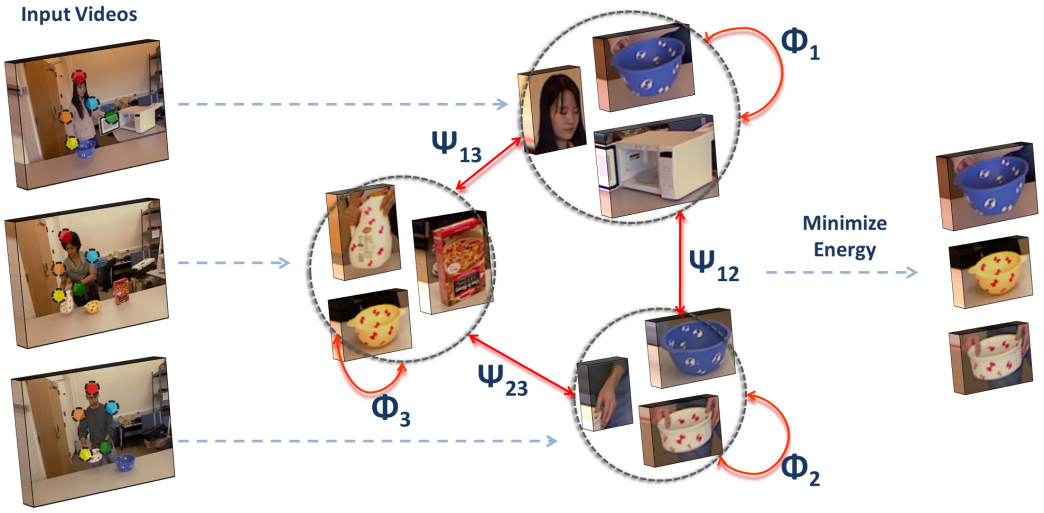


Fig. 5.1: Processing pipeline: Input is a set of action videos with human pose. Multiple sequences of object proposals (tubes) are generated from each video. By defining a model that encodes the similarity between tubes in terms of appearance and object functionality, instances of the common object class are detected.

on object functionality in the context of hand-object interactions in [219] or based on high level attribute co-occurrence statistics in [223, 224]. In [180], activities and object affordances are learned simultaneously, while [225] deals with appearance based object detection based on weak action-object labels in egocentric videos.

Human models have also been used to hallucinate their interactions with given scenes. In [214], scene locations that can afford the action *sittable* are learned through geometric relations between the scene and a human pose representing the action. A similar approach is incorporated for 3D scene labelling in [212, 216] and extracting scene geometry in [213] by modeling relations between objects and human pose. An opposite approach is followed by [215] where human poses are inferred based on scene geometry.

5.3. LEARNING OBJECT MODELS FROM ACTIVITIES

Fig 5.1 illustrates the pipeline for detecting instances of an object class in a set of RGBD or RGB videos. Input to the pipeline is a collection of videos that is labeled with the involved activity of human-object interaction. E.g., the label *cleaning microwave* indicates the presence of a microwave. It is also assumed that the 2d or 3d human pose has already been extracted. This is readily feasible because of freely available SDKs for RGBD data and due to significant progress in 2d pose estimation in the recent past. No further restrictions are imposed on the nature of input videos in that they may contain a multitude of activities, persons and/or objects. For instance, the labels *eating cereal* and *stacking bowls* are different activities that, among many other objects, commonly involve a bowl.

The first step involves generating several object proposals per video. An object proposal is modeled as a spatio-temporal region in the video, also called a tube. Multiple tubes are

sampled from a video using a simple graphical model representing human-object interactions. This procedure is explained in Section 5.3.1. While the purpose is to extract tubes that significantly overlap with the objects of interest, this is hardly true in practice as they overlap mostly with background clutter or body parts; thereby lacking object information. To this end, given a collection of tubes from all videos, a subset of tubes best describing the object from each video is selected. This is realized by minimizing an energy functional that comprises unary and pairwise potentials. Unary potentials evaluate the presence or absence of an object in a tube and pairwise potentials evaluate the similarity of objects between two tubes. All potentials incorporate appearance and functionality as described in Section 5.3.3.

5.3.1. Generating tubes

Extracting dominant motion segments as in [116, 226] is a naive way of generating tubes. Such methods cannot generate meaningful tubes in the present context as dominant motion segments mostly correspond to body parts. Instead, a tube T_v is generated from video v by tracking a frame based superpixel S over time. Owing to the rich variety of objects and actions, no unique universal setting that yielded tubes of good quality was found for either superpixel selection or tracking. Therefore, this uncertainty is modeled by randomly selecting a tracking algorithm τ from a pool of tracking algorithms. In other words, a set of tubes is obtained by sampling from the probabilistic graphical model defined over the tubes, given by

$$p(T_v, \tau, S) = p(T_v|\tau, S)p(\tau)p(S). \tag{5.1}$$

In practice, a pool of two tracking algorithms that are selected with uniform probability *i.e.* $p(\tau) = 0.5$ is used. The first method is based on propagating a superpixel (cf. Section 2.3) based on median optical flow [157] (cf. Section 2.4) into the neighboring frame. The second method is based on mean shift [227]. While the first method tracks medium sized rigid objects well, it is easily misled by fast motion, background clutter or small objects. The second method is more robust to fast motion but gets misled by occlusions during human-object interactions. Since either case fails for long term tracking, the length of each tube is limited to a maximum of 300 frames.

For generating superpixels S, the method from [156] is modified to incorporate depth as an additional feature. Since the relevance of depth information depends on material properties, object size and object characteristics, a pool of data channels is used. In practice, the pool is defined as $\sigma \in \{RGB, D, RGBD\}$. Each configuration in the pool represents the data using which superpixels are generated. The probability of selecting a superpixel also depends on frame f and a spatial prior that depends on the frame $p(l|f)$. A superpixel is obtained by sampling from

$$p(S, f, l, \sigma) = p(S|f, \sigma)p(l|f)p(f)p(\sigma). \tag{5.2}$$

A uniform prior is set over σ. $p(f)$ is a temporal prior that represents the probability of close human-object interaction in frame f. While a high level representation of humans and objects can be utilized to model this probability, a uniform distribution is employed here. In other words, it is assumed that human-object interaction occurs in all frames. As for the spatial prior $p(l|f)$, human pose information is incorporated. To this end, the joint with the highest variance in location, computed within a temporal neighborhood of 15 frames is selected. The probability $p(l|f)$ is then modeled as an isotropic uniform distribution at joint location j at

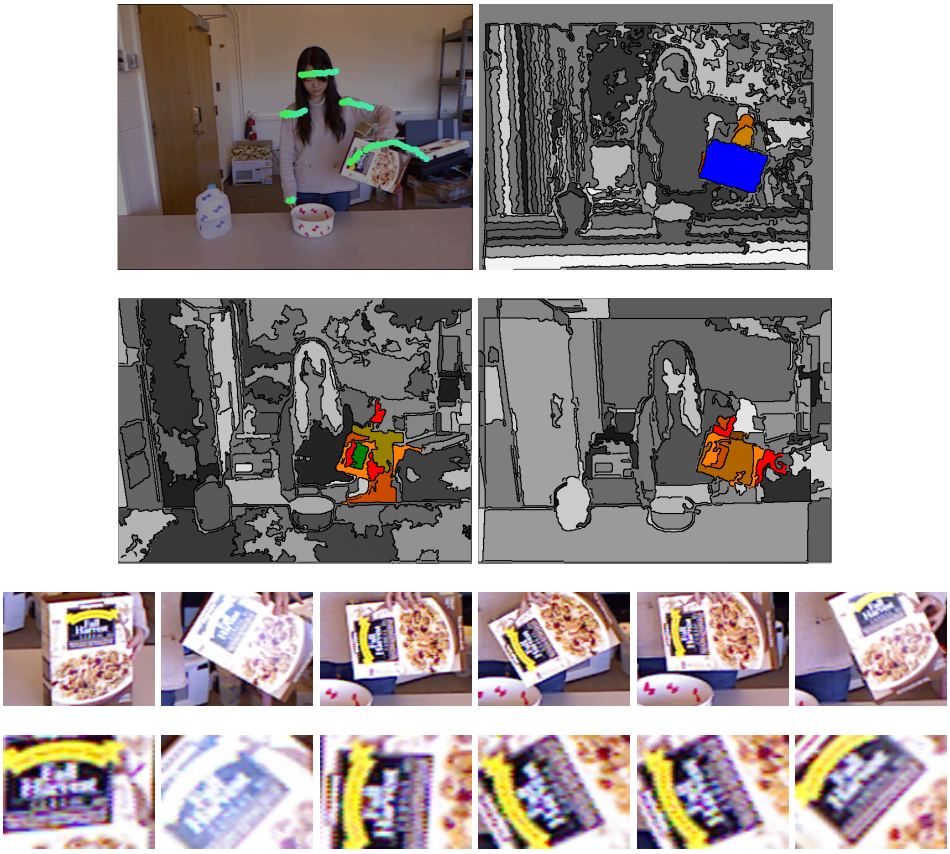


Fig. 5.2: Illustrating the tube generation process. Images of the top half: The first image shows joint trajectories. The most active joint is used to compute the spatial prior for selecting superpixels. The three images next to it show three superpixel representations computed using depth (D), color (RGB) and both (RGBD). Colored superpixels are within the specified distance of the most active joint. Each of the last two rows visualizes a tube T_v sampled from the blue and green superpixel S respectively.

frame f with radius 400mm in case of RGBD videos. Since human pose from RGB does not provide 3d information, the location of the parent joint j_p is used to compute the radius of the circle $\|\gamma(j - j_p)\|$ and its center $j + \gamma(j - j_p)$. In practice, γ is set to 0.2.

Sampling a tube from Eqn (5.1) corresponds to sampling a superpixel and a tracking method. Sampling a superpixel S from Eqn (5.2) involves sampling a configuration σ to generate a superpixel segmentation of a randomly selected frame f among which one superpixel S is chosen based on the spatial prior $p(l|f)$. This is then tracked over time using a randomly sampled tracking algorithm τ as per Eqn (5.1) to generate a tube T_v. The procedure is illustrated in Fig 5.2. As for the number of tubes generated per video, it is to 30 for all our experiments.

5.3.2. Generating object hypotheses

Given a set of candidate tubes $\mathcal{T}_v$ in each video v, the goal of [116, 160, 228] is to select one tube per video that contains the object class and is tight around the object. This has been

formulated in [116] as an energy minimization problem defined jointly over all N videos. Let $l_v \in \mathcal{L}_v = \{1, \ldots, |\mathcal{T}_v|\}$ be a label that selects one tube out of a video. The energy of all selected tubes $(l_1, \ldots, l_N)$ is defined as

$$E(l_1, \ldots, l_N) = \sum_{v=1}^{N} \left(\Phi(l_v) + \sum_{w=v+1}^{N} \Psi(l_v, l_w) \right) \tag{5.3}$$

where the unary potentials Φ measure the likelihood of a single tube being a tight fit around an object. The binary potentials Ψ measure the homogeneity in object appearance and functionality of a pair of tubes.

The constraint of selecting exactly one tube per video, however, assumes that there is at least one tube containing the object and limits the amount of information extracted from the data. In some cases, a video might contain more than one object instance or might not contain the object at all. Therefore, Eqn (5.3) is reformulated to select a varying number of tubes from each video. To this end, the objective is to find a set of tubes $S_v \subseteq \mathcal{L}_v$ for each video, which can also be an empty set. The energy of a configuration $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_N)$ is then defined as

$$E\left(\mathcal{S}\right) = \sum_{v=1}^{N} \left\{ \sum_{j=1}^{|\mathcal{S}_v|} \Phi\left(l_v^j\right) + \sum_{w=v+1}^{N} \sum_{j=1}^{|\mathcal{S}_v|} \sum_{k=1}^{|\mathcal{S}_w|} \Psi\left(l_v^j, l_w^k\right) + \alpha \left(1 - \frac{\gamma^{|\mathcal{S}_v|} e^{-\gamma}}{|\mathcal{S}_v|!} \right) \right\}. \tag{5.4}$$

The first two terms Φ and Ψ are the same as in Eqn (5.3), but they are computed over all selected tubes $\mathcal{S}_v$ for each video. The last term is a prior on the number of expected tubes with object instances per video, modeled by a Poisson distribution $P_\gamma(|\mathcal{S}_v|)$. The term $1 - P_\gamma(|\mathcal{S}_v|)$ is used since Eqn (5.4) is minimized. The parameter γ represents the expected number of object-overlapping tubes. The impact of this prior is controlled by α. Upon enforcing the constraint $|S_v| = 1$ for all videos v, minimizing Eqn (5.4) is equivalent to minimizing Eqn (5.3) since the last term reduces to a constant.

To minimize Eqn (5.4), an iterative, greedy approach is proposed. To this end, the label set is extended by an auxiliary label, *i.e.* , $\hat{\mathcal{L}}_v = \{0, 1, \ldots, |\mathcal{T}_v|\}$. Let $\mathcal{S}_v^{t-1}$ denote the selected tubes for each video at the end of iteration $t-1$. In the next iteration, either one tube or no tube, which corresponds to $\hat{l}_v^t = 0$, is selected. The already selected tubes are excluded as $\hat{l}_v^t \in \hat{\mathcal{L}}_v^t = \hat{\mathcal{L}}_v \backslash \mathcal{S}_v^{t-1}$ and the energy for iteration t is defined by

$$E(\hat{l}_1^t, \ldots, \hat{l}_N^t | \mathcal{S}^{t-1}) = \sum_{v=1}^{N} \left(\Phi(\hat{l}_v^t) + \sum_{w=v+1}^{N} \sum_{k=1}^{|\mathcal{S}_w^{t-1}|} \Psi(\hat{l}_v^t, l_w^k) + \sum_{w=v+1}^{N} \Psi(\hat{l}_v^t, \hat{l}_w^t) \right) \tag{5.5}$$

where

$$\begin{gathered} \Phi(\hat{l}_v^t{=}0) = \alpha \left(1 - \sum_{n=0}^{|\mathcal{S}_v^{t-1}|} \frac{\gamma^n e^{-\gamma}}{n!} \right) \\ \text{and} \quad \Psi(0, \hat{l}_w) = \Psi(\hat{l}_v, 0) = 0. \end{gathered} \tag{5.6}$$

In Eqn (5.5), the constant terms

$$\sum_{v=1}^{N} \sum_{j=1}^{|\mathcal{S}_v^{t-1}|} \Phi\left(l_v^j\right) \quad \text{and} \quad \sum_{v=1}^{N} \sum_{w=v+1}^{N} \sum_{j=1}^{|\mathcal{S}_v^{t-1}|} \sum_{k=1}^{|\mathcal{S}_w^{t-1}|} \Psi\left(l_v^j, l_w^k\right) \tag{5.7}$$

Algorithm 3 Greedy inference procedure

Initialize $\mathcal{S}_v^0 = \varnothing$, $\hat{\mathcal{L}}_v = \{0, 1, \ldots, |\mathcal{T}_v|\}\ \forall\ 1 \leqslant v \leqslant N$
Precompute unaries $\Phi(\hat{l}_v)$ and binaries $\Psi(\hat{l}_v, \hat{l}_w)$
Iterator $t = 0$
Continue $= True$
while Continue **do**
  $t = t + 1$
  Update set of possible labels as $\hat{\mathcal{L}}_v^t = \hat{\mathcal{L}}_v \backslash \mathcal{S}_v^{t-1}$
  Obtain $(\hat{l}_1^t, \ldots, \hat{l}_N^t)$ by minimizing Eqn (5.9)
  Update $\mathcal{S}_v^t = \mathcal{S}_v^{t-1} \cup \hat{l}_v^t$ if $\hat{l}_v^t \neq 0$
  Continue $= True$ **iff** $\hat{l}_v^t \neq 0$ for any v **else** $False$
return $\{\mathcal{S}_1^t, \ldots, \mathcal{S}_N^t\}$

are omitted. The Poisson prior $P_\gamma(|\mathcal{S}_v|)$ is expressed in the greedy approach by Eqn (5.6). In other words, the cost of selecting no tube corresponds to the probability that the video contains more than $|\mathcal{S}_v^{t-1}|$ tubes with object instances. Using $\hat{\Phi}(\hat{l}_v^t) = \Phi(\hat{l}_v^t) + \sum_w \sum_k \Psi(\hat{l}_v^t, l_w^k)$, Eqn (5.5) can be rewritten as

$$E(\hat{l}_1^t, \ldots, \hat{l}_N^t | \mathcal{S}^{t-1}) = \sum_{v=1}^{N} \left(\hat{\Phi}(\hat{l}_v^t) + \sum_{w=v+1}^{N} \Psi(\hat{l}_v^t, \hat{l}_w^t) \right). \tag{5.8}$$

Accumulating binary potentials into the unaries as in Eqn (5.8) encourage tubes selected in the present iteration to be similar to those in the past. This can cause undesirable effects as errors in the present iteration are propagated to the next. Therefore, independently optimizing each iteration can be advantageous and is formulated as

$$E(\hat{l}_1^t, \ldots, \hat{l}_N^t | \mathcal{S}^{t-1}) = \sum_{v=1}^{N} \left(\Phi(\hat{l}_v^t) + \sum_{w=v+1}^{N} \Psi(\hat{l}_v^t, \hat{l}_w^t) \right). \tag{5.9}$$

Tree-Reweighted Message Passing (cf. Section 2.2) is used for minimizing Eqn (5.8) or (5.9) and the solution set is updated for each video by $\mathcal{S}_v^t = \mathcal{S}_v^{t-1} \cup \hat{l}_v^t$ if $\hat{l}_v^t \neq 0$. The optimization procedure terminates if $\hat{l}_v^t = 0$ for all videos v. The greedy approach is described in Algorithm 3. While this does not necessarily converge to the global minimum of Eqn (5.4), it produces satisfying results as shown in the experiments.

The proposed formulation can also be used to infer instances of object classes from videos or images without human context since it can be combined with any type of unary and pairwise potentials. This chapter, however, focuses on explicit modeling of human context for the task and therefore introduces potentials that model appearance similarity as well as functionality of the object class.

5.3.3. Unary potentials Φ

Unary potential is used to measure the quality of tube l_v in video v. It is composed of four aspects, each of which aims to select tubes tightly bound to objects and interacted with. They are described as follows.

5.3.3.1. Appearance Saliency

Appearance saliency is a commonly used objectness measure since the appearance of an object generally stands out from the background. The saliency of the k^{th} frame of a tube is based on two distributions. While the first captures the RGB or RGBD distribution computed on region I_k inside the tube, the latter captures the distribution from the region S_k equal to and surrounding I_k. Saliency for frame k is then computed as the χ^2 distance between the two. Assuming tube saliency factorizes over individual frames, the potential is given by

$$\Phi^{app}(l_v) = \frac{1}{K}\sum_{k=1}^{K}\left(1 - \frac{1}{2}\sum_i \frac{(I_{k,i} - S_{k,i})^2}{I_{k,i} + S_{k,i}}\right). \tag{5.10}$$

The effect of the unary potential is that it penalizes tubes that are loosely or partially bound around objects. In either case, appearance inside and outside the tube is more similar than for a tightly bound case.

5.3.3.2. Pose-object Relation

This is a measure to evaluate if the tube is being interacted with by the human. Given the frame k, this is evaluated as the 2d or 3d Euclidean distance between the locally active end effector joint j_k of the human pose and the center c_k of the tube in that frame. To make the measure robust to pose estimation errors and interactions spanning short time duration, *e.g.*, interaction with a bowl during *eating cereal*, $\alpha = 0.3$ trimmed mean filtering is performed. Assuming that the measure factorizes over individual frames, the potential is given by

$$\Phi^{Pose}(l_v) = \frac{1}{K}\sum_{k=\alpha\cdot K}^{(1-\alpha)\cdot K} \|c_{\mathcal{D}(k)} - j_{\mathcal{D}(k)}\|, \tag{5.11}$$

where $\mathcal{D}$ is a lookup table to index over the sorted list of distances.

5.3.3.3. Body part avoidance

Body part avoidance guides the energy functional towards meaningful solutions in the weakly supervised setting. The need is highlighted in the case of body parts which are consistently present in all videos, thereby guiding the optimization to these trivial solutions. Without the aid of this term, background regions corresponding to body parts such as faces and hands, which occur in all videos, will be selected instead of objects. The appearance of the body is a mixture comprising models for skin, upper and lower bodies. The potential is then defined as

$$\begin{aligned} \Phi^{body}(l_v) &= \max\left\{\bar{p}_{skin}(I), \bar{p}_{upper}(I), \bar{p}_{lower}(I)\right\}, \\ \text{with} \quad \bar{p}_x(I) &= \frac{1}{K}\sum_k p_x(I_k), \end{aligned} \tag{5.12}$$

where I_k is the color histogram of the tube at frame k. 5-component Gaussian mixture models (GMM) are used for both upper and lower bodies, learned directly using pixels around relevant joints of the estimated pose. As for skin color, the generic model from [229] is used.

5.3.3.4. Size prior

A prior on the size of an object is an important cue that can be inferred relative to the human size in human-object interaction scenarios. In other words, there are bounds on the physical size of an object a human can interact with. *E.g.*, interactions with phone, plate and markers are possible, but not with the floor or the cap of a marker. Such image level priors can be helpful when tubes are very small, rendering other potentials unreliable. The prior on the object size is modeled as a Gaussian distribution given as

$$\Phi^{size}(l_v) = \exp\left(\frac{(w_{l_v} - 2w_h)^2 + (h_{l_v} - 2h_h)^2}{2\sigma_h^2}\right), \tag{5.13}$$

where (w_h, h_h) and (w_{l_v}, h_{l_v}) are average width and height of the hand and tube respectively and σ_h is 1.5 times the average size of the hand.

5.3.3.5. Unary potential

The final unary potential is formed by linearly combining the four terms as

$$\Phi(l_v) = \lambda_1 \Phi^{app}(l_v) + \lambda_2 \Phi^{pose}(l_v) + \lambda_3 \Phi^{body}(l_v) + \lambda_4 \Phi^{size}(l_v), \tag{5.14}$$

where the weighting parameters λ_i are learned from a held-out validation set as explained in Section 5.4.

5.3.4. Pairwise potentials Ψ

The pairwise potential measures the similarity between two tubes l_v and l_w and is composed of two terms. The first term measures the inter-tube appearance similarity and the second term measures the similarity of their motion during interaction.

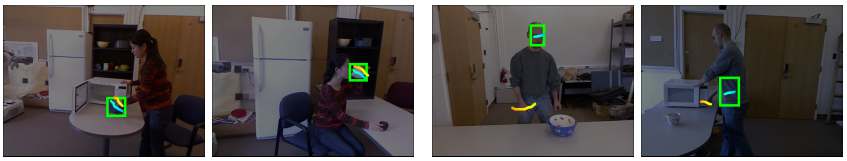
5.3.4.1. Shape

Following [116], the appearance similarity between two tubes is based on Pyramid-HOG (PHOG) [230]. The appearance of a tube is described by a multiresolution histogram of gradients computed over 50 uniformly sampled frames in the tube. Further, the two sequences are first aligned using dynamic time warping to account for varying object appearance during interaction. The warping is performed using the joint locations of the head, shoulders and hands as features. Since the alignment between two distinct action sequences is meaningless, original tubes are retained if the alignment error exceeds a certain threshold. The pairwise potential $\Psi^{shape}(l_v, l_w)$ is defined as the median χ^2 distance between PHOG features from corresponding frames k of tubes l_v and l_w is given as

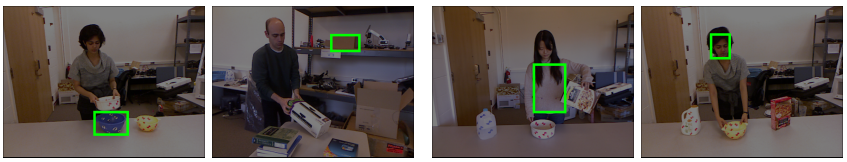
$$\Psi^{shape}(l_v, l_w) = \underset{k}{\text{median}} \left\{ \frac{1}{2} \sum_i \frac{\left(P_{\omega_v(k),i} - P_{\omega_w(k),i}\right)^2}{P_{\omega_v(k),i} + P_{\omega_w(k),i}} \right\}, \tag{5.15}$$

where ω_u is the dynamic time warping (DTW) function for tube l_u and $P_{\omega_u(k),i}$ is i^{th} bin of the PHOG feature extracted from the k^{th} frame of tube l_u after warping.

(a) Unary Potential (See caption for details): Appearance Saliency



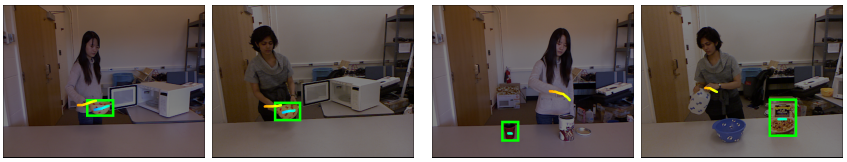
(b) Unary Potential: Pose-object Relation



(c) Unary Potential: Body part avoidance

(d) Unary Potential: Size prior

(e) Pairwise Potential: Shape



(f) Pairwise Potential: Functionality

Figure 5.3: Illustrating the unary and pairwise potentials. Bounding boxes in the first two columns favour the energy in Eqn (5.4) by decreasing it in comparison with those in the last two columns. (a)–(d) correspond to unary potentials and illustrate two distinct favorable cases and unfavorable cases each. (e)–(f) correspond to pairwise potentials and illustrate a single favorable case and unfavorable case. Temporal paths of the most active joint location and the bounding box are marked in yellow and cyan respectively.

5.3.4.2. Functionality

Assuming relative trajectories of objects with respect to the human correlated with object functionality, the relative Euclidean distance between the center of the tube and the human is measured. After having preprocessed the tubes as for the shape potential, 50 pairs of corresponding frames are sampled uniformly. Given frame k, the distance between the center $c_{u(k)}$ of the tube l_u and the head position $h_{u(k)}$ is computed and normalized by the distance between the head and the locally active end effector $j_{u(k)}$:

$$d_{u(k)} = \frac{\|h_{u(k)} - c_{u(k)}\|}{\|h_{u(k)} - j_{u(k)}\|}. \tag{5.16}$$

While the normalization accounts for lack of 3d information in 2d human poses, it also compensates for varying body sizes in 3d human poses. Given the DTW functions ω_*, the potential $\Psi^{func}(l_v, l_w)$ is then the median of these differences:

$$\Psi^{func}(l_v, l_w) = \underset{k}{\text{median}} \left\{ |d_{\omega_v(k)} - d_{\omega_w(k)}| \right\}. \tag{5.17}$$

Pairwise potential The final pairwise potential is formed by the linear combination given by

$$\Psi\left(l_v, l_w\right) = \lambda_5 \Psi^{shape}\left(l_v, l_w\right) + \lambda_6 \Psi^{func}\left(l_v, l_w\right), \tag{5.18}$$

where weighting parameters λ_i are learned together with the weights of the unary potential (5.14) from a validation set.

5.4. EXPERIMENTS

The proposed method is evaluated on two RGBD datasets and one RGB dataset[1], which represent a rich variety of modalities: ETHZ-Action [231], CAD-120 [180] and MPII-Cooking [185]. The ETHZ-Action is an RGBD dataset composed of a time-of-flight and a color camera with a resolution of 170×144 and 640×480, respectively. The dataset contains 6 different actors, each performing high level activities with 12 objects totaling to 143 video sequences. An 8-joint upper body 3d human pose is extracted using a model based method. While interactions are mostly restricted to a single object, there is significant intra-class variation in object appearance due to the interaction. The 12 objects range from being medium sized, *e.g.*, *brush* and *teapot* to small sized, *e.g.*, *marker* and *videogame*. A typical frame illustrating the relative size of an object is shown in Fig 5.4.

CAD-120 is an RGBD dataset captured using the Kinect sensor. Therefore, both color and depth images have a resolution of 640×480. The dataset contains 4 actors performing 10 different high level activities totaling to 120 video sequences. The OpenNI SDK is used to extract human pose consisting of 15 3d whole body joint locations with binary confidence flags for each joint. Noise in the pose is more pronounced for limb joints, *i.e.*, hands and legs. Some activities involve multiple instances of the same object, *e.g.*, *stacking objects* or multiple objects, *e.g.*, *taking medicine* that indicates presence of *medicinebox* and *cup*. It must be noted that the classes *book* and *remote* appear in only three video sequences each.

[1] Annotations for all three datasets can be found at `http://ps.is.tue.mpg.de/person/srikantha`

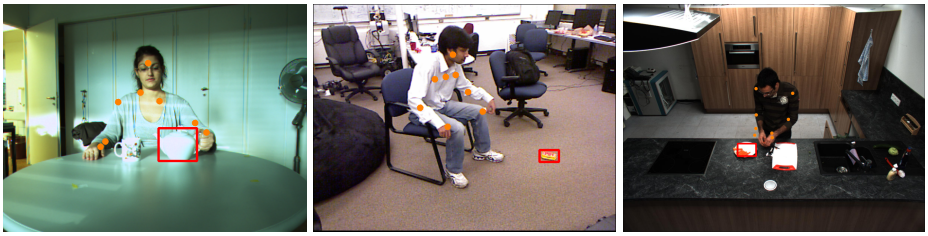


Fig. 5.4: Illustrating human-object interaction from ETHZ-Action dataset, CAD-120 dataset and MPII cooking dataset with human pose overlayed in orange and objects with a red bounding box.

The MPII-Cooking is a high resolution (1624×1224) RGB dataset. It contains 2 high level activities performed by 12 different actors totaling to 65 video sequences. The extracted human pose consists of 8 2d joint locations for the arms. Therefore, in the pairwise potential $\Psi^{func}(l_v, l_w)$ in Eqn (5.17), the location of the head is replaced by the mean location of both shoulders. This is a challenging dataset where objects evolve in appearance and frequently undergo occlusions. *E.g.*, *bread* evolves from being a layer of dough to an arrangement of vegetables during the preparation a pizza.

Further, comparison with a weakly supervised approach [116] and an unsupervised approach [231] is provided. The method in [231] discovers objects by clustering trajectories of human joint locations. The method in [116] uses motion segments to generate object proposals, which are then fed into an energy functional similar to Eqn (5.3). The unary and pairwise potentials are inspired purely by appearance features. While unary potentials are composed of objectness [73], intra-tube shape consistency and bounding-box heuristics, pairwise potentials are based on inter-tube shape consistency. Their solution involves extracting one tube per video which best represents the latent object.

5.4.1. Inference

The output of the system is a collection of tubes that best describe an object class common to all input videos. Detected instances of object classes are shown in Fig 5.8. In order to evaluate the quality of these tubes, frame- and class-wise PASCAL IOU measures are presented. A frame-IOU measure is defined as a ratio of areas of intersection over union of the ground truth and inferred bounding boxes. A tube-IOU is defined as the average of all frame-IOUs. Similarly, a class-IOU is defined as the average of all inferred tube-IOUs.

Validation dataset comprises ground truth annotations of one randomly chosen object class per dataset: *puncher* (ETHZ-Action), *milkbox* (CAD-120) and *whisker* (MPII-Cooking). The configuration of model parameters $\lambda \in \{0.05, 0.25, 0.50, 0.75, 1.00\}$, α and γ is set via grid-search in with an objective to maximize class-IOU for the validation class. These are therefore excluded from all performance evaluations that follow.

5.4.2. Comparison

In the context of detecting objects from videos with activities, the experiments show that naive motion based segmentation as in [116] and object proposal method [232] fail at varying levels

	[116]	modif-[116]	Eqn (5.3)	Eqn (5.8)	Eqn (5.9)
ETHZ-Action	0.063	0.249	0.447	0.439	0.471
CAD-120	0.039	0.246	0.410	0.393	0.423
MPII-Cooking	0.023	0.221	0.342	0.333	0.348

Table 5.1: Average class-IOU of the proposed model for the three datasets. The Eqn (5.9) which infers multiple tubes per video outperforms Eqn (5.3) which extracts a single tube per video and [116] which relies on motion segments and object appearance and ignores object functionality.

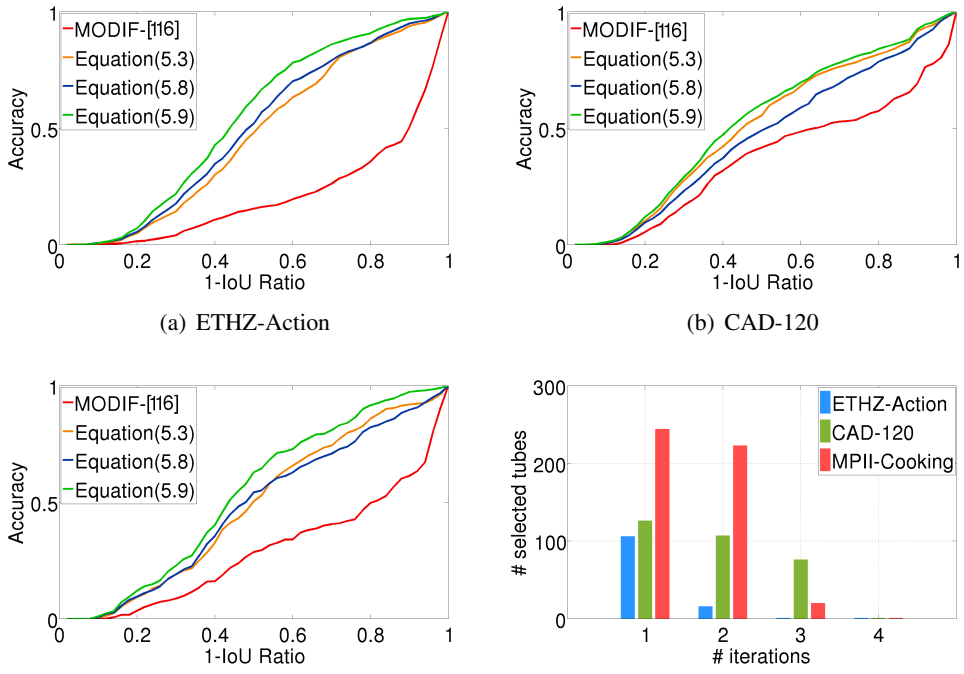
of severity. Improved performance is shown in Table 5.1 due to improved object proposals as generated by Section 5.3.1 and the inclusion of object functionality in Eqn (5.3). Further extending Eqn (5.3) to select a varying number of tubes from each video as per Eqn (5.8) and (5.9) improves the quality of inferred tubes and the subsequent object detection performance. It is found that the framework presented in Eqn (5.8) is prone to noise, thereby often yielding suboptimal solutions and the independence assumption incorporated in Eqn (5.9) helps alleviate this limitation. Further experimental evaluation is presented below.

Firstly, an object proposal technique [232] is compared against the proposed tube generation process. For this experiment, every 10^{th} frame in the ETHZ-Action dataset is considered. The recall of [232] for (10^2, 10^3, 10^4) proposals per image is (0.19, 0.58, 0.67), respectively, against 0.65 for 30 tube proposals as generated in Section 5.3.1.

Regarding overall accuracy, a method for learning from weakly labeled videos [116] is compared with an approach that optimizes Eqn (5.3). The average class-IOU for all three datasets is presented in Table 5.1. Optimizing Eqn (5.3) outperforms [116] significantly. The poor performance of [116] is due to the inferior quality of object proposals which are extracted based on dominant motion segments, which overlap mostly with human body parts instead of objects. The method is therefore modified by replacing object proposals with those generated from Section 5.3.1, but retaining the energy functional proposed in [116]. The modified approach is denoted as modif-[116] in Table 5.1. While modif-[116] performs significantly better than its baseline [116], it still underperforms when compared to Eqn (5.3).

Equations (5.8), (5.9) extend Eqn (5.3) by inferring multiple tubes. While the former lags behind the baseline Eqn (5.3) on all three datasets, the latter performs favorably in ETHZ-Action and CAD-120 datasets and comparably in the MPII-Cooking dataset. To reason about the superior performance of Eqn (5.9) against that of Eqn (5.8), the energy obtained by both methods as per Eqn (5.4) is compared. It is found that energies pertaining to Eqn (5.9) are lower in 9 out of 12 classes in ETHZ-Action and 6 out of 9 classes in CAD-120 dataset. A possible reasoning for this is that assuming independence between iterations in Eqn (5.9) can better handle noise without propagating it into further iterations.

In order to further evaluate the quality of inferred tubes, class-accuracy is defined as the fraction of bounding boxes with an IOU ratio greater than a given threshold. Fig 5.5 shows class-accuracy averaged over all classes for decreasing IOU ratios. Because of the inferior performance of [116], accuracy is reported for modif-[116]. As can be seen, modif-[116] underperforms in all three datasets verifying the suboptimality of related potentials. The in-



(a) ETHZ-Action (b) CAD-120

(c) MPII-Cooking (d) Algorithm (3)

Fig. 5.5: (a–c) The accuracy is measured as the fraction of bounding boxes with IOU ratio greater than a given threshold. The x-axis plots 1-IOU *i.e.* the higher the value on the x-axis, the more tolerant is the success threshold and the higher the accuracy. The accuracy presented is averaged over all classes. (d) Number of selected tubes inferred in each iteration. After the third iteration, the approach has converged because no new tubes are added to the set of selected tubes $\mathcal{S}_v^t$.

troduction of new potentials as in Eqn (5.3) shows improvements, the biggest of which is for the ETHZ-Action dataset at 1-IOU=0.8 where the former performs at 0.36 and the latter at 0.86. Although introducing multi-tube inference as in Eqn (5.8) results in reduced performance, the independence assumption in Eqn (5.9) is favorable on all three datasets. Significant improvements are found in ETHZ-Action and MPII-Cooking at 1-IOU=0.5 with around 10% increase in accuracy from the performance of Eqn (5.3). At IOU=0.5, the accuracy of Equations (5.9), (5.3) and modif-[116] are (0.62, 0.48, 0.16) for ETHZ-Action, (0.60, 0.56, 0.42) for CAD-120 and (0.63, 0.53, 0.29) for the MPII-Cooking dataset, respectively.

The number of tubes selected in each iteration for the datasets is shown in Fig 5.5(d). For the ETHZ-Action dataset, all tubes are selected after two iterations. For the other two datasets, the approach converges after three iterations. Using the multiple instance inference of Eqn (5.9), a total of 124, 310 and 488 tubes are selected for the ETHZ-Action, CAD-120 and MPII-Cooking dataset, respectively. As a comparison, single instance inference of Eqn (5.3) selects only 106, 126 and 244 tubes for the datasets.

Regarding running times, the CPU only implementation takes about 1 hour to extract 30 tubes per video and about 5 hours to precompute unary and pairwise potentials. The inference procedure is fast and takes about 15 seconds for a collection of 20 videos.

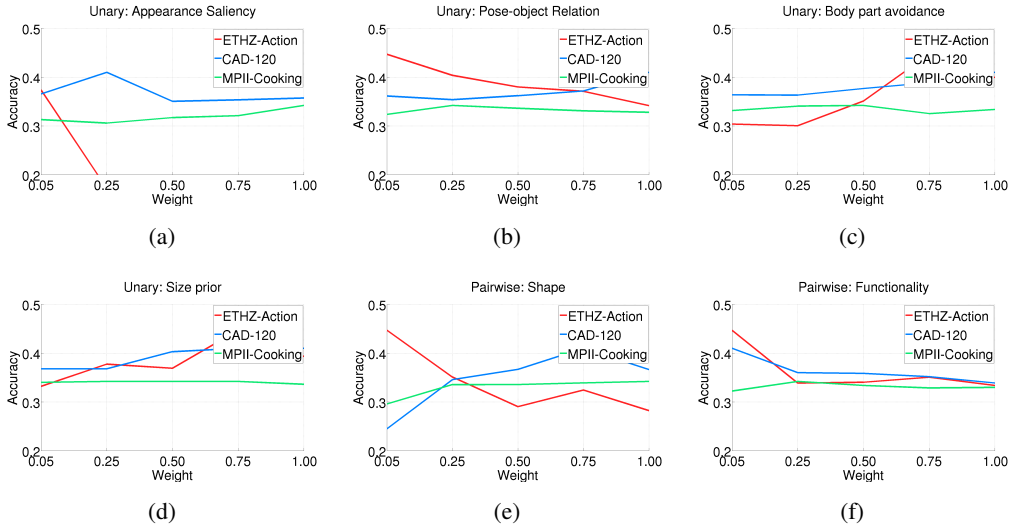


Fig. 5.6: Sensitivity of parameters for Equations (5.14) and (5.18). Accuracy is measured by average class-IOU.

5.4.3. Evaluating parameter sensitivity

The parameters of Equations (5.14) and (5.18) are estimated on a validation set as described in Section 5.4.1. In order to show the sensitivity of these parameters, each of the learned weights is varied and the resulting average class-IOU is shown in Fig 5.6(a)–(f). As can be seen, while varying almost any potential has minimal effect on MPII-Cooking, the effects are more drastic for ETHZ-Action. The performance on CAD-120 is sensitive to variations in Φ^{body} and Ψ^{shape}.

5.4.4. Impact of Potentials

The potentials are grouped into three categories to study the nature of contributions from the designed potentials. They are: APP consisting of potentials that are inherent to object appearance $\{\Phi^{app}, \Psi^{shape}\}$, SIZ denotes the size prior $\{\Phi^{size}\}$ and FUN consisting of potentials derived from human-object interaction $\{\Phi^{pose}, \Phi^{body}, \Psi^{func}\}$. Table 5.2 presents the performance of Eqn (5.3) under various group combinations.

The foremost observation is that the group APP underperforms in comparison with modif-[116] for all datasets. This is an expected fall in performance due to the difference in the representation of appearance information by both methods. The performance improves upon adding the size prior (APP+SIZ). The importance of incorporating human-object interaction is seen when the functionality terms (FUN) outperform modif-[116] and APP on both ETHZ-Action and MPII-Cooking datasets. Further, combination of (FUN+APP) outperforms individual settings indicating that both groups encode complementary information. Finally, the pair of (FUN+SIZ) performs best among all proper subset combinations attaining more than 80% of the maximum recorded performance. This indicates that all potential groups are important for achieving maximum performance.

Additionally, the effect of discarding a single potential from the model in Eqn (5.9) is

Eqn (5.3)	modif-[116]	Eqn (5.3)	APP	APP+SIZ	FUN	APP+FUN	FUN+SIZ
ETHZ-Action	0.249	0.447	0.192	0.305	0.292	0.312	0.390
CAD-120	0.246	0.410	0.168	0.191	0.147	0.202	0.350
MPII-Cooking	0.221	0.342	0.079	0.149	0.229	0.235	0.288

Table 5.2: Studying the contribution of various potential groups in Eqn (5.3). Average class-IOU is presented for (APP+SIZ+FUN) for the three datasets. All three types of potentials that model object appearance (APP), size prior (SIZ) and object functionality (FUN) are important for the final performance.

Eqn (5.9)	Φ^{app}	Φ^{pose}	Φ^{body}	Φ^{size}	Ψ^{shape}	Ψ^{func}
ETHZ-Action	-3.27	-11.40	-6.09	-13.17	-2.43	-3.00
CAD-120	-9.48	-0.85	-6.38	-9.19	-10.06	-11.71
MPII-Cooking	-10.33	-7.47	-7.47	-3.54	-9.79	-34.00

Table 5.3: Percentage change in average class-IOU performance when any given potential is discarded from Eqn (5.9).

Eqn (5.3)	Φ^{app}	Φ^{pose}	Φ^{body}	Φ^{size}	Ψ^{shape}	Ψ^{func}
ETHZ-Action	0.35	1.88	-25.49	-13.50	-4.62	-8.86
CAD-120	-48.66	-15.73	-18.89	-20.80	-40.15	-9.19
MPII-Cooking	-15.85	0.06	-31.09	-10.70	0.058	-60.95

Table 5.4: Percentage change in average class-IOU performance when any given potential is discarded from Eqn (5.3).

presented in Table 5.3. It can be observed that eliminating any potential causes a drop in performance. Appearance based features have minimal impact on the ETHZ-Action dataset as they are not reliable for small objects. Discarding Ψ^{func} most adversely affects the MPII-Cooking dataset due to closer interaction between human and objects in comparison with the other two datasets. On the other hand, discarding Φ^{pose} has the least impact on the CAD-120 dataset. This is because the inferred human pose is noisy due to missing joints and poor localization accuracy. In fact a qualitative evaluation confirmed that the pose quality for CAD-120 is the lowest among the three datasets. Φ^{body} and Φ^{size} reduce the performance for all three datasets. Due to the small size of the objects in ETHZ-Action, Φ^{size} has the biggest impact on this dataset. Study on Eqn (5.3) showed similar results and is presented in Table 5.4.

Further, the robustness of pose-related potentials is studied with respect to strong pose estimation noise on the CAD-120 dataset. To this end, normally distributed noise with variance $100cm^2$, $200cm^2$ and $400cm^2$ is added to each 3d joint position. The average class-IOU then drops to 0.365, 0.342 and 0.323 respectively from the baseline of 0.423 (see Table 5.1). The performance, however, is still higher than without using these potentials (see APP+SIZ in Table 5.2).

Class	GTr.	Infer	Eqn 4.11	Class	Gtr.	Infer	Eqn 4.11
ETHZ-Action							
brush	45.1	38.0	37.0	calcul.	100.0	100.0	80.0
camera	83.5	73.0	73.0	remote	49.4	36.7	32.2
mug	38.0	30.2	31.4	headph.	69.8	63.7	36.1
marker	39.7	39.7	05.1	teapot	63.2	59.2	60.3
videog.	78.3	77.6	46.1	roller	99.6	66.1	46.3
phone	0.05	11.9	05.5	**Avg.**	60.6	54.2	41.2
CAD-120							
book	11.2	03.2	03.2	medbox.	58.3	53.3	38.1
bowl	24.5	24.5	24.5	mwave.	71.4	71.0	30.0
box	24.4	21.5	21.0	plate	16.2	14.3	11.4
cup	14.8	12.9	13.7	remote	14.1	08.3	08.3
cloth	20.1	18.6	05.6	**Avg.**	29.4	25.3	17.3
MPII-Cooking							
bowl	69.2	64.4	41.0	spiceh.	100.0	100.0	63.2
bread	25.5	13.2	13.2	squeez.	61.5	61.5	61.5
plate	43.4	43.4	55.0	tin	33.0	26.4	32.6
grater	02.2	01.2	01.2	**Avg.**	47.8	44.3	34.2

Table 5.5: Average precision (%) for different datasets comparing object models built from ground truth data (GTr.) and inferred data (Infer) from Eqn (5.9).

5.4.5. Evaluating object models

The quality of inferred tubes from Eqn (5.9) are now evaluated in terms of object detection performance. Training and testing data are obtained by defining splits on each dataset such that they share no common actors. For training, data from 3 out of 4 actors in CAD-120, 5 out of 6 actors in ETHZ-Action and 9 out of 12 actors in the MPII-Cooking dataset are considered. The rest of the data *i.e. Subject-1* for CAD-120, *actor-14* for ETHZ-Action and {*s18,s19,s20*} for MPII-Cooking is used for testing.

For object detection, a Hough forest (cf. Section 2.1) with 5 trees is used. Each tree is trained until a maximal depth of 25 and with 50,000 positive and 50,000 negative patches (drawn uniformly from the background). Depth data is not used for this experiment. The fully supervised baseline, denoted as 'GTr.' in Table 5.5, is based on manually annotated bounding boxes of training images, *i.e.*, every 10^{th} frame of the training sequences. The 'Infer' training data is based on an equal number of frames from the automatically extracted tubes by Eqn (5.9). Further, 'Eqn 4.11' incorporates human pose as context as described in Section 4.4.4.

The results show that optimal performance is achieved for categories like *calculator*, *marker* in ETHZ-Action, *bowl*, *microwave* in CAD-120 and *spiceholder*, *squeezer* in MPII-Cooking. On the other hand, a loss in performance is observed for many categories due to weaker supervision. This is due to noisier extracted tubes in comparison with manually annotated data. Nevertheless, performances of the object detectors trained on weakly supervised videos achieve 89.4% (ETHZ-Action), 86.1% (CAD-120) and 92.6% (MPII-Cooking) of that from full supervision. Further, incorporating human pose as per Eqn 4.11 generally results

in worse performance because the training data is biased towards humans closely interacting with objects unlike in the testing data, and yet, improvements are seen in classes like *plate* and *tin* where the degree of interaction is more consistent.

A comparison of the object detection performance when training data is obtained from Equations (5.3), (5.8), (5.9) is shown in Fig 5.7. It can be observed that object detectors based on Eqn (5.9) generally outperform those from Equations (5.3) and (5.8). Particularly, the average precision is improved when compared to Eqn (5.3) in all three datasets from 53.2% to 54.2% for ETHZ-Action, 24.4% to 25.3% for CAD-120 and 35.3% to 44.3% in the MPII-Cooking dataset. However, there is a small loss in performance for a few classes such as *camera, headphone* in ETHZ-Action and *book, remote* in the CAD-120 dataset.

Also, a comparison with [231] which is an unsupervised approach that segments and clusters videos based on pose features is presented. [231] generates 20 clusters for the ETHZ-Action dataset without labels and only 3–21 object samples per cluster, while our approach generates more than 300 samples per class. Although the resulting clusters cannot be directly compared, they are labeled manually to train object detectors for all 12 classes. The average precision on ETHZ-Action is 24.85% in comparison to 54.20% of the proposed approach.

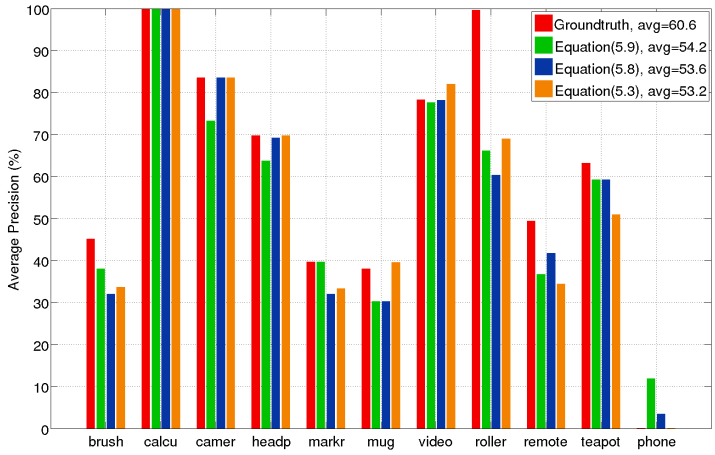
5.4.6. Refining objectness using object detectors

Approaches like [136, 233] propose a weakly supervised method where a detector is initialized using a few seed examples and later refined by incorporating new detections. To evaluate whether iterating between training the detector and inferring training data from videos improves accuracy, the object detector (Section 5.4.5) is applied to the tubes and the detector confidence is used as a fifth unary potential in Eqn (5.14). The process is iterated until there is no change in the set of selected tubes.

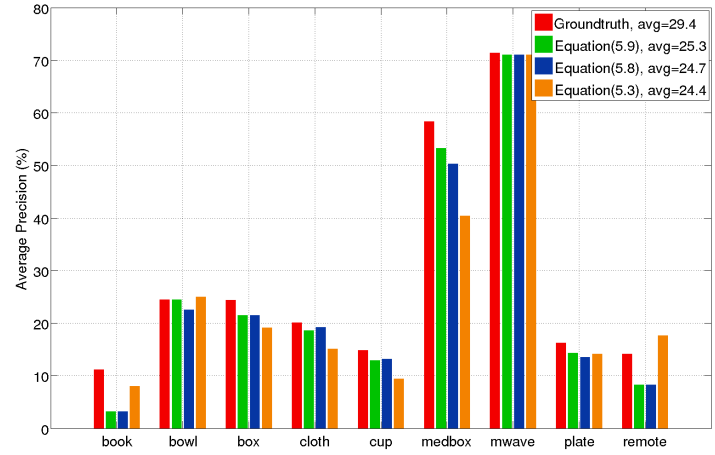
Repeating the experiments as described in Sections 5.4.1–5.4.2 with the augmented model, the procedure for ETHZ-Action and CAD-120 terminated after the first iteration without any improvement in average class-IOU measure. However, the procedure for MPII-Cooking terminated after two iterations, yielding a marginal improvement from 0.342 to 0.343 (cf. Table 5.2). The object detection performance remained unchanged for all datasets.

5.5. SUMMARY

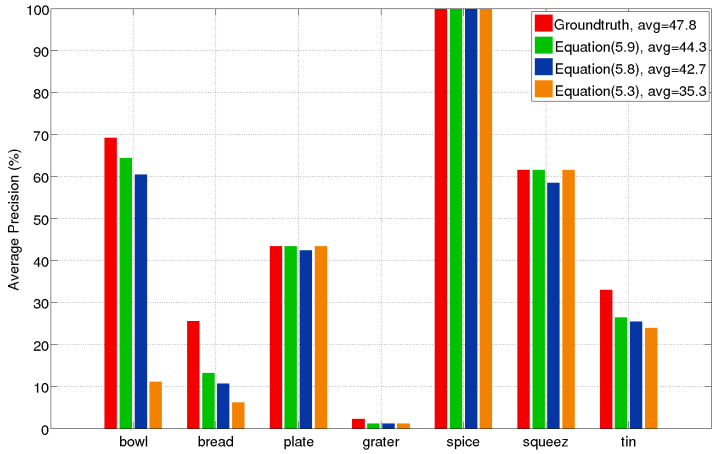
This chapter addresses the problem of detecting instances of small and medium sized objects from weakly labeled activity videos. The experiments show that approaches relying entirely on object motion or appearance fail for this task. Although using only object appearance is shown to be insufficient, coupling it with object functionality leads to greatly improved performance. An interesting aspect is that the results reveal the complementary nature of functionality and appearance related potentials for detecting objects. In order to maximize utilization of data, a framework for inferring multiple object instances from each video is proposed which is solved using a greedy approach. The superior quality of these tubes are verified by the experiments. The generalization capabilities are demonstrated on three datasets that span a variety of different activities, modalities (RGB vs. RGBD), and pose representations (2d vs. 3d). Finally, the weakly supervised approach outperforms an unsupervised approach and achieves between 86% and 92% of the performance of a fully supervised approach for object detection.



(a)



(b)



(c)

Fig. 5.7: Average precision (%) for object detection on different datasets given training data from ground truth and from Equations (5.9), (5.8) and (5.3).

Fig. 5.8: Detected instances of the object classes as in Eqn (5.3): *Marker, Mug, Camera, Roller, Milkbox, Bowl, Cloth, Microwave, Plate, Tin, Bread, Squeezer* and Failure cases *Teapot, Brush*. The first image in each row shows relative object size by illustrating a typical action scene with overlayed human pose and a bounding box around the object of interest. Since the objects are relatively small, images are best viewed by zooming in.

CHAPTER 6

Weakly Supervised Segmentation of Object Affordances

Contents

6.1. INTRODUCTION

The previous chapters dealt with detecting objects up to the resolution of bounding boxes. While the appearance features alone are unreliable for small and medium sized objects, human context was successfully incorporated in object-human interaction scenarios to improve object detection. Further, improvements in weakly supervised object detection was demonstrated by integrating complementary cues based on human centric object functionality. Such localizing about objects through bounding boxes might be sufficient for numerous applications. However, more detailed information about object (part) affordances/functionalities might be useful for other collaborative applications.

Object centric functionality can be categorized into abstract descriptive properties called *attributes* [234, 235, 236] or physically grounded regions called *affordances*. Affordances are important as they form the key representation to describe potential interactions. For instance, autonomous navigation depends heavily on understanding outdoor semantics to decide if the lane is *changable* or if the way ahead is *drivable* [237]. Similarly, assistive robots must have the capability of anticipating indoor semantics like which regions of the kitchen are *openable* or *placeable* [180]. Further, because forms of interaction are predetermined for virtually any

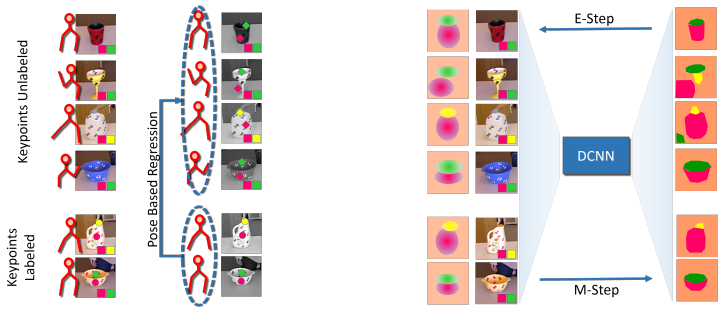


Fig. 6.1: Overview of the weakly supervised approach for affordance labeling. (Left) Human pose information available for all images is used to regress weak labels (clickpoints) in the image subset without such annotation. (Section 6.4.4). (Right) The estimated clickpoints are used to initialize the EM framework (Section 6.4.3). The E-step computes a point estimate of the latent segmentation based on Gaussian distributions. The M-step learns the parameters of the DCNN considering the point estimate as ground truth segmentation (Section 6.4.2).

object class, it is desirable to have recognition systems that are capable of localizing functionally meaningful regions or *affordances* alongside contemporary object recognition systems.

In most previous works, affordance labeling has been addressed as a stand-alone task. For instance, the methods [131, 238, 239, 240] learn pixel-wise affordance labels using supervised learning techniques. Creating pixelwise annotated datasets, however, is heavily labor intensive. Therefore, in order to simplify the annotation process, current affordance datasets have been captured in highly controlled environments like a turntable setting [131]. However, this does not allow to study contextual information, specially those of humans, which affordances are intrinsically related to. One of the contributions of this chapter is to propose a pixel annotated affordance dataset within the purview of human interactions, thus creating possibilities to tap rich contextual information thereby fostering work towards reduced supervision levels. In addition, it is shown that state-of-the-art end-to-end learning techniques in semantic segmentation significantly outperform state-of-the-art supervised learning methods for affordances.

The chapter also proposes a weakly supervised learning approach for affordance segmentation. The approach is based on the expectation-maximization (EM) framework as proposed in [241] where a constant bias term is used to learn a deep convolutional neural network (DCNN) for semantic segmentation only from image level labels. In the proposed method, clickpoints are considered as weak annotations as they are easy to obtain and have been used in [242] for annotating a large material database. In order to learn from clickpoints, the framework is extended to handle spatial dependencies. The approach can also be used to learn from mixed sets of training images where one set is annotated by clickpoints and the other is annotated by image labels. An overview of the proposed EM approach is illustrated in Figure 6.1.

Further, it is shown that automatically extracted human pose information can be effectively utilized as context for affordances. It is used to transfer clickpoint annotations to images without clickpoint annotations, which are then used to initialize the proposed EM approach.

6.2. RELATED WORK

Properties of objects can be described at various levels of abstraction by a variety of attributes including visual properties [234, 243, 244, 245] *e.g.* object color, shape and object parts, physical properties [246, 247], *e.g.* weight, size and material characteristics and categorical properties [248, 249]. Object affordances, which describe potential uses of an object, can also be considered as other attributes. For instance, [250] describes affordances by object-action pairs whose plausibility is determined either by mining word co-occurances in textual data or by measuring visual consistency in images returned by an image search. [247] proposes to represent objects in a densely connected graph structure. While a node represents one of the various visual, categorical, physical or functional aspects of the object, an edge indicates the plausibility of both node entities to occur jointly. Upon querying the graph with observed information, *e.g. {round, red}*, the result is a set of most likely nodes, *e.g. {tomato, edible, 10-100gm, pizza}*.

Affordances have also been used as an intermediate representation for higher level tasks. In [251], object functionality is defined in terms of the pose of relevant hand-grasp during interactions. Object recognition is performed by combining individual classifiers based on object appearance and hand pose. [252] uses affordances as a part of a task oriented object modeling. They formulate a generative framework that encapsulates the underlying physics, functions and causality of objects being used as tools. Their representation combines extrinsic factors that include human pose sequences and physical forces such as velocity and pressure and intrinsic factors that represent object part affordances. [253] models action segments using CRFs which are described by human pose, object affordance and their appearances. Using a particle filter framework, future actions are anticipated by sampling from a pool of possible CRFs thereby performing a temporal segmentation of action labels and object affordances. [217] jointly models object appearance and hand pose during interactions. They demonstrate simultaneous hand action localization and object detection through implicit modeling of affordances.

Localizing object affordances based on supervised learning has been popular in the robotics community. [238] performs robotic manipulations on objects based on affordances which are inferred from the orientations of object surfaces. [239] learns a discriminative model to perform affordance segmentation of point clouds based on surface geometry. [131] uses RGBD data to learn pixelwise labeling of affordances for common household objects. They explore two different features: one based on a hierarchical matching pursuit (HMP) and another based on normal and curvature features derived from RGBD data. [240] learns to infer object level affordance labels based on attributes derived from appearance features. [254] proposes a two stage cascade approach based on RGBD data to regress potential grasp locations of objects. In [130], pixelwise affordance labels of objects are obtained by warping the query image to the K-nearest training images based on part locations inferred using DPMs. [255] combines top-down object pose based affordance labels with those obtained from bottom-up appearance based features to infer part based object affordances. Top-down approaches for affordance labeling has been explored in [214, 216] where scene labeling is performed by observing possible interactions between scene geometry and hallucinated human poses. Localizing object affordances based on human context has been also studied in [256]. They propose a graphical model where spatial and temporal extents of object affordances are in-

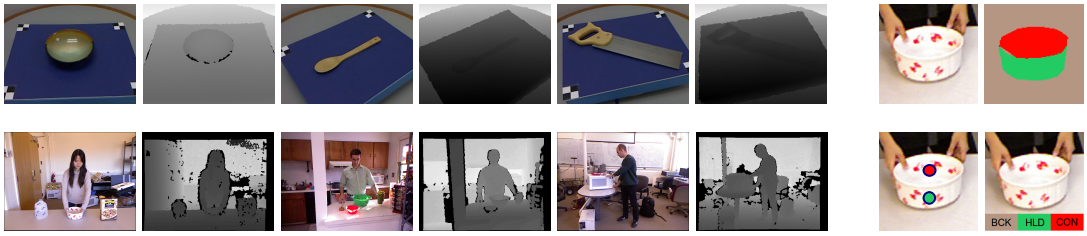


Fig. 6.2: (left) RGBD image pairs illustrating images from (top row) the UMD turntable affordance dataset and (bottom row) the CAD-120 dataset. (right) Illustrating the various levels of annotation (clockwise) original image, pixel level annotation, image level annotation, clickpoint level annotation.

ferred based on observed human pose and object locations. A mixture model is used to model temporal trajectories where each component represents a single type of motion *e.g.* repetitive or random motion. The approach, however, does not provide pixelwise segmentation. Instead, the coarse location of an affordance is described by a probability distribution.

Weakly-supervised learning for semantic image segmentation has been investigated in several works. In this context, training images are only annotated at the image level and not at pixel level. For instance, [257] formulate the weakly supervised segmentation task as a multi-instance multi-task learning problem. Further, [258, 259] incorporate latent correlations among superpixels that share the same labels but originate from different images. [260] simplifies the above formulation by a graphical model that simultaneously encodes semantic labels of superpixels and presence or absence of labels in images. [261] handles noisy labels from social images by using robust mid level representations derived through topic modeling in a CRF framework. More recently, a weakly-supervised approach based on a deep DCNN [262] has been proposed in [241]. It uses an EM framework to iteratively learn the latent pixel labels of the training data and the parameters of the DCNN. A similar approach is followed by [263] where linear constraints derived from weak image labels are imposed on the label prediction distribution of the neural network.

To investigate the problem of weakly labeled affordance segmentation, a pixel-wise labeled dataset that contains objects within the context of human-object interactions is first introduced in Section 6.3.

Next, various forms for weak labels are investigated and an EM framework that is adaptive to local image statistics is proposed in Section 6.4. In Section 6.4.4 it is shown that contextual information in terms of automatically extracted human pose can be utilized to initialize the EM framework thereby further reducing the need for labeled data. Finally, evaluations are presented in Section 6.5.

6.3. AFFORDANCE DATASETS

There are not many datasets with pixelwise affordance labels. The RGBD dataset proposed by [131] is an exception and focuses on part affordances of everyday tools. The dataset consisting of 28,074 images is collected using a Kinect sensor, which records RGB and depth images at a resolution of 640×480 pixels and provides 7-class pixelwise affordance labels for objects from 17 categories. Each object is recorded on a revolving turntable to cover a

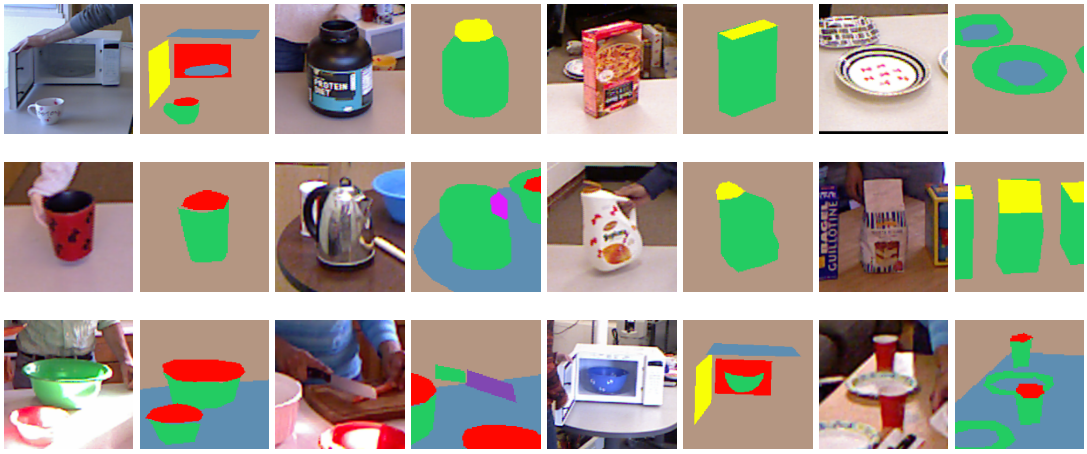


Fig. 6.3: Sample images from the proposed CAD-120 affordance dataset. The affordance labels are background (brown), holdable (green), openable (yellow), supportable (blue), containable (red), cuttable (purple) and pourable (magenta).

full 360° range of views providing clutter-free images of the object as shown in Figure 6.2. While such lab recordings provide images with high quality, they lack important contextual information such as human-interaction.

Therefore, a dataset that contains objects within the context of human-interactions in a more realistic environment is adopted. In this regard, the extended CAD-120 dataset [256] is a well tailored for this purpose. It consists of 215 videos in which 8 actors perform 14 different high level activities. Each high level activity is composed of sub-activities, which in turn involve one or more objects. In total, there are 32 different sub-activities and 35 object classes. A few images of the dataset are shown in Figure 6.2. The dataset also provides framewise annotation of the sub-activity, object bounding boxes and automatically extracted human pose.

Regarding annotation, affordance labels *openable, cuttable, containable, pourable, supportable, holdable* are annotated for every 10^{th} frame from sequences involving an active human-object interaction resulting in 3090 frames. Each frame contains between 1 and 12 object instances resulting in 9916 objects in total. All object instances are annotated with pixelwise affordance labels. A few images from the dataset are shown in Figure 6.3. As can be seen, the appearance of affordances can vary significantly, *e.g.* visually distinct object parts like the lid of a box, the cap of a bottle and the door of a microwave all have the affordance *openable*. Similarly, the interiors of a bowl and a microwave are *containable*.

Figure 6.4 presents statistics of affordance segments at the level of object bounding boxes. As can be seen, affordances *holdable, supportable* are most likely to occur because most interacted objects are handheld in the context of a supportive structure. Also, affordances like *openable, containable* which are a result of generic interactions have a fair chance to be observed. However, precise affordances like *cuttable, pourable* not only occur rarely, but also cover a minuscule portion of the object bounding box. All other affordances are well represented visually in that they cover at least 15% of the object bounding boxes, which have a median dimension of 68×57. The dataset is also well balanced in terms of the number of images contributed by each actor with a median of 382 and a range of 227–606 images.

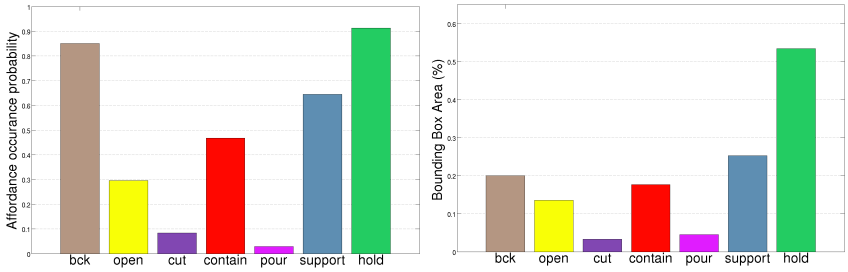


Fig. 6.4: Distribution of affordance labels at the object bounding box level in the proposed dataset (left) probability of observing an affordance (right) median area covered by an affordance segment in relation to its object bounding box.

6.4. PROPOSED METHOD

Supervised learning of affordances using appearance features has been addressed in [131, 238, 239, 240, 254, 255]. Recently, a supervised framework for object affordance labeling in RGBD images is proposed by [131]. The framework treats each class independently by learning standalone one-vs-all classifiers, affecting model scalability adversely. In this regard, the proposed method builds on DeepLab [262] which is a state-of-the-art end-to-end technique for semantic segmentation. DeepLab uses a DCNN to predict the label distribution per pixel, followed by a fully connected CRF to smooth predictions while preserving image edges.

The learning procedure is now described at various levels of supervision. Given an image $\mathbf{I}$ with n pixels, the image values are denoted as $X = \{x_1, x_2, \ldots, x_n\}$ where $x_i \in \mathbb{R}^3$ in case of RGB images. The corresponding labeling is denoted as $Y = \{y_1, y_2, \ldots, y_n\}$ where $y_i \in \mathcal{C}$ takes one of the $|\mathcal{C}|$ discrete labels $\mathcal{C} = \{1, 2, \ldots, |\mathcal{C}|\}$ with $y_i = 1$ indicating the background class. Note that these pixel level labels may not be available for the training set. Instead, two cases of weak annotations are considered. In the first case, a set of image level labels are provided. They are denoted by $Z = \{z_1, z_2, \ldots\}$, where $z_l \in \mathcal{C}$ and $\sum_i [y_i = z_l] > 0$, *i.e.* Z contains the classes that are present anywhere in the image. In the second case, an additional reference point in the image is provided for each $z \in Z$. We denote this by $Z_x = \{(z_1, p_1, x_1), (z_2, p_2, x_2), \ldots\}$ where $p_l \in \Omega$ is the pixel location with label z_l with value x_l. The latter case of weak annotation is based on single clickpoints annotated by users. This technique has been used to scale up the annotation process for a large-scale material database in [242]. Figure 6.2 illustrates the various levels of annotation.

The supervised learning based on [262] is briefly summarized in Section 6.4.1. In Section 6.4.2, an approach for weakly supervised learning is proposed and its initialization is discussed in Section 6.4.3. Finally, an approach that transfers annotations of the type Z_x to images with weaker annotations of type Z is proposed in Section 6.4.4. Automatically extracted human pose as context is exploited for the annotation transfer.

6.4.1. Pixel level annotation

In the fully supervised case, the objective function is the log likelihood given by

$$J(\theta) = \log p(Y|X;\theta) = \sum_{i=1}^{n} \log p(y_i|X;\theta), \tag{6.1}$$

where θ is the vector of DCNN parameters. The per-pixel label distribution is then given by

$$p(y_i|X;\theta) \propto \exp\left(f_i\left(y_i|X;\theta\right)\right), \tag{6.2}$$

where $f_i\left(y_i|X;\theta\right)$ is the output of the DCNN at pixel i. For optimizing $J(\theta)$, we adopt the implementation provided by [262].

6.4.2. Weak annotation

Considering the case when only weak image level annotation is available, the observed variables are image data X and image level labels Z. The second case Z_x is very similar and will be discussed in Section 6.4.3. The pixel level segmentation Y forms the latent variables. The proposed approach is based on the EM framework that has been proposed in [241]. While [241] introduces class dependent bias terms that are constant for an entire image, *i.e.* independent of the image location, this framework is extended to handle spatial dependencies. In this way, the method can not only use image level labels Z, but can also use weak annotations of the second type Z_x.

An EM approach is formulated in order to learn the parameters θ of the DCNN model, which is given by

$$\begin{aligned} p(X,Y,Z;\theta) &= p(Y|X,Z;\theta)\, p(X,Z) \\ &= \prod_{i=1}^{n} p(y_i|X,Z;\theta)\, p(X,Z). \end{aligned} \tag{6.3}$$

The M-step involves updating the model parameters θ by treating the point estimate $\hat{Y}$ as ground truth segmentation and optimizing

$$\sum_{Y} p(Y|X,Z;\theta^{old})\ \log p(Y|Z;\theta) \approx \log p(\hat{Y}|X;\theta) = \sum_{i=1}^{n} \log p(\hat{y}_i|X;\theta), \tag{6.4}$$

which can be efficiently performed by minibatch stochastic gradient descent (SGD) as in (6.1).

While the M-step is the same as in [241], the E-step differs since spatial dependencies of the label distribution conditioned on Z are modeled. The E-step amounts to computing the point estimate $\hat{Y}$ of the latent segmentation as

$$\hat{Y} = \underset{Y}{\operatorname{argmax}}\ \log p(Y|X,Z;\theta) = \underset{\{y_1,\ldots,y_n\}}{\operatorname{argmax}} \sum_{i=1}^{n} \log p(y_i|X,Z;\theta) \tag{6.5}$$

$$p(y_i|X,Z;\theta) = \begin{cases} \frac{f_{bg}}{f_{bg}+\sum_{z\in Z\setminus\{1\}}\sum_k \boldsymbol{\pi}_{z,k}\mathcal{N}(i;\boldsymbol{\mu}_{z,k}(X;\theta),\boldsymbol{\Sigma}_{z,k}(X;\theta))}, & \text{if } y_i = 1 \\ \frac{\sum_k \boldsymbol{\pi}_{y_i,k}\mathcal{N}(i;\boldsymbol{\mu}_{y_i,k}(X;\theta),\boldsymbol{\Sigma}_{y_i,k}(X;\theta))}{f_{bg}+\sum_{z\in Z\setminus\{1\}}\sum_k \boldsymbol{\pi}_{z,k}\mathcal{N}(i;\boldsymbol{\mu}_{z,k}(X;\theta),\boldsymbol{\Sigma}_{z,k}(X;\theta))}, & \text{if } y_i \in Z\setminus\{1\} \\ 0, & \text{otherwise} \end{cases} \tag{6.6}$$

where $\setminus$ indicates set subtraction. For the background class, a spatially constant probability f_{bg} is assumed. For affordances that are not part of the weak labels Z the probability is set to zero. The spatial distribution of an affordance $z \in Z$ is modeled by a Gaussian Mixture

distribution with weights $\boldsymbol{\pi}_{z,k}$, means $\boldsymbol{\mu}_{z,k}$ and covariance matrices $\boldsymbol{\Sigma}_{z,k}$, which depend on θ and X. Given the output of the DCNN, *i.e.* $p(y_i|X;\theta)$ from (6.2), the set of pixels that are labeled by z, *i.e.* $\{i : z = \text{argmax}_{y_i} p(y_i|X;\theta)\}$ are computed. A binary Grabcut segmentation is initialized with this set as foreground and the rest of the pixels as background. 8-neighbor connected regions of size larger than 10% of the largest region are considered to estimate parameters $\boldsymbol{\pi}_{z,k}$, $\boldsymbol{\mu}_{z,k}$ and $\boldsymbol{\Sigma}_{z,k}$.

6.4.3. Initialization

Two sets of training images are considered. While the first set is annotated by a set of clickpoints Z_x, a second set contains only image level labels Z as shown in Figure 6.2. The procedure starts with the first set annotated with Z_x and initialize the Gaussian Mixture for z_l by a single Gaussian with $\boldsymbol{\mu}_{z_l} = x_l$ and $\boldsymbol{\Sigma}_{z_l} = 40\boldsymbol{I}$ where $\boldsymbol{I}$ indicates an identity matrix. The E-step is performed to learn the initial point estimate $\hat{Y}$ using (6.5). The DCNN model is initialized by a pre-trained model VGG16 [264] and the model parameters are updated according to the M-step (6.4). The updated DCNN model is then applied to all training images to compute (6.2) before continuing with the E-step. For the clickpoints (z_l, x_l), the means of the Gaussians $\boldsymbol{\mu}_{z_l}$ are retained as x_l. The approach is then iterated until convergence.

6.4.4. Estimating clickpoints from human pose

Since affordances can be observed in the context of human-object interaction, clickpoint annotations from images can be transferred those that contain only image level labels. Given the automatically extracted 2d human pose and detected bounding boxes of objects, the human pose h is represented as a $2J$ dimensional vector of joint locations where J denotes the number of joints. For each affordance $z \in \mathcal{C}$, all annotated clickpoints x_t are collected together with the pose h_t as a training set. The pose vector h_t and x_t are normalized by subtracting the center of the object bounding box followed by mean and variance normalization over the training data, *i.e.* setting mean to zero and standard deviation to one. K-means clustering is then performed on these poses to learn a dictionary of size D, denoted as $\mathbf{h}_D$. For regressing the normalized clickpoint x of an affordance z given the normalized pose h, a regularized non-linear regression is used with an radial basis function (RBF) kernel

$$x = \boldsymbol{\theta}^T \boldsymbol{\psi}(h, \mathbf{h}_D) = \sum_{d=1}^{D} \theta_d \exp\left(-\frac{\|h - h_d\|_2^2}{\gamma^2}\right) \tag{6.7}$$

where h_d is the d^{th} entry of dictionary $\mathbf{h}_D$. The regression weight $\boldsymbol{\theta}$ is learned in the least squared error sense. Hyperparameter γ, regularization parameter λ and dictionary size D are learned through cross validation, which is obtained by splitting the training data in half.

6.5. EXPERIMENTS

Section 6.4.1 evaluates the fully supervised approach and compares it with other fully supervised approaches for affordance segmentation. Section 6.4.2 then presents the weakly supervised baselines and compares it against the fully supervised approaches. Evaluation is performed on two affordance datasets presented in Section 6.3. As for the evaluation protocol,

UMD Turntable	Grasp	Cut	Scoop	Contain	Pound	Support	Wgrasp	Mean
Fully Supervised Weighted F-Measure								
HMP + SVM	0.37	0.37	0.42	0.81	0.64	0.52	0.77	0.557
DEP + SRF	0.31	0.28	0.41	0.63	0.43	0.48	0.66	0.457
DeepLab	0.59	0.71	0.55	0.90	0.33	0.70	0.87	0.664
Fully Supervised IOU								
HMP + SVM	0.31	0.11	0.07	0.30	0.06	0.06	0.17	0.154
DEP + SRF	0.26	0.01	0.06	0.28	0.04	0.03	0.19	0.124
DeepLab	0.51	0.63	0.49	0.85	0.26	0.63	0.80	0.596
Weakly Supervised Weighted F-Measure								
Weak+DeepLab [241]	0.09	0.11	0.14	0.30	0.12	0.20	0.27	0.176
Proposed	0.47	0.66	0.50	0.84	0.29	0.59	0.70	0.579
Weakly Supervised IOU								
Weak+DeepLab [241]	0.09	0.10	0.13	0.27	0.10	0.18	0.23	0.157
CCNN [263]	0.09	0.00	0.00	0.31	0.00	0.00	0.29	0.099
Proposed + EM(1 Iter.)	0.28	0.38	0.30	0.61	0.24	0.44	0.60	0.407
Proposed	0.40	0.57	0.43	0.78	0.22	0.52	0.63	0.507

Table 6.1: Evaluating fully supervised approaches for affordance segmentation on the UMD turntable dataset. Evaluation metrics based on weigted F-measure and IOU. HMP+SVM and DEP+SRF are proposed in [131] and DeepLab in [262].

predefined train-test split for the UMD turntable dataset is employed. Regarding the CAD-120 affordance dataset, images from actors $\{5, 9\}$ are reserved as $test$ and images from actors $\{\{1, 6\}, \{3, 7\}, \{4, 8\}\}$ are referred to as $\{TrainA, TrainB, TrainC\}$ respectively. Further, the union of the three training sets is indicated as *allTrain*.

For quantitative evaluation, the measure reported is per class IOU, also known as Jaccard index, for both datasets. Since [131] reports performance in terms of a Weighted F-Measure, this metric is also reported for the UMD turntable dataset.

6.5.1. UMD Turntable Dataset

6.5.1.1. Supervised training

In [131], two approaches have been presented for learning affordances from local appearance and geometric features. The first approach is based on features derived from a superpixel based hierarchical matching pursuit (HMP) together with a linear SVM and the second approach is based on curvature and normal features derived from depth data used within a structured random forest (SRF). For the fully supervised setting, DeepLab [262] is initialized by VGG16 [264] and trained by SGD with a mini-batch of 6 images and an initial learning rate of 0.001 (0.01 for the final classifier layer), multiplying the learning rate by 0.1 after every 2000 iterations. Momentum is set to 0.9, weight decay to 0.0005 and the network is trained for 6000 iterations. The performance comparison on both IOU and weighted F-measure metrics are shown in Table 6.1.

As can be observed, the trend in performance is similar irrespective of the evaluation metric. The `HMP+SVM` consistently outperforms the `DEP+SRF` combination, indicating that learning features from data is more effective than learning complex classifiers on handcrafted features. DeepLab in turn outperforms `HMP+SVM` in almost all classes reconfirming the ef-

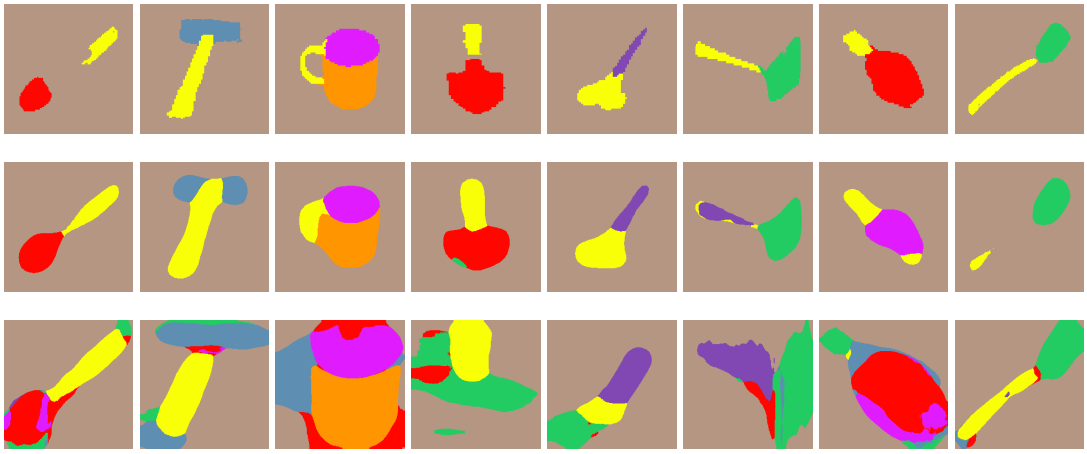


Fig. 6.5: Results from supervised learning on UMD turntable dataset using (top) ground truth segmentation (middle) DeepLab (bottom) DEP+SRF.

fectiveness of end-to-end learning. However, in spite of powerful learning techniques, the performance for small affordance regions like *pound*, *support* is considerably low. A few qualitative results are shown in Figure 6.5.

6.5.1.2. Weakly supervised training

In this setting, weak labels are derived from the pixel-annotated dataset. While the centroids of 8-connected components of labeled foreground regions are used as clickpoint annotations Z_x, image level labels Z are computed as a union of pixel-wise labels. Baseline performances of weakly supervised approaches are presented in Table 6.1. The method [241] is trained using image labels on the training set with fg–bg bias set to 0.3–0.2. This performs with a mean accuracy of 0.16 (`Weak+DeepLab`). Further, another similar approach [263] performs similarly with a mean accuracy of 0.10 (`CCNN`). Both these approaches underperform in comparison with the proposed method (`Proposed+EM(1 Iter.)`) which is trained for a single E- and M-step using clickpoint annotations on the training set. The proposed method converges in two iterations with a performance of 0.51 IOU (`Proposed`).

6.5.2. CAD-120 Affordance Dataset

Since object bounding boxes in the dataset are pre-annotated, all experiments are performed on cropped images after extending the bounding boxes by (a maximum of) 30px in each direction.

6.5.2.1. Supervised training

Firstly, the fully supervised approach is evaluated and compared. In this setting, the entire training data *allTrain* with pixelwise label annotations is used as training set. The approaches proposed in [131] are based on depth data. However, owing to noisy depth images from the dataset, the performance of both approaches with depth features is found to be substantially lower than that with CNN features. Therefore, the accuracy of the best setting is reported, which is obtained by using features of all layers [66] of the VGG16 [264] network for the

SRF. It must be noted that DeepLab [241] is also finetuned from a similar network. Referring to Table 6.2, it can be seen that DeepLab performs at a mean IOU of 0.42 whereas the SRF based approach performs at 0.26. In spite of the high quality of annotation and the rich model, the accuracy for the classes *cut* and *pour* is almost zero for all methods. This is because of the small spatial extent of both classes and the lack of training data (see Figure 6.4). Qualitative results for both methods are presented in Figure 6.6.

6.5.2.2. Weakly supervised training with clickpoint annotations

In the second setting, pixel-wise label annotations are replaced by clickpoint annotations *i.e.* the entire *allTrain* is used with Z_x annotations as the training set. For the first experiment, the proposed approach is initialized as described in Section 6.4.3 where Gaussians are initialied based on the clickpoints and a single E- and M-step is performed. This is denoted as `allTrain+EM(1 iter.)` in Table 6.2. Compared to the fully supervised setting, the mean accuracy decreases from 0.42 to 0.28. The proposed EM approach is found to converge within 3–4 iterations resulting in increased accuracy for all classes. This is denoted as `allTrain+EM`. The largest improvement can be observed for the class *support*, which increases from 0.35 to 0.44. The mean accuracy increases from 0.28 to 0.31.

In (6.5), the spatial distributions of the affordances are modeled by Gaussian mixture distributions. As a heuristic, the output of the Grabcut segmentation can also be used as $\hat{Y}$. This approach is denoted as `allTrain+EM(1 iter.)+onlyGC`. Similarly, the Grabcut segmentation can be skipped and Gaussian mixture parameters can be estimated directly from (6.2), denoted as `allTrain+EM(1 iter.)+onlyGM`. The substantial drop in performance in both cases indicates that these components are critical for performance.

6.5.2.3. Weakly supervised training with mixed clickpoint and image annotations:

In the third setting, there exist two sets of training data. The first set is annotated by clickpoints Z_x and the second set is annotated by image labels Z. An evaluation averaged over three splits is presented. For a split, one of the subsets *trainA*, *trainB*, or *trainC* is annotated with Z_x and the other subsets are annotated with Z.

To begin with, the proposed approach is trained only on *trainX*, *i.e.* the subset annotated with Z_x without using the training images annotated with Z. The approach is denoted as `TrainXOnly+EM(1 iter.)` in Table 6.2. As expected, the reduction of training data by one third decreases the mean accuracy from 0.28 to 0.22. Running the proposed EM approach until convergence improves the results by 3% to 0.25, denoted as `TrainXOnly+EM`. The approach serves as baseline for other weakly supervised approaches that use additional training data annotated by Z.

The weakly supervised approach is compared with [241]. The method is initialized on *TrainX* in the same way as the proposed method and the fg–bg bias is set to 0.3–0.2. The best result is achieved using the semi-supervised mode of [241] where the initial segmentation results on *TrainX* are not changed. This performs with a mean accuracy of only 0.16 (`Semi+TrainX+DeepLab`), which is lower than the baseline `TrainXonly+EM`. This shows that constant bias terms proposed for the E-step in [241] are insufficient for the task of affordance segmentation. Further, another weakly supervised approach [263] performs with

Experiment Setting	Bck	Open	Contain	Support	Hold	Mean
Supervised training (allTrain)						
DeepLab [262]	0.75	0.46	0.52	0.64	0.60	0.42
VGG + SRF [131]	0.62	0.20	0.22	0.39	0.39	0.26
Weakly supervised training (state-of-the-art)						
Semi+TrainX+DeepLab [241]	0.42	0.17	0.09	0.26	0.20	0.16
CCNN [263]	0.46	0.60	0.10	0.25	0.15	0.14
Weakly supervised training (allTrain with clickpoints)						
allTrain EM(1 iter.)	0.65	0.27	0.30	0.35	0.38	0.28
allTrain EM	0.67	0.29	0.34	0.44	0.42	0.31
allTrain EM(1 iter.)+onlyGC	0.60	0.19	0.19	0.23	0.30	0.22
allTrain EM(1 iter.)+onlyGM	0.40	0.19	0.09	0.27	0.20	0.16
Weakly supervised training (trainX with clickpoints)						
TrainXOnly+EM(1 iter.)	0.48	0.17	0.20	0.41	0.31	0.22
TrainXOnly+EM	0.58	0.20	0.26	0.33	0.40	0.25
Weakly supervised training (trainX with clickpoints, rest with image labels)						
TrainX+Pose+EM(1 iter.)	0.61	0.22	0.24	0.33	0.37	0.25
TrainX+Pose+EM	0.63	0.21	0.24	0.39	0.39	0.27
TrainX+EM	0.38	0.17	0.21	0.39	0.29	0.21
TrainX+BB+EM(1 iter.)	0.34	0.14	0.12	0.26	0.17	0.15

Table 6.2: Evaluating affordance segmentation on the CAD-120 affordance dataset under various settings. The evaluation metric used is IOU. While the mean is computed over all classes, class results are shown only for a subset of classes.

a mean accuracy of 0.14 (`CCNN`). Although the method also uses weak labels with respect to region sizes, its performance does not surpass that of [241].

The effect of transferring clickpoints is now studied. In this regard, clickpoints Z_x from *TrainX* are transferred to other images in *allTrain* using the method described in Section 6.4.4. The parameters of the regression in (6.7) are obtained by cross validation. This resulted in $D = 200$, $\gamma = 10$, $\lambda = 1$ which are used for all experiments. Referring to `TrainX+Pose+EM(1 iter.)`, the clickpoint transfer based on human context followed by a single EM iteration improves the mean accuracy by 3% to 0.25. The performance further increases to 0.27 when iterated until convergence. The same experiment performed without clickpoint regression, *i.e.* using *TrainXOnly* for the first EM iteration and using all training images for further iterations resulted in a slight drop in performance to 0.21 (`TrainX+EM`).

In order to demonstrate that the performance gain is indeed from using human pose, the above experiment is repeated but clickpoints are regressed from object bounding boxes instead of human pose. To this end, the $2J$ dimensional pose vector h_t is replaced by a 6d vector of the bounding box consisting of the x- and y-coordinates of the top left corner, width and height of the object bounding box. This setting, tabulated as `TrainX+BB+EM`, performs substantially worse, showing that human pose provides a valuable source for weakly supervised learning of affordances.

Figure 6.6 presents qualitative results of various discussed approaches presented in Table 6.2. In the supervised setting, segmentation generated by `DeepLab` are greatly superior to those generated from `VGG+SRF`, which performs poorly even for affordances with large spatial extent like *containable, openable*. However, both approaches perform poorly for difficult affordances like *cuttable*. Regarding weakly supervised setting with clickpoints, there

is a visible drop of quality for `allTrain+EM(1 Iter.)` when compared to the fully supervised approach as expected. A further degradation is seen with `TrainXOnly+EM(1 Iter.)` due to the reduced training data. Comparing weakly supervised approaches with mixed annotations, `Semi+TrainX+DeepLab`[262] allocates equally sized segments for all affordances. By contrast, improvements due to the proposed EM approach can already be seen for `TrainXOnly+EM(1 Iter.)`. Further, improved spatial localization of affordances due to regressing clickpoints from human pose is seen for `TrainX+HPose+EM(1 Iter.)` which is further refined by `TrainX+HPose+EM`. Finally, the last row shows the poor performance of `TrainX+BB+EM(1 Iter)`. When compared with `TrainX+HPose+EM(1 Iter.)`, this indicates the importance of clickpoint transfer based on human pose.

6.6. SUMMARY

This chapter addresses the problem of weakly supervised affordance segmentation. To this end, an expectation-maximization approach that can be trained on weak clickpoint annotations is proposed. In addition, it is shown how contextual information from human-object interaction can be used to transfer such annotations to images with only image level annotations. This improved the segmentation accuracy of the proposed EM approach substantially. For evaluation, a pixel-wise annotated affordance dataset containing 3090 images and 9916 object instances with rich contextual information is introduced.

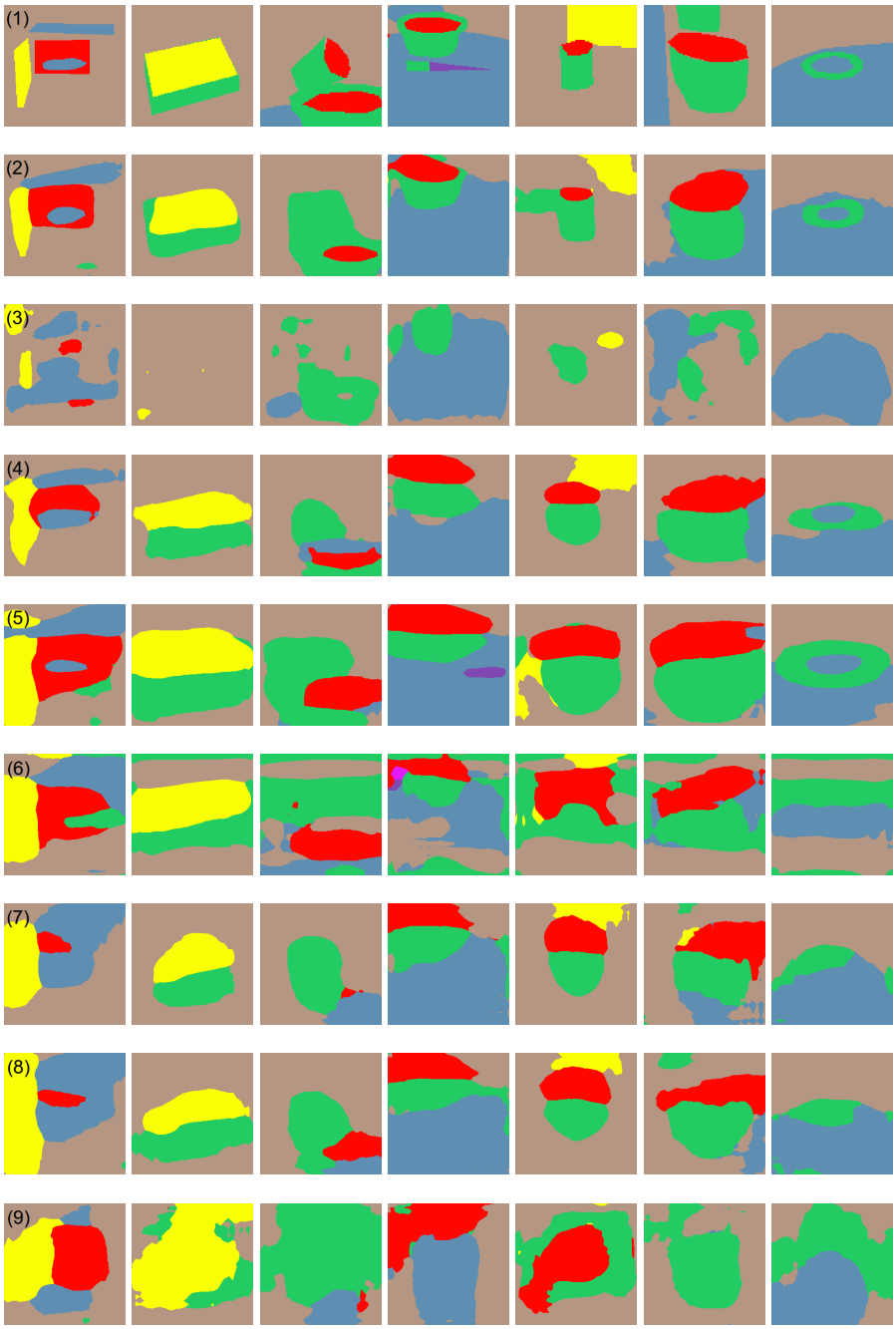


Fig. 6.6: Results from CAD-120 affordance dataset as presented in Table 6.2. (1) Ground truth (2) Supervised `DeepLab`[262] (3) Supervised `VGG+SRF`[131] (4) `allTrain+EM(1 Iter.)` (5) `TrainX+EM(1 Iter.)` (6) `Semi+TrainX+DeepLab`[241] (7) `TrainX+Pose+EM(1 Iter.)` (8) `TrainX+Pose+EM` (9) `TrainX+BB+EM(1 Iter.)` Image best viewed in color.

CHAPTER 7

Conclusion

Contents

Computer vision, which aims at *understanding* digital images, has been crucial for ushering new technologies such as virtual reality and robot assisted living where the ability to detect and reason about objects is indispensable. Although humans perform such tasks effortlessly, replicating such skills into computing agents has proved to be challenging. This is because modeling the interplay between real world phenomena *e.g.* object deformations, interactions and occlusions and imaging induced flaws *e.g.* truncation, noise and motion blur is a perplexing problem. In practice, this is dealt using machine learning techniques that are trained to infer tight bounding boxes around instances of object categories. Current state-of-the-art methods are based on large annotated datasets in conjunction with high capacity models. While these are highly successful in detecting large objects which admit reliable visual evidence, the case of small and medium sized household objects is still an open problem.

This dissertation addresses the problem of characterizing such household objects under human interaction where an actor performs high level activities using (possibly multiple) object classes *e.g. making cereal* involves a box and a bowl (cf. Fig 1.1). In such scenarios, detection performance of traditional appearance based techniques is limited due to unreliable visual evidence, varying object pose, occlusions and motion blur, as verified by our experiments (cf. Sections 4.5.2–4.5.4). However, the presence of human context is ubiquitous, aptly forming the central theme of this dissertation. More specifically, we investigate the utility of human pose in *characterizing* objects *i.e.* detecting objects and localizing their functional regions.

To this end, we first consider the problem of *object detection* at various levels of supervision. For full supervision, we show that appearance and human pose based detectors can be combined for improved performance. As for combining both modalities, our experiments show that a late combination is best suited due to inherently disparate localization capabilities of both features. To address the demanding annotation requirements of the fully supervised setup, we turn to weakly labeled activity videos as a source of object examples. We show that both appearance and motion is primarily dominated by humans and are insufficient to describe objects. In addition, we show that greatly improved performance can be achieved by incorporating human pose centric object functionality (cf. Section 5.4.2).

Further, we observe that formalizing the localization aspect of object detection is somewhat unclear. While bounding box level information might suffice for many applications, detailed pixelwise descriptions might be more suitable for others. To address this concern, we turn to *segmenting affordances* or functional regions within objects. We cast the problem

as that of semantic segmentation and tackle laborious pixelwise annotation needed by using image and clickpoint level annotations (cf. Sections 6.5.1–6.5.2). Further, the utility of human pose is demonstrated by transferring clickpoint annotations to new images yielding improved performance.

The generalization capabilities of the proposed methods are mainly evaluated on (a subset of) three human-object interaction datasets that span a variety of different activities, modalities and human pose representations.

7.1. CONTRIBUTIONS

In detail, the dissertation addresses the following questions in order to characterize objects in images.

Improving object detection with mid level appearance representation: The Hough-based object detection framework assumes independence within patch-based individual features for computational efficiency. In Chapter 3, we investigate an improved mid level appearance representation by grouping patch-based features over a local neighborhood. We show that small datasets and highly specific individual features cause grouped features to overfit. While this is partly alleviated through oblique instead of axis-aligned forests, the benefits of large training data is seen in VOCB3DO dataset where grouped features consistently outperform individual features, unlike in the smaller ETHZ dataset. Further, improved performance is obtained by linearly combining both features thus revealing their complementary nature. Also, parallel developments in end-to-end feature learning using large annotated datasets and high-capacity models have been successful in establishing reliable mid level features.

Comparing star and tree object models: As discussed in Section 1.3.1, the star model considers the cumulative evidence of object parts. The tree model, on the other hand, uses finer object part annotations and explicitly considers their relative locations during inference. In order to ascertain their strengths and limitations, we evaluated both models on three challenging human-object interaction datasets (cf. Sections 4.5.2–4.5.4). Considering overall performance, the tree model outperforms the star model with $3\% - 10\%$ relative improvement which is more pronounced for articulated objects *e.g. microwave* and *videogame*.

Using human pose to improve supervised object detection: During human-object interactions, human pose can be indicative of the object involved *e.g.* routine actions such as "making a phone call" or "clicking a photo" would involve *phone* and *camera* respectively. Moreover, human interactions result in occlusions and poor visual representations therefore forcing appearance-only based methods to underperform. To alleviate this limitation, we investigate ubiquitously available human pose as an alternative modality for object detection. Our experiments show that while human pose features are able to reasonably classify objects, they are intrinsically unable to resolve finer details regarding their location. This is verified by the poor performance of detectors based on human pose which are (optionally) concatenated with appearance features (cf. Section 4.5.2–4.5.4). Furthermore, improved performance can be achieved by linearly combining a separate detector for each modality. We observe that the linear coefficients are class dependent and that the performance of the combined model is independent of the underlying human pose estimation technique.

Unifying human pose estimation and object detection in a single model: As discussed previously, improved object detection performance has been achieved by using pose estimates obtained independently by a model tuned for human pose estimation. In order to investigate if both problems can be solved jointly in a unified framework, we extend the tree model for human pose estimation to include objects (and their parts, cf. Section 4.5.2). We observe that the ensuing results are similar those from state-of-the-art methods although the previously discussed cascade system outperforms the unified model.

Using human pose to improve weakly supervised object detection: To address the demanding annotation requirements of the fully supervised setup, we turn to learning object models with reduced supervision. To this end, object detectors are trained using instances automatically detected from weakly labeled activity videos. We observe that frameworks that entirely rely on object appearance or motion perform poorly indicating the complexity of the task. We show that human centric object functionality encodes complementary information and results in greatly improved performance (cf. Sections 5.4.2–5.4.4). In order to maximize the utility of data, we model to infer multiple object instances from each video. The inference follows a greedy procedure resulting in tubes of superior quality, as verified by the experiments (cf. Section 5.4.2). The weakly supervised approach also outperforms an unsupervised approach and performs between $86\% - 92\%$ of its fully supervised counterpart.

Going beyond bounding boxes: The common consensus towards localizing objects in images is through bounding boxes. Although this might be sufficient for numerous applications, other applications can utilize finer part based information. To this end, we turn to deducing pixel wise affordance or object functionality within object bounding boxes. This is formulated as a (weakly) supervised semantic segmentation problem. We observe that while DCNN already yields state-of-the-art results, they require exorbitant annotation efforts. Therefore, we propose a weakly supervised approach that uses less expensive image level and/or clickpoint annotations. This performs between $50\% - 74\%$ of its fully supervised counterpart (cf. Sections 6.5.1–6.5.2). Further, we show that human context can help transfer clickpoint annotations to examples with only image level annotations.

New datasets and annotations: While generic object detection has seen steady advances in the recent past, the case for small and medium sized objects under human interaction has seen little consideration. In this regard, conceiving and designing datasets is indispensable for building and comparing methods in the present era of data driven techniques. In line with extending standard protocols, we adopt three human-object interaction datasets to train and evaluate various object detection techniques. To this end, we have annotated objects and their parts with bounding boxes and keypoints respectively (cf. Section 4.5). Further, we have adopted the CAD-120 dataset for affordance reasoning by designing a pixel wise annotated dataset containing 3090 images and 9916 object instances with rich human-contextual information (cf. Section 6.3). We intend to make these annotations publicly available.

7.2. PERSPECTIVES

Generalizing human context: In this dissertation, the datasets considered are based on single actors in lab environments. Generalizing approaches to human-object interactions *in*

the wild would need to reconcile varying resolutions of pose estimates (*e.g.* upper body, full body pose) and consider multiple persons (*e.g.* in [265], for human pose estimation). Further, human context is represented as a coarse body pose (bodypart locations) owing to significant strides in the domain. Although deducing finer pose details *e.g.* hand grasp, head pose etc. might be challenging due to poor resolution, appearance information can serve as an intermediary (*e.g.* in [266], for action recognition), but is currently forgone.

To invest in data or modeling? In this dissertation, we have used relatively simple linear models to combine appearance and contextual human pose information. However, understanding objects from generic video data is a challenging task. There is a wealth of real world video data available online in public archives. Such data, however, is ridden with myriad varieties of noise *e.g.* marketing, entertainment and education media are composed of fundamentally different camera work and visual distractions. Inferring from such data should therefore combine more sophisticated data driven and model driven techniques. This is because while some problems *e.g.* localizing regions of interest, identifying visual distractions etc. are more suited to learning from data, other problems *e.g.* noise robustness, transfer learning etc. are traditionally treated via modeling.

Detecting objects from activities: In Section 5.3, we have discussed a weakly supervised approach for object detection. Here, a set of object proposals are first generated using a simple generative model that encodes human interaction. Borrowing inspiration from recent works on object proposals [82] and tracking, similar strategies can be tuned towards human-object interaction centric proposals. Also, at present, weak labels are assumed to be decisively indicative of object classes present in each activity. Relaxing this assumption is vital to extend the framework to more generic videos as discusses above. This can be achieved by incorporating cues from unsupervised clustering as used in [261]. Further, present measures of functional similarity based on trajectory information can prove to be inadequate for generic videos. Improved measures that incorporate 2d–3d pose estimates [267] and/or scene affordances are promising directions.

Objects, humans and activities: We have shown in Section 4.5.2 that a single unified model for human pose estimation and object detection performs poorly. This is because of the inherent difficulty in modeling spatial distributions between parts of humans and objects. Recent advances in data driven techniques for human pose estimation [268] offers a promising direction where contextual information is naturally integrated into learning features from images within deep neural architectures. Further, the independence assumptions imposed by tree models to represent spatial priors between parts can be restrictive and more generalized densely connected graphs might be a more suitable approach. Further, previous works have demonstrated the benefit of action priors on human pose estimation [269, 270] and object priors on action recognition [223, 224]. This motivates a broader unified formulation of all three problems as they are characteristically similar and can provide potentially complementary information to each other.

Temporal evolution of objects: Traditionally, objects are not assumed to undergo topological changes. This assumption also applies to approaches presented in Chapter 4 and Chapter 5.

Consequently, modeling cases that violate this assumption *e.g.* dicing a tomato, slicing a pizza etc. is an open problem. Prior work in [271] detects singularities caused by topological changes by analyzing the displacement field of the underlying deformable model which are then applied to a physics based model to track the object. Similarly, deformable tracking in a household scenario is presented in [272]. Others resort to using mixture models [35] which is further improved by imposing temporal order by grammar based models [180] or HMM [273]. However, end-to-end learning of suitable discriminative models requires further attention.

Inferring multiple affordance labels per pixel: In Chapter 6, we have assumed that a single affordance is associated with each pixel. This simplification allows for adapting popular semantic segmentation techniques for affordance segmentation. However, this is not true in practice as object parts can serve multiple functionalities. E.g., the mouth of a cup can be both *containable* and *pourable*. To this end, it would be fruitful to gather data in this more realistic scenario and tune a suitable loss function.

Bibliography

[1] K. Schwab, "The fourth industrial revolution," in *World Economic Forum*, 2016. (Cited on page 3.)

[2] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *IJCV*, vol. 88, pp. 303–338, 2010. (Cited on pages 5, 10, 32, 41 and 45.)

[3] R. Girshick, "Fast r-cnn," in *ICCV*, pp. 1440–1448, 2015. (Cited on pages 5, 39, 46, 47, 48 and 51.)

[4] S. J. Dickinson, *Object categorization: computer and human vision perspectives*. Cambridge University Press, 2009. (Cited on page 5.)

[5] K. Saenko and T. Darrell, "Unsupervised learning of visual sense models for polysemous words," in *NIPS*, pp. 1393–1400, 2009. (Cited on page 5.)

[6] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *PAMI*, vol. 24, no. 7, pp. 971–987, 2002. (Cited on pages 7 and 9.)

[7] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection.," in *CVPR*, pp. 886–893, 2005. (Cited on pages 7 and 9.)

[8] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991. (Cited on pages 7 and 8.)

[9] P. Viola and M. J. Jones, "Robust real-time face detection," *IJCV*, vol. 57, no. 2, pp. 137–154, 2004. (Cited on page 8.)

[10] C. Redondo-Cabrera and R. J. López-Sastre, "Because better detections are still possible: Multi-aspect object detection with boosted hough forest.," in *BMVC*, pp. 63–1, 2015. (Cited on page 8.)

[11] H. Ren and Z.-N. Li, "Basis mapping based boosting for object detection," in *CVPR*, pp. 1583–1591, 2015. (Cited on page 8.)

[12] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-svms for object detection and beyond," in *ICCV*, pp. 89–96, 2011. (Cited on page 8.)

[13] C. Gu, P. Arbeláez, Y. Lin, K. Yu, and J. Malik, "Multi-component models for object detection," in *ECCV*, pp. 445–458, 2012. (Cited on page 8.)

[14] N. Razavi, J. Gall, P. Kohli, and L. Van Gool, "Latent hough transform for object detection," in *ECCV*, pp. 312–325, 2012. (Cited on pages 8 and 27.)

[15] X. Wang, M. Yang, S. Zhu, and Y. Lin, "Regionlets for generic object detection," in *ICCV*, pp. 17–24, 2013. (Cited on page 8.)

[16] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *ECCV-Workshops*, vol. 1, pp. 1–2, 2004. (Cited on page 8.)

[17] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, vol. 2, pp. 2169–2178, 2006. (Cited on page 9.)

[18] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *IJCV*, vol. 77, no. 1-3, pp. 259–289, 2008. (Cited on pages 9, 27 and 29.)

[19] S. Maji and J. Malik, "Object detection using a max-margin hough transform.," in *CVPR*, pp. 1038–1045, 2009. (Cited on pages 9, 27, 32, 34 and 36.)

[20] J. Gall and V. Lempitsky, "Class-specific hough forests for object detection.," in *CVPR*, pp. 1022–1029, 2009. (Cited on pages 9, 13 and 17.)

[21] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky, "Hough forests for object detection, tracking, and action recognition," *PAMI*, pp. 2188–2202, 2011. (Cited on pages 9, 13, 17, 18, 27, 30, 32, 36, 43, 45, 46, 47, 48 and 53.)

[22] S. Schulter, P. Wohlhart, C. Leistner, A. Saffari, P. Roth, and H. Bischof, "Alternating decision forests," in *CVPR*, pp. 508–515, 2013. (Cited on pages 9 and 27.)

[23] S. Schulter, C. Leistner, P. Wohlhart, P. M. Roth, and H. Bischof, "Alternating regression forests for object detection and pose estimation," in *ICCV*, pp. 417–424, 2013. (Cited on page 9.)

[24] S. Schulter, C. Leistner, P. Wohlhart, P. M. Roth, and H. Bischof, "Accurate object detection with joint classification-regression random forests," in *CVPR*, pp. 923–930, 2014. (Cited on page 9.)

[25] N. Razavi, J. Gall, and L. Van Gool, "Scalable multi-class object detection," in *CVPR*, pp. 1505–1512, 2011. (Cited on page 9.)

[26] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *CVPR*, pp. 1–8, 2008. (Cited on pages 9, 12, 39, 52 and 53.)

[27] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models.," *PAMI*, vol. 32, pp. 1627–1645, 2010. (Cited on pages 9, 27, 34 and 35.)

[28] M. Simon and E. Rodner, "Neural activation constellations: Unsupervised part model discovery with convolutional networks," in *ICCV*, pp. 1143–1151, 2015. (Cited on pages 9 and 12.)

[29] H. O. Song, Y. J. Lee, S. Jegelka, and T. Darrell, "Weakly-supervised discovery of visual pattern configurations," in *NIPS*, pp. 1637–1645, 2014. (Cited on page 9.)

[30] C. Zhou and J. Yuan, "Non-rectangular part discovery for object detection.," in *BMVC*, 2014. (Cited on page 9.)

[31] I. Endres, K. J. Shih, J. Jiaa, and D. Hoiem, "Learning collections of part models for object recognition," in *CVPR*, pp. 939–946, 2013. (Cited on page 9.)

[32] X. Zhu, D. Anguelov, and D. Ramanan, "Capturing long-tail distributions of object subcategories," in *CVPR*, pp. 915–922, 2014. (Cited on pages 9 and 12.)

[33] B. Hariharan, C. L. Zitnick, and P. Dollár, "Detecting objects using deformation dictionaries," in *CVPR*, pp. 1987–1994, 2014. (Cited on page 9.)

[34] W. Ouyang, X. Wang, X. Zeng, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, C.-C. Loy, *et al.*, "Deepid-net: Deformable deep convolutional neural networks for object detection," in *CVPR*, pp. 2403–2412, 2015. (Cited on page 9.)

[35] C. Desai and D. Ramanan, "Detecting actions, poses, and objects with relational phraselets," in *ECCV*, pp. 158–172, 2012. (Cited on pages 9, 11, 39, 40, 45, 46, 47, 48, 51, 52 and 91.)

[36] B. Pepik, M. Stark, P. Gehler, and B. Schiele, "Occlusion patterns for object class detection," in *CVPR*, pp. 3286–3293, 2013. (Cited on pages 9 and 52.)

[37] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille, "Detect what you can: Detecting and representing objects using holistic models and body parts," in *CVPR*, 2014. (Cited on pages 9 and 52.)

[38] T. Wang, X. He, and N. Barnes, "Learning structured hough voting for joint object detection and occlusion reasoning.," in *CVPR*, pp. 1790–1797, 2013. (Cited on pages 9 and 36.)

[39] Q. Wu and P. Hall, "Modelling visual objects invariant to depictive style.," in *BMVC*, 2013. (Cited on page 9.)

[40] C. Lu, Y. Lu, H. Chen, and C.-K. Tang, "Square localization for efficient and accurate object detection," in *ICCV*, pp. 2560–2568, 2015. (Cited on page 9.)

[41] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, pp. 580–587, 2014. (Cited on pages 9 and 53.)

[42] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NIPS*, pp. 91–99, 2015. (Cited on page 9.)

[43] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016. (Cited on page 9.)

[44] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *CVPR*, pp. 648–656, 2015. (Cited on page 9.)

[45] B. Pepik, R. Benenson, T. Ritschel, and B. Schiele, "What is holding back convnets for detection?," in *GCPR*, pp. 517–528, 2015. (Cited on pages 9 and 52.)

[46] X. Zhu, C. Vondrick, D. Ramanan, and C. Fowlkes, "Do we need more training data or better models for object detection?.," in *BMVC*, 2012. (Cited on page 9.)

[47] C. Harris and M. Stephens, "A combined corner and edge detector.," in *Alvey vision conference*, vol. 15, p. 50, 1988. (Cited on page 9.)

[48] D. G. Lowe, "Object recognition from local scale-invariant features," in *ICCV*. (Cited on page 9.)

[49] D. Engel and C. Curio, "Shape centered interest points for feature grouping," in *CVPR-Workshops*, pp. 9–16, 2010. (Cited on page 9.)

[50] S. K. Ravindran and A. Mittal, "Comal: Good features to match on object boundaries," in *CVPR*, 2016. (Cited on page 9.)

[51] H. Ren and Z.-N. Li, "Object detection using generalization and efficiency balanced co-occurrence features," in *ICCV*, pp. 46–54, 2015. (Cited on pages 9 and 11.)

[52] Z. Ren and E. B. Sudderth, "Three-dimensional object detection and layout prediction using clouds of oriented gradients," in *CVPR*, 2016. (Cited on page 9.)

[53] J. Yan, Z. Lei, L. Wen, and S. Z. Li, "The fastest deformable part model for object detection," in *CVPR*, pp. 2497–2504, 2014. (Cited on page 9.)

[54] T. Dean, M. A. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik, "Fast, accurate detection of 100,000 object classes on a single machine," in *CVPR*, pp. 1814–1821, 2013. (Cited on page 9.)

[55] X. Ren and D. Ramanan, "Histograms of sparse codes for object detection," in *CVPR*, pp. 3246–3253, 2013. (Cited on page 9.)

[56] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from rgb-d images for object detection and segmentation," in *ECCV*, pp. 345–360, 2014. (Cited on pages 9 and 11.)

[57] A. Janoch, S. Karayev, Y. Jia, J. Barron, M. Fritz, K. Saenko, and T. Darrell, "A category-level 3d object dataset: Putting the kinect to work," in *Consumer Depth Cameras for Computer Vision*, pp. 141–165, 2013. (Cited on pages 9, 35 and 36.)

[58] A. Khoreva, R. Benenson, M. Omran, M. Hein, and B. Schiele, "Weakly supervised object boundaries," in *CVPR*, 2016. (Cited on page 9.)

[59] P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," in *ICCV*, pp. 1841–1848, 2013. (Cited on page 9.)

[60] H. Riemenschneider, M. Donoser, and H. Bischof, "Using partial edge contour matches for efficient object category localization.," in *ECCV*, pp. 29–42, 2010. (Cited on pages 9, 28 and 34.)

[61] V. Ferrari, F. Jurie, and C. Schmid, "From images to shape models for object detection.," *IJCV*, vol. 32, pp. 284–303, 2010. (Cited on pages 9, 28, 32 and 34.)

[62] P. Yarlagadda, A. Monroy, and B. Ommer, "Voting by grouping dependent parts," in *ECCV*, pp. 197–210, 2010. (Cited on pages 9, 28 and 36.)

[63] P. Srinivasan, Q. Zhu, and J. Shi, "Many-to-one contour matching for describing and discriminating object shape.," in *CVPR*, pp. 1673–1680, 2010. (Cited on pages 9, 29 and 34.)

[64] J. J. Lim, C. L. Zitnick, and P. Dollár, "Sketch tokens: A learned mid-level representation for contour and object detection," in *CVPR*, pp. 3158–3165, 2013. (Cited on page 9.)

[65] Y.-H. Tsai, O. C. Hamsici, and M.-H. Yang, "Adaptive region pooling for object detection," in *CVPR*, pp. 731–739, 2015. (Cited on page 10.)

[66] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *CVPR*, pp. 447–456, 2015. (Cited on pages 10 and 82.)

[67] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *IJCV*, vol. 105, no. 3, pp. 222–245, 2013. (Cited on page 10.)

[68] R. Gokberk Cinbis, J. Verbeek, and C. Schmid, "Segmentation driven object detection with fisher vectors," in *ICCV*, pp. 2968–2975, 2013. (Cited on pages 10 and 11.)

[69] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *IJCV*, vol. 42, no. 3, pp. 145–175, 2001. (Cited on page 10.)

[70] C. Käding, A. Freytag, E. Rodner, P. Bodesheim, and J. Denzler, "Active learning and discovery of object categories in the presence of unnameable instances," in *CVPR*, pp. 4343–4352, 2015. (Cited on page 10.)

[71] S. Singh, D. Hoiem, and D. Forsyth, "Learning a sequential search for landmarks," in *CVPR*, pp. 3422–3430, 2015. (Cited on page 10.)

[72] P. Siva, C. Russell, T. Xiang, and L. Agapito, "Looking beyond the image: Unsupervised learning for object saliency and detection," in *CVPR*, 2013. (Cited on page 10.)

[73] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?," in *CVPR*, pp. 73–80, 2010. (Cited on pages 10 and 63.)

[74] W.-C. Tu, S. He, Q. Yang, and S.-Y. Chien, "Real-time salient object detection with a minimum spanning tree," in *CVPR*, 2016. (Cited on page 10.)

[75] P. Krähenbühl and V. Koltun, "Learning to propose objects," in *CVPR*, pp. 1574–1582, 2015. (Cited on page 10.)

[76] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *IJCV*, vol. 104, no. 2, pp. 154–171, 2013. (Cited on page 10.)

[77] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *CVPR*, pp. 2147–2154, 2014. (Cited on page 10.)

[78] D. Novotny and J. Matas, "Cascaded sparse spatial bins for efficient and effective generic object detection," in *ICCV*, pp. 1152–1160, 2015. (Cited on page 10.)

[79] A. Ghodrati, A. Diba, M. Pedersoli, T. Tuytelaars, and L. Van Gool, "Deepproposal: Hunting objects by cascading deep convolutional layers," in *ICCV*, pp. 2578–2586, 2015. (Cited on page 10.)

[80] J. A. Rodriguez Serrano and D. Larlus, "Predicting an object location using a global image representation," in *ICCV*, pp. 1729–1736, 2013. (Cited on page 10.)

[81] M. Najibi, M. Rastegari, and L. S. Davis, "G-cnn: An iterative grid based object detector," in *CVPR*, 2016. (Cited on page 10.)

[82] T. Kong, A. Yao, Y. Chen, and F. Sun, "Hypernet: Towards accurate region proposal generation and joint object detection," in *CVPR*, 2016. (Cited on pages 10, 39 and 90.)

[83] H. Zhu, S. Lu, J. Cai, and Q. Lee, "Diagnosing state-of-the-art object proposal methods," *BMVC*, 2015. (Cited on page 10.)

[84] S. Liu, C. Lu, and J. Jia, "Box aggregation for proposal decimation: Last mile of object detection," in *ICCV*, pp. 2569–2577, 2015. (Cited on page 10.)

[85] R. Rothe, M. Guillaumin, and L. Van Gool, "Non-maximum suppression for object detection by passing messages between windows," in *ACCV*, pp. 290–306, 2014. (Cited on page 10.)

[86] C. Desai, D. Ramanan, and C. Fowlkes, "Discriminative models for static human-object interactions," in *CVPR-Workshops*, pp. 9–16, 2010. (Cited on pages 10 and 40.)

[87] D. Mrowca, M. Rohrbach, J. Hoffman, R. Hu, K. Saenko, and T. Darrell, "Spatial semantic regularisation for large scale object detection," in *ICCV*, pp. 2003–2011, 2015. (Cited on page 10.)

[88] C. Desai, D. Ramanan, and C. C. Fowlkes, "Discriminative models for multi-class object layout," *IJCV*, vol. 95, no. 1, pp. 1–12, 2011. (Cited on pages 10 and 11.)

[89] D. Yoo, S. Park, J.-Y. Lee, A. S. Paek, and I. So Kweon, "Attentionnet: Aggregating weak directions for accurate object detection," in *ICCV*, pp. 2659–2667, 2015. (Cited on page 10.)

[90] J. C. Caicedo and S. Lazebnik, "Active object localization with deep reinforcement learning," in *ICCV*, pp. 2488–2496, 2015. (Cited on page 10.)

[91] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert, "An empirical study of context in object detection," in *CVPR*, pp. 1271–1278, 2009. (Cited on page 10.)

[92] A. Vezhnevets and V. Ferrari, "Object localization in imagenet by looking out of the window," in *BMVC*, 2015. (Cited on page 10.)

[93] C. Sun, M. Paluri, R. Collobert, R. Nevatia, and L. Bourdev, "Pronet: Learning to propose object-specific boxes for cascaded neural networks," in *CVPR*, 2016. (Cited on pages 11 and 12.)

[94] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *CVPR*, pp. 891–898, 2014. (Cited on page 11.)

[95] G. Heitz and D. Koller, "Learning spatial context: Using stuff to find things," in *ECCV*, pp. 30–43, 2008. (Cited on page 11.)

[96] A. Torralba, "Contextual priming for object detection," *IJCV*, vol. 53, no. 2, pp. 169–191, 2003. (Cited on page 11.)

[97] S. Karayev, M. Fritz, and T. Darrell, "Anytime recognition of objects and scenes," in *CVPR*, pp. 572–579, 2014. (Cited on page 11.)

[98] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie, "Objects in context," in *ICCV*, pp. 1–8, 2007. (Cited on page 11.)

[99] C. Li, D. Parikh, and T. Chen, "Extracting adaptive contextual cues from unlabeled regions," in *ICCV*, pp. 511–518, 2011. (Cited on page 11.)

[100] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool, "Real-time facial feature detection using conditional regression forests," in *CVPR*, pp. 2578–2585, 2012. (Cited on page 11.)

[101] M. Dantone, J. Gall, C. Leistner, and L. Van Gool, "Human pose estimation using body parts dependent joint regressors," in *CVPR*, pp. 3041–3048, 2013. (Cited on pages 11 and 46.)

[102] A. Gonzalez-Garcia, A. Vezhnevets, and V. Ferrari, "An active search strategy for efficient object class detection," in *CVPR*, pp. 3022–3031, 2015. (Cited on page 11.)

[103] D. Modolo, A. Vezhnevets, and V. Ferrari, "Context forest for efficient object detection with large mixture models," in *BMVC*, 2015. (Cited on page 11.)

[104] V. K. Nagaraja, V. I. Morariu, and L. S. Davis, "Searching for objects using structure in indoor scenes," in *BMVC*, 2015. (Cited on page 11.)

[105] L. Karlinsky, M. Dinerstein, D. Harari, and S. Ullman, "The chains model for detecting parts by their context," in *CVPR*, pp. 25–32, 2010. (Cited on page 11.)

[106] A. Kuznetsova, S. J. Hwang, B. Rosenhahn, and L. Sigal, "Expanding object detector's horizon: incremental learning framework for object detection in videos," in *CVPR*, pp. 28–36, 2015. (Cited on page 11.)

[107] Y. Yang, G. Shu, and M. Shah, "Semi-supervised learning of feature hierarchies for object detection in a video," in *CVPR*, pp. 1650–1657, 2013. (Cited on pages 11 and 12.)

[108] P. Sharma and R. Nevatia, "Efficient detector adaptation for object detection in a video," in *CVPR*, pp. 3254–3261, 2013. (Cited on page 11.)

[109] C. Leistner, M. Godec, S. Schulter, A. Saffari, M. Werlberger, and H. Bischof, "Improving classifiers with unlabeled weakly-related videos.," in *CVPR*, pp. 2753–2760, 2011. (Cited on pages 11, 52 and 53.)

[110] K. Kang, W. Ouyang, H. Li, and X. Wang, "Object detection from video tubelets with convolutional neural networks," in *CVPR*, 2016. (Cited on page 11.)

[111] S. Schulter, C. Leistner, P. M. Roth, and H. Bischof, "Unsupervised object discovery and segmentation in videos," in *BMVC*, pp. 1–12, 2013. (Cited on pages 11 and 53.)

[112] S. Kwak, M. Cho, I. Laptev, J. Ponce, and C. Schmid, "Unsupervised object discovery and tracking in video collections," in *ICCV*, pp. 3173–3181, 2015. (Cited on page 11.)

[113] A. Joulin, K. Tang, and L. Fei-Fei, "Efficient image and video co-localization with frank-wolfe algorithm," in *ECCV*, pp. 253–268, 2014. (Cited on page 11.)

[114] L. Wang, G. Hua, R. Sukthankar, J. Xue, and N. Zheng, "Video object discovery and co-segmentation with extremely weak supervision," in *ECCV*, pp. 640–655, 2014. (Cited on page 11.)

[115] D. Zhang, O. Javed, and M. Shah, "Video object co-segmentation by regulated maximum weight cliques," in *ECCV*, pp. 551–566, 2014. (Cited on page 11.)

[116] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari, "Learning object class detectors from weakly annotated video," in *CVPR*, pp. 3282–3289, 2012. (Cited on pages 11, 52, 53, 55, 56, 57, 60, 63, 64, 65, 66 and 67.)

[117] J. Yan, Y. Yu, X. Zhu, Z. Lei, and S. Z. Li, "Object detection by labeling superpixels," in *CVPR*, pp. 5107–5116, 2015. (Cited on page 11.)

[118] A. Angelova and S. Zhu, "Efficient object detection and segmentation for fine-grained recognition," in *CVPR*, pp. 811–818, 2013. (Cited on page 11.)

[119] B. Kim, S. Xu, and S. Savarese, "Accurate localization of 3d objects from rgb-d data using segmentation hypotheses.," in *CVPR*, pp. 3182–3189, 2013. (Cited on pages 11 and 36.)

[120] X. Chen, A. Shrivastava, and A. Gupta, "Enriching visual knowledge bases via object discovery and segmentation," in *CVPR*, pp. 2027–2034, 2014. (Cited on page 11.)

[121] Y.-H. Tsai, J. Yang, and M.-H. Yang, "Decomposed learning for joint object segmentation and categorization," in *BMVC*, 2013. (Cited on page 11.)

[122] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *ECCV*, pp. 297–312, 2014. (Cited on page 11.)

[123] J. Dong, Q. Chen, S. Yan, and A. Yuille, "Towards unified object detection and semantic segmentation," in *ECCV*, pp. 299–314, 2014. (Cited on page 11.)

[124] S. Gidaris and N. Komodakis, "Object detection via a multi-region and semantic segmentation-aware cnn model," in *ICCV*, pp. 1134–1142, 2015. (Cited on page 11.)

[125] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu, "Unsupervised joint object discovery and segmentation in internet images," in *CVPR*, pp. 1939–1946, 2013. (Cited on pages 11, 12, 52 and 53.)

[126] A. Tejani, D. Tang, R. Kouskouridas, and T.-K. Kim, "Latent-class hough forests for 3d object detection and pose estimation," in *ECCV*, pp. 462–477, 2014. (Cited on page 11.)

[127] M. Fenzi and J. Ostermann, "Embedding geometry in generative models for pose estimation of object categories.," in *BMVC*, vol. 1, p. 3, 2014. (Cited on page 11.)

[128] P. Wei, Y. Zhao, N. Zheng, and S.-C. Zhu, "Modeling 4d human-object interactions for event and object recognition," in *ICCV*, pp. 3272–3279, 2013. (Cited on page 11.)

[129] H. Bilen, V. P. Namboodiri, and L. J. Van Gool, "Object and action classification with latent variables.," in *BMVC*, vol. 2, p. 3, 2011. (Cited on page 11.)

[130] C. Desai and D. Ramanan, "Predicting functional regions on objects," in *CVPR-Workshops*, pp. 968–975, 2013. (Cited on pages 12 and 75.)

[131] A. Myers, C. L. Teo, C. Fermuller, and Y. Aloimonos, "Affordance detection of tool parts from geometric features," in *ICRA*, pp. 1374–1381, 2015. (Cited on pages 12, 74, 75, 76, 78, 81, 82, 84 and 86.)

[132] D. P. Papadopoulos, J. R. R. Uijlings, F. Keller, and V. Ferrari, "We don't need no bounding-boxes: Training object class detectors using only human verification," in *CVPR*, 2016. (Cited on page 12.)

[133] S. Ebert, M. Fritz, and B. Schiele, "Ralf: A reinforced active learning formulation for object class recognition," in *CVPR*, pp. 3626–3633, 2012. (Cited on page 12.)

[134] A. Yao, J. Gall, C. Leistner, and L. Van Gool, "Interactive object detection," in *CVPR*, pp. 3242–3249, 2012. (Cited on page 12.)

[135] S. Ebert, D. Larlus, and B. Schiele, "Extracting structures in image collections for object recognition," in *ECCV*, pp. 720–733, 2010. (Cited on page 12.)

[136] I. Misra, A. Shrivastava, and M. Hebert, "Watch and learn: Semi-supervised learning for object detectors from video," in *CVPR*, pp. 3593–3602, 2015. (Cited on pages 12 and 69.)

[137] X. Liang, S. Liu, Y. Wei, L. Liu, L. Lin, and S. Yan, "Towards computational baby learning: A weakly-supervised approach for object detection," in *ICCV*, pp. 999–1007, 2015. (Cited on page 12.)

[138] K. Kumar Singh, F. Xiao, and Y. Jae Lee, "Track and transfer: Watching videos to simulate strong human supervision for weakly-supervised object detection," in *CVPR*, 2016. (Cited on page 12.)

[139] X. Peng, B. Sun, K. Ali, and K. Saenko, "Learning deep object detectors from 3d models," in *ICCV*, pp. 1278–1286, 2015. (Cited on page 12.)

[140] B. Sun and K. Saenko, "From virtual to reality: Fast adaptation of virtual object detectors to real domains," in *BMVC*, 2014. (Cited on page 12.)

[141] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *CVPR*, 2016. (Cited on page 12.)

[142] H. Bilen and A. Vedaldi, "Weakly supervised deep detection networks," in *CVPR*, 2016. (Cited on page 12.)

[143] D. Li, J.-B. Huang, Y. Li, S. Wang, and M.-H. Yang, "Weakly supervised object localization with progressive domain adaptation," in *CVPR*, 2016. (Cited on page 12.)

[144] H. Bilen, M. Pedersoli, and T. Tuytelaars, "Weakly supervised object detection with posterior regularization," in *BMVC*, 2014. (Cited on pages 12, 52 and 53.)

[145] H. Bilen, M. Pedersoli, and T. Tuytelaars, "Weakly supervised object detection with convex clustering," in *CVPR*, pp. 1081–1089, 2015. (Cited on pages 12, 52 and 53.)

[146] R. G. Cinbis, J. Verbeek, and C. Schmid, "Multi-fold mil training for weakly supervised object localization," in *CVPR*, pp. 2409–2416, 2014. (Cited on pages 12, 52 and 53.)

[147] K. Ali and K. Saenko, "Confidence-rated multiple instance boosting for object detection," in *CVPR*, pp. 2433–2440, 2014. (Cited on page 12.)

[148] X. Guo, D. Liu, B. Jou, M. Zhu, A. Cai, and S.-F. Chang, "Robust object co-detection," in *CVPR*, pp. 3206–3213, 2013. (Cited on page 12.)

[149] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Is object localization for free?–weakly-supervised learning with convolutional neural networks," in *CVPR*, pp. 685–694, 2015. (Cited on pages 12 and 53.)

[150] A. Kolesnikov and C. H. Lampert, "Improving weakly-supervised object localization by micro-annotation," *ArXiv*, 2016. (Cited on page 12.)

[151] T. Tuytelaars, C. H. Lampert, M. B. Blaschko, and W. Buntine, "Unsupervised object discovery: A comparison," *IJCV*, vol. 88, no. 2, pp. 284–302, 2010. (Cited on pages 12 and 53.)

[152] Y.-X. Wang and M. Hebert, "Model recommendation: Generating object detectors from few samples," in *CVPR*, pp. 1619–1628, 2015. (Cited on pages 12 and 53.)

[153] M. Cho, S. Kwak, C. Schmid, and J. Ponce, "Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals," in *CVPR*, pp. 1201–1210, 2015. (Cited on page 12.)

[154] J. Gall, A. Fossati, and L. Van Gool, "Functional categorization of objects using real-time markerless motion capture," in *CVPR*, pp. 1969–1976, 2011. (Cited on pages 12, 40, 44, 52 and 53.)

[155] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *PAMI*, vol. 28, no. 10, pp. 1568–1583, 2006. (Cited on pages 13 and 20.)

[156] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *IJCV*, vol. 59, no. 2, pp. 167–181, 2004. (Cited on pages 13, 22 and 55.)

[157] T. Brox and J. Malik, "Large displacement optical flow: descriptor matching in variational motion estimation," *PAMI*, vol. 33, no. 3, pp. 500–513, 2011. (Cited on pages 13, 24, 25 and 55.)

[158] A. Srikantha and J. Gall, "Hough-based object detection with grouped features," in *ICIP*, pp. 1653–1657, 2014. (Cited on page 13.)

[159] A. Srikantha and J. Gall, "Human pose as context for object detection," in *BMVC*, 2015. (Cited on page 14.)

[160] A. Srikantha and J. Gall, "Discovering object classes from activities," in *ECCV*, pp. 415–430, 2014. (Cited on pages 14, 45, 53 and 56.)

[161] A. Srikantha and J. Gall, "Weakly supervised learning of affordances," in *Arxiv*, 2016. (Cited on page 14.)

[162] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *IJCV*, vol. 70, no. 1, pp. 41–54, 2006. (Cited on page 21.)

[163] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *ECCV*, pp. 25–36, 2004. (Cited on page 24.)

[164] A. Opelt, A. Pinz, and A. Zisserman, "Learning an alphabet of shape and appearance for multi-class object detection," *IJCV*, vol. 80, no. 1, pp. 16–44, 2008. (Cited on page 27.)

[165] J. Shotton, A. Blake, and R. Cipolla, "Multiscale categorical object recognition using contour fragments," *PAMI*, vol. 30, no. 7, pp. 1270–1281, 2008. (Cited on page 27.)

[166] M. Sun, G. Bradski, B.-X. Xu, and S. Savarese, "Depth-encoded hough voting for joint object detection and shape recovery," in *ECCV*, pp. 658–671, 2010. (Cited on page 27.)

[167] Y. Zhang and T. Chen, "Implicit shape kernel for discriminative learning of the hough transform detector," in *BMVC*, pp. 105.1–105.11, 2010. (Cited on page 27.)

[168] B. Ommer and J. Malik, "Multiscale object detection by clustering lines.," in *ICCV*, pp. 484–491, 2009. (Cited on pages 27, 28 and 34.)

[169] R. Okada, "Discriminative generalized hough transform for object dectection," in *ICCV*, pp. 2000–2005, 2009. (Cited on page 27.)

[170] A. Criminisi and J. S. (Editors), *Decision Forests for Computer Vision and Medical Image Analysis*. Springer, 2013. (Cited on page 27.)

[171] L. Breiman, "Random forests," in *Machine Learning*, pp. 5–32, 2001. (Cited on pages 27 and 31.)

[172] N. Razavi, N. Alvar, J. Gall, and L. Van Gool, "Sparsity potentials for detecting objects with the hough transform," in *BMVC*, pp. 11.1–11.10, 2012. (Cited on page 27.)

[173] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation.," in *CVPR*, pp. 1–8, 2008. (Cited on pages 28, 30 and 31.)

[174] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration.," *CVIU*, vol. 89, no. 2–3, pp. 114–141, 2003. (Cited on page 28.)

[175] T. Ma and L. Latecki, "From partial shape matching through local deformation to robust global shape similarity for object detection.," in *CVPR*, pp. 1441–1448, 2011. (Cited on pages 28 and 34.)

[176] P. Yarlagadda and B. Ommer, "From meaningful contours to discriminative object shape.," in *ECCV*, pp. 776–779, 2012. (Cited on pages 28, 34 and 36.)

[177] A. Toshev, B. Taskar, and K. Daniilidis, "Object detection via boundary structure segmentation.," in *CVPR*, pp. 950–957, 2010. (Cited on pages 29, 34 and 36.)

[178] P. Dollar, S. Belongie, and P. Perona, "The fastest pedestrian detector in the west.," in *BMVC*, vol. 2, p. 7, 2010. (Cited on page 32.)

[179] J. Shotton, A. Blake, and R. Cipolla, "Efficiently combining contour and texture cues for object recognition.," in *BMVC*, 2008. (Cited on page 36.)

[180] H. S. Koppula, R. Gupta, and A. Saxena, "Learning human activities and object affordances from rgb-d videos," *IJRR*, vol. 32, no. 8, pp. 951–970, 2013. (Cited on pages 39, 40, 44, 52, 53, 54, 62, 73 and 91.)

[181] Y. Jiang and A. Saxena, "Hallucinating humans for learning robotic placement of objects," in *Experimental Robotics*, pp. 921–937, 2013. (Cited on pages 39 and 40.)

[182] A. Gupta and L. S. Davis, "Objects in action: An approach for combining action understanding and object perception," in *CVPR*, pp. 1–8, 2007. (Cited on pages 39, 40 and 53.)

[183] B. Yao and L. Fei-Fei, "Modeling mutual context of object and human pose in human-object interaction activities," in *CVPR*, pp. 17–24, 2010. (Cited on pages 39 and 40.)

[184] M. Sun and S. Savarese, "Articulated part-based model for joint object detection and pose estimation," in *ICCV*, pp. 723–730, 2011. (Cited on pages 39, 40 and 52.)

[185] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele, "A database for fine grained activity detection of cooking activities," in *CVPR*, pp. 1194–1201, 2012. (Cited on pages 40, 44, 45, 46, 52 and 62.)

[186] A. Prest, C. Schmid, and V. Ferrari, "Weakly supervised learning of interactions between humans and objects," *PAMI*, pp. 601–614, 2012. (Cited on page 40.)

[187] B. Yao, J. Ma, and L. Fei-Fei, "Discovering object functionality," in *ICCV*, pp. 2512–2519, 2013. (Cited on page 40.)

[188] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *IJCV*, pp. 55–79, 2005. (Cited on pages 40 and 41.)

[189] A. Yao, J. Gall, and L. Van Gool, "Coupled action recognition and pose estimation from multiple views," *IJCV*, pp. 16–37, 2012. (Cited on page 43.)

[190] P. Getreuer, "Automatic Color Enhancement (ACE) and its fast implementation," *Image Processing On Line*, pp. 266–277, 2012. (Cited on page 47.)

[191] M. B. Blaschko, A. Vedaldi, and A. Zisserman, "Simultaneous object detection and ranking with weak supervision," in *NIPS*, pp. 235–243, 2010. (Cited on pages 52 and 53.)

[192] O. Chum and A. Zisserman, "An exemplar model for learning object classes," in *CVPR*, pp. 1–8, 2007. (Cited on pages 52 and 53.)

[193] Y. J. Lee and K. Grauman, "Learning the easy things first: Self-paced visual category discovery.," in *CVPR*, pp. 1721–1728, 2011. (Cited on pages 52 and 53.)

[194] J. M. Winn and N. Jojic, "Locus: Learning object classes with unsupervised segmentation.," in *ICCV*, pp. 756–763, 2005. (Cited on pages 52 and 53.)

[195] D. Ramanan, D. A. Forsyth, and K. Barnard, "Building models of animals from video.," *PAMI*, vol. 28, no. 8, pp. 1319–1334, 2006. (Cited on pages 52 and 53.)

[196] E. Hsiao and M. Hebert, "Occlusion reasoning for object detectionunder arbitrary viewpoint," *PAMI*, vol. 36, no. 9, pp. 1803–1815, 2014. (Cited on page 52.)

[197] T. Gao, B. Packer, and D. Koller, "A segmentation-aware object detection model with occlusion handling," in *CVPR*, pp. 1361–1368, 2011. (Cited on page 52.)

[198] P. Yadollahpour, D. Batra, and G. Shakhnarovich, "Discriminative re-ranking of diverse segmentations," in *CVPR*, pp. 1923–1930, 2013. (Cited on page 53.)

[199] Y. Yang and D. Ramanan, "Articulated pose estimation with flexible mixtures-of-parts," in *CVPR*, pp. 1385–1392, 2011. (Cited on page 53.)

[200] T. Brox, L. Bourdev, S. Maji, and J. Malik, "Object segmentation by alignment of poselet activations to image contours," in *CVPR*, pp. 2225–2232, 2011. (Cited on page 53.)

[201] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *CVPR*, pp. 2411–2418, 2013. (Cited on page 53.)

[202] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *CVPR*, pp. 1201–1208, 2011. (Cited on page 53.)

[203] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Online multiperson tracking-by-detection from a single, uncalibrated camera," *PAMI*, vol. 33, no. 9, pp. 1820–1833, 2011. (Cited on page 53.)

[204] Y. J. Lee, J. Kim, and K. Grauman, "Key-segments for video object segmentation," in *ICCV*, pp. 1995–2002, 2011. (Cited on page 53.)

[205] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid, "Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation," in *ICCV*, pp. 309–316, 2009. (Cited on page 53.)

[206] A. Gaidon, M. Marszalek, and C. Schmid, "Mining visual actions from movies," in *BMVC*, pp. 125–1, 2009. (Cited on page 53.)

[207] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," in *CVPR*, pp. 2929–2936, 2009. (Cited on page 53.)

[208] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. Efros, "What makes paris look like paris?," *ACM Transactions on Graphics*, vol. 31, no. 4, 2012. (Cited on page 53.)

[209] V. Ordonez, G. Kulkarni, and T. L. Berg, "Im2text: Describing images using 1 million captioned photographs," in *NIPS*, pp. 1143–1151, 2011. (Cited on page 53.)

[210] B. Ommer, T. Mader, and J. Buhmann, "Seeing the Objects Behind the Dots: Recognition in Videos from a Moving Camera," *IJCV*, vol. 83, pp. 57–71, 2009. (Cited on page 53.)

[211] P. Tokmakov, K. Alahari, and C. Schmid, "Weakly-supervised semantic segmentation using motion cues," *ArXiv*, 2016. (Cited on page 53.)

[212] V. Delaitre, D. Fouhey, I. Laptev, J. Sivic, A. Gupta, and A. Efros, "Scene semantics from long-term observation of people," in *ECCV*, pp. 284–298, 2012. (Cited on pages 53 and 54.)

[213] D. F. Fouhey, V. Delaitre, A. Gupta, A. A. Efros, I. Laptev, and J. Sivic, "People watching: Human actions as a cue for single-view geometry," in *ECCV*, pp. 259–274, 2012. (Cited on pages 53 and 54.)

[214] H. Grabner, J. Gall, and L. Van Gool, "What makes a chair a chair?," in *CVPR*, pp. 1529–1536, 2011. (Cited on pages 53, 54 and 75.)

[215] A. Gupta, S. Satkin, A. A. Efros, and M. Hebert, "From 3d scene geometry to human workspace," in *CVPR*, pp. 1961–1968, 2011. (Cited on pages 53 and 54.)

[216] Y. Jiang, H. Koppula, and A. Saxena, "Hallucinated humans as the hidden context for labeling 3d scenes," in *CVPR*, pp. 2993–3000, 2013. (Cited on pages 53, 54 and 75.)

[217] H. Kjellström, J. Romero, and D. Kragić, "Visual object-action recognition: Inferring object affordances from human demonstration," *CVIU*, vol. 115, no. 1, pp. 81–90, 2011. (Cited on pages 53 and 75.)

[218] P. Peursum, G. West, and S. Venkatesh, "Combining image regions and human activity for indirect object recognition in indoor wide-angle views," in *ICCV*, pp. 82–89, 2005. (Cited on page 53.)

[219] A. Pieropan, C. H. Ek, and H. Kjellstrom, "Functional object descriptors for human activity modeling," in *ICRA*, pp. 1282–1289, 2013. (Cited on pages 53 and 54.)

[220] M. W. Turek, A. Hoogs, and R. Collins, "Unsupervised learning of functional categories in video scenes," in *ECCV*, pp. 664–677, 2010. (Cited on page 53.)

[221] R. Filipovych and E. Ribeiro, "Recognizing primitive interactions by exploring actor-object states," in *CVPR*, pp. 1–7, 2008. (Cited on page 53.)

[222] D. Moore, I. Essa, and M. Hayes, "Exploiting human actions and object context for recognition tasks," in *ICCV*, pp. 80–86, 1999. (Cited on page 53.)

[223] M. Jain, J. C. van Gemert, and C. G. Snoek, "What do 15,000 object categories tell us about classifying and localizing actions?," in *CVPR*, pp. 46–55, 2015. (Cited on pages 54 and 90.)

[224] M. Jain, J. C. van Gemert, T. Mensink, and C. G. Snoek, "Objects2action: Classifying and localizing actions without any video example," in *ICCV*, pp. 4588–4596, 2015. (Cited on pages 54 and 90.)

[225] A. Fathi, X. Ren, and J. Rehg, "Learning to recognize objects in egocentric activities," in *CVPR*, pp. 3281–3288, 2011. (Cited on page 54.)

[226] T. Brox and J. Malik, "Object segmentation by long term analysis of point trajectories," in *ECCV*, pp. 282–295, 2010. (Cited on page 55.)

[227] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *PAMI*, vol. 24, no. 5, pp. 603–619, 2002. (Cited on page 55.)

[228] T. Deselaers, B. Alexe, and V. Ferrari, "Localizing objects while learning their appearance.," in *ECCV*, vol. 6314, pp. 452–466, 2010. (Cited on page 56.)

[229] M. Jones and J. Rehg., "Statistical color models with application to skin detection," *IJCV*, vol. 46, no. 1, pp. 81–96, 2002. (Cited on page 59.)

[230] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *ACM Int. Conf. on Image and Video Retrieval*, pp. 401–408, 2007. (Cited on page 60.)

[231] A. Fossati, J. Gall, H. Grabner, X. Ren, and K. Konolige, eds., *Consumer Depth Cameras for Computer Vision*, ch. Human Body Analysis. Springer, 2013. (Cited on pages 62, 63 and 69.)

[232] S. Manen, M. Guillaumin, and L. Van Gool, "Prime object proposals with randomized prim's algorithm," in *ICCV*, pp. 2536–2543, 2013. (Cited on pages 63 and 64.)

[233] P. Siva and T. Xiang, "Weakly supervised object detector learning with model drift detection," in *ICCV*, pp. 343–350, 2011. (Cited on page 69.)

[234] D. Parikh and K. Grauman, "Relative attributes," in *ICCV*, pp. 503–510, 2011. (Cited on pages 73 and 75.)

[235] J. Liu, B. Kuipers, and S. Savarese, "Recognizing human actions by attributes," in *CVPR*, pp. 3337–3344, 2011. (Cited on page 73.)

[236] G. Patterson and J. Hays, "Sun attribute database: Discovering, annotating, and recognizing scene attributes," in *CVPR*, pp. 2751–2758, 2012. (Cited on page 73.)

[237] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *ICCV*, pp. 2722–2730, 2015. (Cited on page 73.)

[238] D. Katz, A. Venkatraman, M. Kazemi, J. A. Bagnell, and A. Stentz, "Perceiving, learning, and exploiting object affordances for autonomous pile manipulation," *Autonomous Robots*, vol. 37, no. 4, pp. 369–382, 2014. (Cited on pages 74, 75 and 78.)

[239] D. I. Kim and G. Sukhatme, "Semantic labeling of 3d point clouds with object affordance for robot manipulation," in *ICRA*, pp. 5578–5584, 2014. (Cited on pages 74, 75 and 78.)

[240] T. Hermans, J. M. Rehg, and A. Bobick, "Affordance prediction via learned object attributes," in *ICRA-Workshops*, 2011. (Cited on pages 74, 75 and 78.)

[241] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille, "Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation," in *ICCV*, pp. 1742–1750, 2015. (Cited on pages 74, 76, 79, 81, 82, 83, 84 and 86.)

[242] S. Bell, P. Upchurch, N. Snavely, and K. Bala, "Material recognition in the wild with the materials in context database," in *CVPR*, pp. 3479–3487, 2015. (Cited on pages 74 and 78.)

[243] F. S. Khan, R. M. Anwer, J. van de Weijer, A. D. Bagdanov, M. Vanrell, and A. M. Lopez, "Color attributes for object detection," in *CVPR*, pp. 3306–3313, 2012. (Cited on page 75.)

[244] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *CVPR*, pp. 1778–1785, 2009. (Cited on page 75.)

[245] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *CVPR*, pp. 951–958, 2009. (Cited on page 75.)

[246] V. Ferrari and A. Zisserman, "Learning visual attributes," in *NIPS*, pp. 433–440, 2007. (Cited on page 75.)

[247] Y. Zhu, A. Fathi, and L. Fei-Fei, "Reasoning about object affordances in a knowledge base representation," in *ECCV*, pp. 408–424, 2014. (Cited on page 75.)

[248] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele, "Evaluation of output embeddings for fine-grained image classification," in *CVPR*, pp. 2927–2936, 2015. (Cited on page 75.)

[249] J. Deng, J. Krause, A. C. Berg, and L. Fei-Fei, "Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition," in *CVPR*, pp. 3450–3457, 2012. (Cited on page 75.)

[250] Y.-W. Chao, Z. Wang, R. Mihalcea, and J. Deng, "Mining semantic affordances of visual object categories," in *CVPR*, pp. 4259–4267, 2015. (Cited on page 75.)

[251] C. Castellini, T. Tommasi, N. Noceti, F. Odone, and B. Caputo, "Using object affordances to improve object recognition," *Autonomous Mental Development*, vol. 3, no. 3, pp. 207–215, 2011. (Cited on page 75.)

[252] Y. Zhu, Y. Zhao, and S. Chun Zhu, "Understanding tools: Task-oriented object modeling, learning and recognition," in *CVPR*, pp. 2855–2864, 2015. (Cited on page 75.)

[253] H. S. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response," *PAMI*, vol. 38, no. 1, pp. 14–29, 2016. (Cited on page 75.)

[254] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *IJRR*, vol. 34, no. 4-5, pp. 705–724, 2015. (Cited on pages 75 and 78.)

[255] H. O. Song, M. Fritz, D. Goehring, and T. Darrell, "Learning to detect visual grasp affordance," 2016. (Cited on pages 75 and 78.)

[256] H. S. Koppula and A. Saxena, "Physically grounded spatio-temporal object affordances," in *ECCV*, pp. 831–847, 2014. (Cited on pages 75 and 77.)

[257] A. Vezhnevets and J. M. Buhmann, "Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning," in *CVPR*, pp. 3249–3256, 2010. (Cited on page 76.)

[258] A. Vezhnevets, V. Ferrari, and J. M. Buhmann, "Weakly supervised semantic segmentation with a multi-image model," in *ICCV*, pp. 643–650, 2011. (Cited on page 76.)

[259] A. Vezhnevets, V. Ferrari, and J. M. Buhmann, "Weakly supervised structured output learning for semantic segmentation," in *CVPR*, pp. 845–852, 2012. (Cited on page 76.)

[260] J. Xu, A. Schwing, and R. Urtasun, "Tell me what you see and i will show you where it is," in *CVPR*, pp. 3190–3197, 2014. (Cited on page 76.)

[261] W. Zhang, S. Zeng, D. Wang, and X. Xue, "Weakly supervised semantic segmentation for social images," in *CVPR*, pp. 2718–2726, 2015. (Cited on pages 76 and 90.)

[262] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *ICLR*, 2015. (Cited on pages 76, 78, 79, 81, 84, 85 and 86.)

[263] D. Pathak, P. Krahenbuhl, and T. Darrell, "Constrained convolutional neural networks for weakly supervised segmentation," in *ICCV*, pp. 1796–1804, 2015. (Cited on pages 76, 81, 82, 83 and 84.)

[264] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ArXiv*, 2014. (Cited on pages 80, 81 and 82.)

[265] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, and B. Schiele, "Deepcut: Joint subset partition and labeling for multi person pose estimation," *CVPR*, 2016. (Cited on page 90.)

[266] M. Rohrbach, A. Rohrbach, M. Regneri, S. Amin, M. Andriluka, M. Pinkal, and B. Schiele, "Recognizing fine-grained and composite activities using hand-centric features and script data," *IJCV*, pp. 1–28. (Cited on page 90.)

[267] H. Yasin, U. Iqbal, B. Krüger, A. Weber, and J. Gall, "A dual-source approach for 3d pose estimation from a single image," CVPR, 2016. (Cited on page 90.)

[268] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *CVPR*, 2016. (Cited on page 90.)

[269] U. Iqbal, M. Garbade, and J. Gall, "Pose for action-action for pose," *ArXiv*, 2016. (Cited on page 90.)

[270] B. Xiaohan Nie, C. Xiong, and S.-C. Zhu, "Joint action recognition and pose estimation from video," in *CVPR*, pp. 1293–1301, 2015. (Cited on page 90.)

[271] C. J. Paulus, N. Haouchine, D. Cazier, and S. Cotin, "Augmented reality during cutting and tearing of deformable objects," in *ISMAR*, pp. 54–59, 2015. (Cited on page 91.)

[272] A. Petit, V. Lippiello, and B. Siciliano, "Tracking an elastic object with an rgb-d sensor for a pizza chef robot," in *Humanoid Robots*, 2014. (Cited on page 91.)

[273] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," *PAMI*, vol. 27, no. 11, pp. 1778–1792, 2005. (Cited on page 91.)

Erklärung

Ich versichere, dass ich die von mir vorgelegte Dissertation selbständig angefertigt, die benutzten Quellen und Hilfsmittel vollständig angegeben und die Stellen der Arbeiteinschlielich Tabellen, Karten und Abbildungen –, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, in jedem Einzelfall als Entlehnung kenntlich gemacht habe; dass diese Dissertation noch keiner anderen Fakultät oder Universität zur Prüfung vorgelegen hat; dass sie noch nicht veröffentlicht worden ist sowie, da ich eine solche Veröffentlichung vor Abschluss des Promotionsverfahrens nicht vornehmen werde. Die Bestimmungen dieser Promotionsordnung sind mir bekannt. Die von mir vorgelegte Dissertation ist von Prof. Dr. Juergen Gall betreut worden.

Unterschift:

_____

Datum:

_____