Thesis for the degree
Doctor of Philosophy

עבודת גמר (תזה) לתואר
דוקטור לפילוסופיה

Submitted to the Scientific Council of the
Weizmann Institute of Science
Rehovot, Israel

מוגשת למועצה המדעית של
מכון ויצמן למדע
רחובות, ישראל

במתכונת "מאמרים"
In a "Published Papers" Format

By
Leonid Karlinsky

מאת
לאוניד קרלינסקי

'משפטים ויזואליים': לקראת זיהוי פעולות ומרכיביהן

'Visual verb phrases': towards the recognition of actions and their components

Advisor:
Prof. Shimon Ullman

מנחה:
פרופ' שמעון אולמן

April, 2010

אייר, תש"ע

Acknowledgment

To Elisha, Adam, Odelia, my parents Bella and Alexander, my grandparents Rosa, Jacob, Gera and Zalman, my sister Irena and all who helped and supported me on my way to completing this thesis. Last but not least, I would like to express my deep and sincere gratitude to my supervisor, Professor Shimon Ullman, who helped shape my scientific thinking and made this thesis possible.

Table of Contents

Introduction

Recognizing actions in static and dynamic images is a basic and important capacity in visual perception. The focus of this study is on the recognition of what might be termed 'visual verb phrases'. These are static or dynamic images which show an action that will typically be described by a short verb phrase, which specifies not only the action verb, but also additional information about the agent performing the action, the recipient, the tool etc. The input to the computation is a video or a single image, and the output is a description such as 'the man picked a glass with his right hand'. The recognition of such actions requires, in addition to naming the action, the identification and localization of different components of the action, including the actor performing the action, the actual action being performed, the recipient of the action, the relevant body parts, and often a tool being used. Figure 1 illustrates examples of the problem and some of our current results. The analysis in these cases and in our work in general combines what we call 'image interpretation' and 'event detection and analysis'. By 'interpretation' we mean the full detection and localization of the different components of the actor and the objects involved in the action, and by 'event detection and analysis' we refer to detecting and analyzing the components of the action itself using the information obtained from the interpretation.

(a)

(b) (c)

(d)

Figure 1: An example of the goal and of the results of two of the developed approaches, namely the body interpretation based approach and the chains model based approach. (a) Different examples of the 'drinking' action, notice the visual dissimilarity between different examples. (b) An example action we want to recognize and interpret. Observing several frames or even a single static frame of the sequence, the interpretation would be: 'a man picked up a glass with his right hand'. The performed action here is 'to pick up', the actor is the 'man', the recipient of the action is the 'glass' and the body part (or the tool) being used is the 'hand'. (c) The output of our body interpretation based approach (Appendix B), both the objects involved (man, glass) are interpreted (their parts marked with colors) and the action is detected (annotation above and in the ROI of the action). The approach uses our automatic body parts interpretation method (described in Appendix A). (d) Example actions automatically identified from single images using our static action recognition approach [Karlinsky et al., 2010b]. For each action, the relevant body parts (face, hand, and elbow) are detected [[Karlinsky et al., 2008b, Karlinsky et al., 2010a] followed by the action recognition using features anchored to the parts. The bar graph on the right of each image depicts the log-likelihood of the 12 considered actions. Note the subtle visual differences that are necessary to detect in order to distinguish between similar actions.

Types of actions

"Action" is one of the most complex "objects" for visual recognition. For example Figure 1a shows a large variety of images depicting the action 'drinking'. 'Drinking' can be performed by a man, a woman, or a child; one can drink with the left hand, the right hand or with both, using a cup, a glass, or a bottle, etc. Although action is an intuitive concept, which corresponds to some common dynamical occurrence the term 'action' refers to a large range of dynamic events. In this study, we typically refer to actions which are assigned verbs in the natural language. More specifically, we are

focusing on actions that can be visually identified from a “not too long” sequence of movements, visual state changes or a particular configuration of objects and their parts. We denote those actions: elementary actions. Examples of elementary actions are walking, taking, pushing, drinking, writing and the like. Examples for the non-elementary actions, which are not addressed directly in this study, would be buying, traveling, etc.

We further differentiate elementary actions between "transitive" and "intransitive" actions. Transitive actions are typically described by transitive verbs: point, touch, hold, etc.; they involve more than a single entity, usually the ‘actor’ who performs the action and the ‘recipient of the action’ – the direct object of the verb. Intransitive actions, described by intransitive verbs, such as: walking, running, jumping, involve only one entity – an agent performing the action, e.g.: a running man, jumping animal, falling stone, etc. In this study we focus mainly on the transitive actions, which are usually more difficult to model and recognize, but the tools and the methodology we develop readily apply to any type of elementary action. Moreover, intransitive actions, such as in the examples above, are characterized by movement and therefore temporal information (a short video sequence) is required in order to recognize them. In contrast, the transitive actions are more characterized by the pose of the actor, the tool she/he is holding, and the presence of the action recipient, and thus in many cases such actions can be readily identified by human observers from only a single image [Karlinsky et al., 2010b].

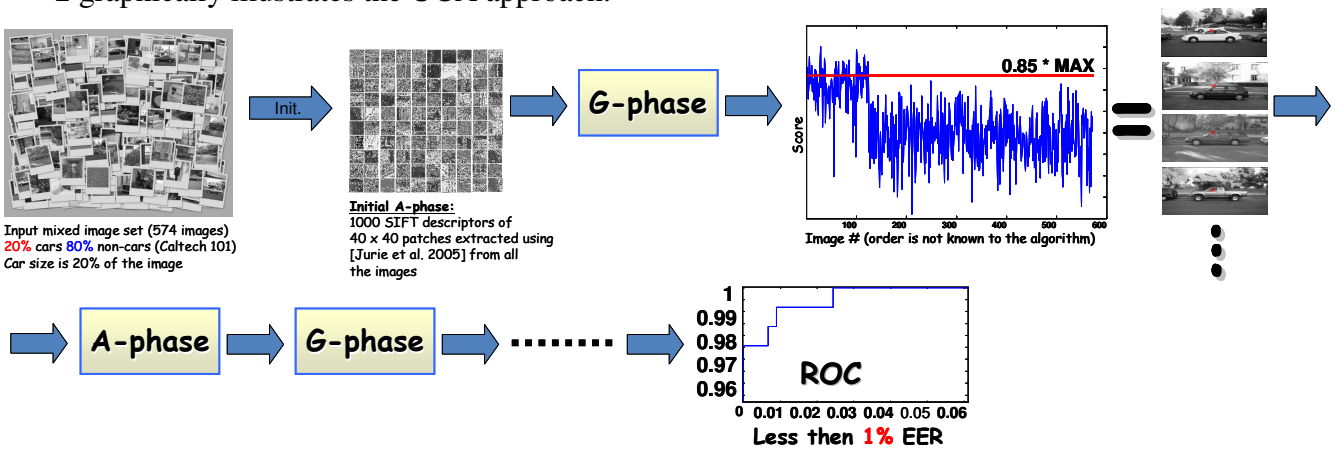
In this study we focus on action recognition based on more complete "interpretation" of objects and their parts, rather than just labeling the image as containing or not containing an action. We argue that to reliably recognize most transitive elementary actions, one has to first to detect not only the objects involved, but also their relevant parts and agent-to-object relations in space and in time. We denote by the term "image interpretation" the result of the process of object, part and relation detection. Examples of interpretation of frames of action sequences are shown in Figure 1c, 1d.

Methods for image interpretation

In the course of this study, we have therefore developed a number of useful tools for image interpretation. These tools are briefly described in the list below and are described in detail in the Papers section and in Appendix A.

1. ***Unsupervised Consistency Amplification (UCA)*** - an iterative unsupervised learning approach described in [Karlinsky et al., 2008a]. A motivating observation is that as the objects we want to classify become more complex, such as in the transition from rigid objects to articulated objects and to actions, the amount of supervision which can be obtained for these objects becomes more and more scarce, as the cost of human annotation (a.k.a. supervision) grows considerably due to the complexity increase. Moreover, even when supervision is supplied, it usually labels the entire configuration rather than the important components. For example, a usual practice is giving what is called 'weak supervision' to the visual learning system, in which the system knows that an image or a video contains an object of interest, but is left oblivious to the object location, pose or (in case of actions) temporal extent and composition. The UCA is therefore a learning methodology that can be used under unsupervised or insufficient supervision conditions. The method is capable of learning effective rigid object detectors from an unsupervised training set - unordered list of

images without object presence and location labels (meaning that the set has both images with and without objects of interest and the objects may appear anywhere on the image). The method can also be applied to multi-class or multi-subclass settings by extracting the classes of interest one by one. In terms of its applications to action recognition, UCA was shown to be capable of successful unsupervised learning of different hand poses (as sub-classes of the hand object), and also developed into a useful face detection and recognition tool [Karlinsky et al., 2008b]. We also believe that in the future UCA can also be used for unsupervised learning of objects that people interact with during actions, thus complementing the image interpretations with the ability to recognize the tools and the recipients of actions in the scene. Figure 2 graphically illustrates the UCA approach.



(a)

(b)

Figure 2: (a) Schematic diagram of the UCA algorithm. The G-phase and the A-phase refer to specific parts of the algorithm, corresponding to unsupervised star model learning (to identify most consistent image sub-configurations) and to semantic feature learning respectively. These phases are iterated to improve the final result; (b) Example results of unsupervised object and part localization on two datasets (UIUC cars, flamingo). The yellow star is the detected model center location (see paper), color coded rectangles are examples of detected object parts (for each object several out of about 150 modeled parts are shown).

2. ***Unsupervised Feature Optimization (UFO)*** – unsupervised bottom-up feature selection, parameter optimization, and parts sub-configuration learning approach

described in [Karlinsky et al., 2009]. The UFO is an unsupervised learning approach, and thus shares with UCA the applications to image interpretation in the service of action recognition. In addition, UFO is a general method for unsupervised object-specific feature selection, as opposed to more common non-object specific feature quantization techniques employed by most existing classification schemes for generating feature codebooks. In this sense it is also a useful tool that can be used in conjunction with existing classification techniques in order to improve their performance. In addition, UFO performs feature parameter optimization jointly with the feature selection. Optimizing the parameters of the features, such as detection thresholds or pairwise offsets, is an important aspect of feature selection. Separating parameter optimization from the feature selection (as in many contemporary object recognition systems) may reduce the system performance. We can view the UFO as a bottom-up unsupervised learning approach. It composes category models out of small features, as opposed to a more standard top-down approach which is fitting some model to the data, e.g., as was done in UCA. Figure 3 illustrates the UFO approach.

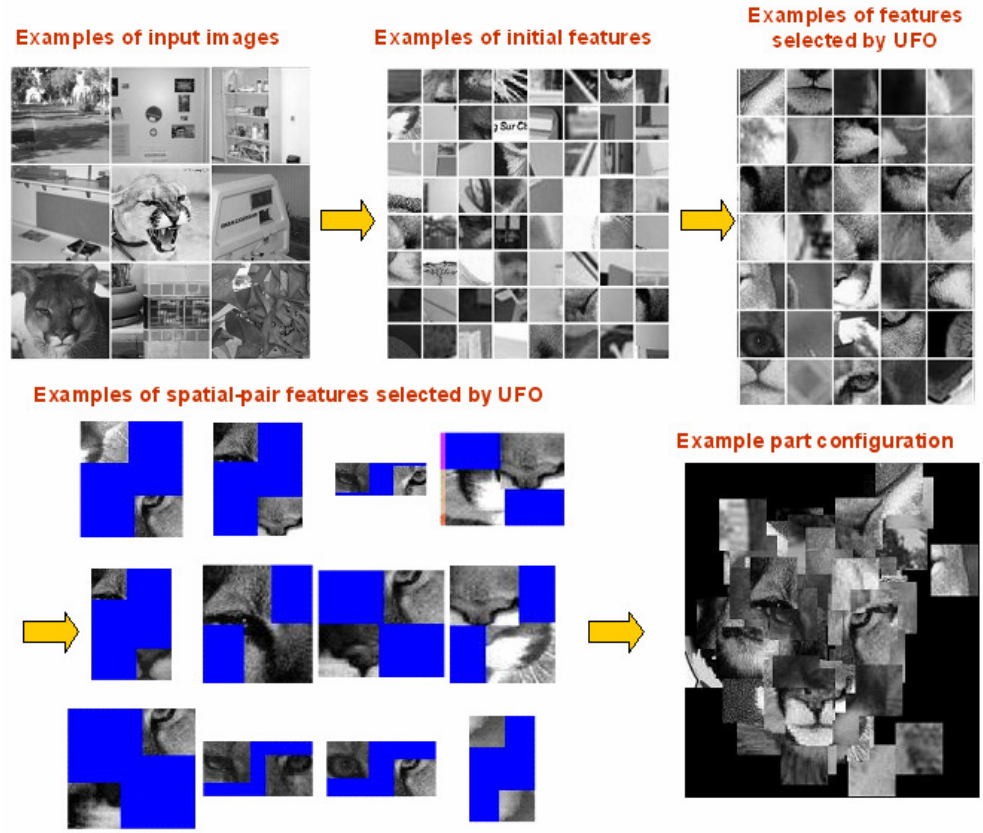


Figure 3: An example of the application of the UFO method to cougar class from Caltech-101.

3. ***The chains model*** – non-parametric model for detecting highly non-rigid and flexible object parts, such as detecting human hands, elbows or horse hooves, described in [Karlinsky et al., 2010a]. The same model was shown to be capable of complete object detection and of salient contour detection. As was shown in the paper, the method can generalize between different people and backgrounds and is capable of accurate detection without using any color or motion information. The core idea of the method is that parts that are difficult to detect on their own, become more easily detectable when considered in a context of surrounding parts. In the method, the context is used through an ensemble of feature chains that connect the easily detectable parts (such as the face) and the hard to detect flexible parts (such as the

hand). Both the easy to detect parts and the chain features form the context are necessary to detect the flexible parts. The approach comes with a simple and efficient method to perform inference over the ensemble of possible feature chains. Figure 4 illustrates the method. Currently, our approach requires prior existence of the source part detectors (such as the face detector). In practice, these source part detectors can be discovered automatically by applying the unsupervised learning techniques, such as UCA or UFO. In fact, we have also shown that the chains model may work even without the existence of the source part and can be used as the source part detector itself [Karlinsky et al., 2010b].

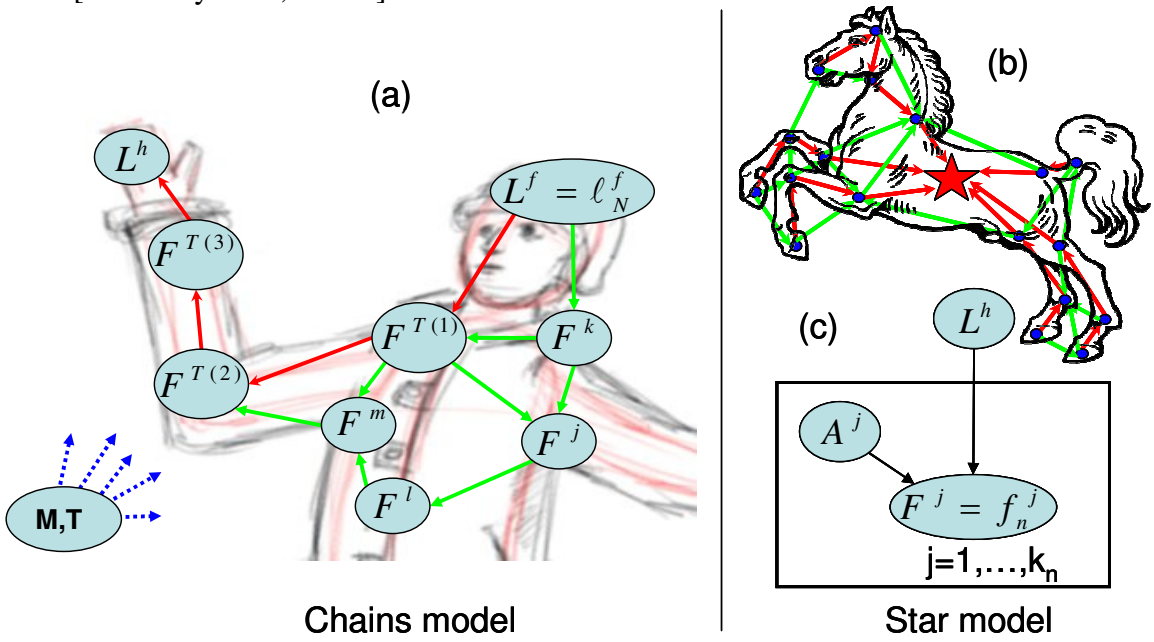


Figure 4: Illustration of the chains method. For details please see paper (Appendix D). (a) Chains model applied for part detection. The unobserved variable L^h is the location of the target part (e.g. hand). $L^f = \ell^f_N$ is the observed location of the reference part (e.g. face) in image I_N. The edges symbolize the chains 'feature graph' constructed over the set of observed features $F^j = f^j_N$. Unobserved variables T and M (affecting all the nodes) select a simple path T (red) of length M on the graph. Features not on this path are generated from their respective 'world' distributions. During inference we marginalize over T and M, summing over paths on the graph going from the reference part to each candidate location of the target part. (b) Chains model applied for full object detection as an extended star model. (c) Star model. Unobserved binary variables A_j (one per feature), control whether the feature is generated with respect to the star center or from its 'world' distribution.

4. ***Human body parts interpretation technique –*** a variant of the articulated model for human body interpretation that employs perceptual grouping and local contextual segmentation techniques for detecting body parts, such as the face, torso and upper and lower arms. It is described in Appendix A. Figure 5 briefly illustrates the steps of the algorithm. An interesting extension of the proposed technique is to connect it to the chains model for the deformable parts detection. This can be achieved in two ways. The simpler way is to filter the generated limb interpretation hypotheses using hands / joints that can be efficiently detected by the chains model. The more interesting way is to use the limb hypotheses, generated either through perceptual grouping or through a local segmentation techniques, as features or parts involved in the chains model inference. Both these extensions might be interesting topics for future research in the direction of human body interpretation.

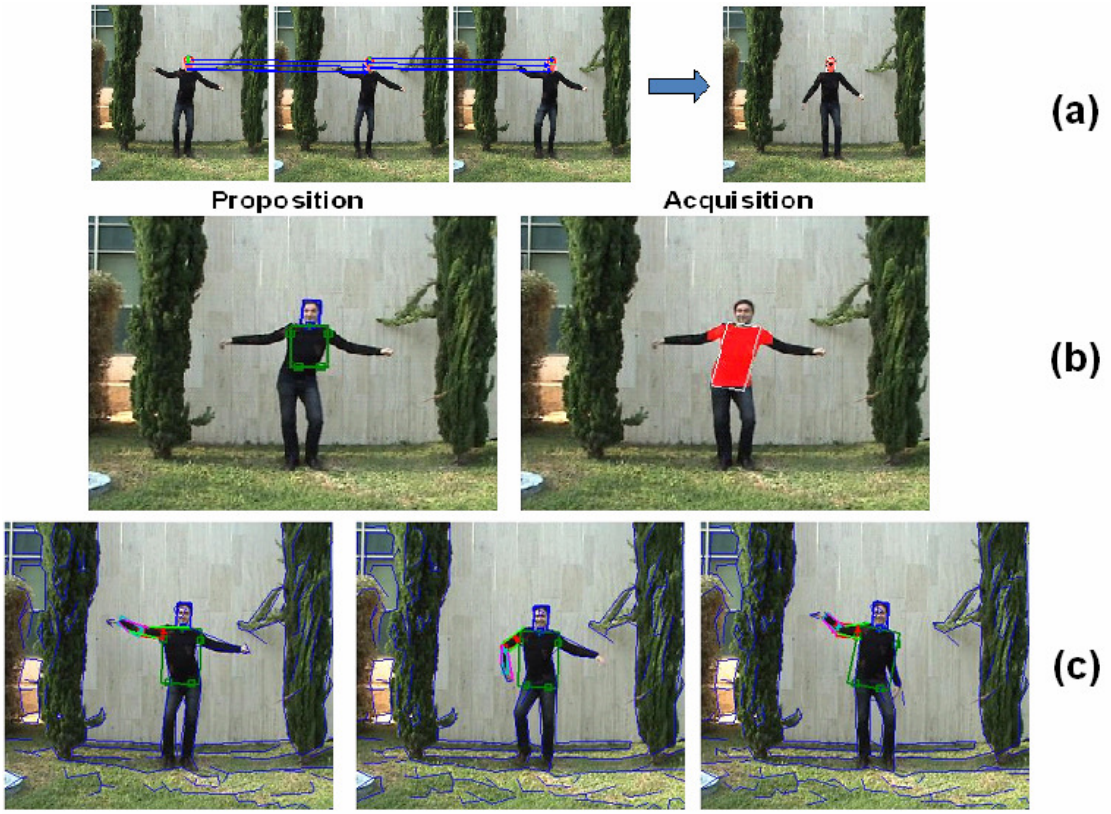


Figure 5: Illustration of the human body parts interpretation technique. For details please see Appendix A. (a) Face detection, tracking and acquisition; (b) Body acquisition; (c) Arms detection based on perceptual grouping. Alternative method of using acquisition by local segmentation to detect the arms is also described in Appendix A.

Methods for action recognition and analysis

In the course of this study we have developed two approaches for static and dynamic transitive action recognition. The static and dynamic approaches are described in detail in [Karlinsky et al., 2010b] and in Appendix B respectively and are briefly summarized below.

1. ***An approach for identifying similar transitive actions in single images*** – described in [Karlinsky et al., 2010b]. The key idea is to use body anchored features, made available by accurate human body parts detection, in order to distinguish between similar actions given only a single image depicting the action. The ability to accurately detect the relevant parts is paramount in the presented approach. During training, this allows the automatic discovery and construction of implicit non-parametric models for different important aspects of the actions, such as relevant objects, types of grasping, and hand-object and body part configurations. During testing, this allows the approach to extract distinctive features specifically from restricted image regions that contain the relevant information for recognizing the action. As a result, we eliminate the need to have a priori models for relevant objects, and become capable of learning from action labels only, without labeling of the body parts and relevant objects. Figure 6 illustrates the similar static transitive action

recognition task and the proposed approach.

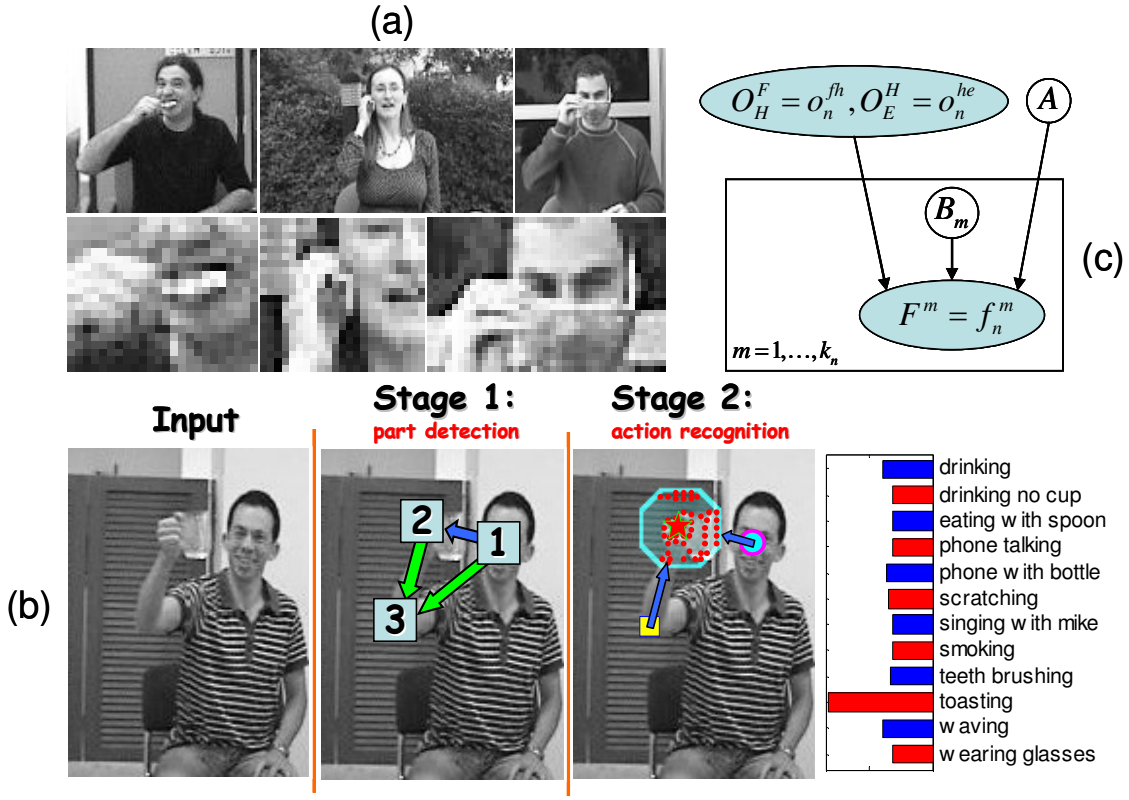
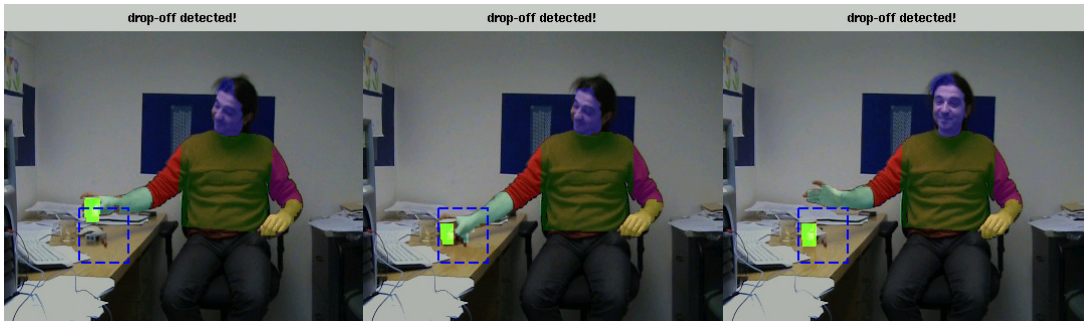


Figure 6: (a) Examples of similar transitive actions identifiable by humans from single images (brushing teeth, talking on a cell phone and wearing glasses). (b) Illustration of a single run of the proposed two-stage approach. In the first stage the parts are detected in the face→hand→elbow order. In the second stage we apply both action learning and action recognition using the configuration of the detected parts and the features anchored to the hand region; the bar graph on the right shows relative log-posterior estimates for the different actions. (c) Graphical representation (in plate notation) of the proposed probabilistic model for action recognition (see section 2.2 in the paper for details).

2. ***Dynamic action recognition through full human body interpretation*** – the approach is summarized in Appendix B. The key idea is to detect the body parts first and then to detect the "seeds" for the action events as changes in the background appearing following a visit from the relevant body part (e.g. hand in a pick-up/drop-off action). These seeds are then propagated in time using tracking to detect both the action and the relevant object involved in it. Prior body parts interpretation allows to both recognize changes occurring following an activity of the relevant body parts and to disregard changes due to motion of the irrelevant body parts. Figure 7 shows several example frames from the automatically detected "pick-up" and subsequent "drop-off" events (no erroneous events where detected in tested video).

(a)



(b)

Figure 7: (a) Three consecutive frames of the detected "pick up" event; (b) Three consecutive frames of the "drop off" event occurring later in the sequence.

Literature review – action recognition & human body interpretation

In this section we will focus on the literature related to action recognition and human body interpretation. The review of the literature that concerns with unsupervised learning and face detection can be found in the respective papers provided in the Papers section.

Extensive literature review on recent advances in action recognition can be found in [Kruger et al., 2007] and in [Karlinsky et al., 2010b]. Their taxonomy of action recognition approaches is shown in Figure 8.

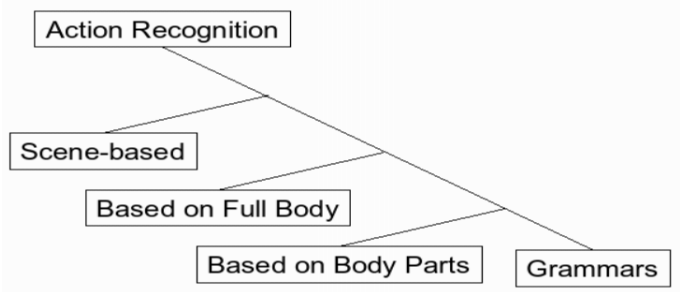


Figure 8: Taxonomy of action recognition approaches in [Kruger et al., 2007].

Our approach fits the "based on body parts" cell of this taxonomy. According to the survey, most other approaches that fit that cell can be categorized as follows. Approaches based on interpreting silhouettes obtained by a change detection mechanism. The 3D-model based body tracking approaches, where the recognition of (usually periodic) action is used as a loop-back to support pose estimation. Approaches based on motion capture systems. In contrast, in this study we approach the human (and object) parts detection problem independently of the action, in single images (without using change detection masks) and by a purely vision based techniques. In that sense, our approach is more related to the literature focusing purely on human body interpretation without concerning with action recognition, such as: pictorial structure based [Felzenszwalb & Huttenlocher, 2005; Ramanan, 2006; Ferrari et al., 2008; Buehler et al., 2008; Andriluka et al., 2009; Kumar et al., 2009], classifying segments to body parts [Mori et al., 2004], or performing a search in complex multi-dimensional parameter space of a very detailed human model [Balan et al., 2007].

Focusing on capability of detecting human body parts in still images requires our approach not to rely too strongly on temporal information comprising an important

backbone of many state-of-the-art methods. In [Ramanan et al., 2005] distinctive whole body poses are detected by rigid models and connected by tracking. Both [Sigal et al., 2004] and [Sigal and Black, 2006] employ tracking body parts via a spatio-temporal probabilistic model, but require manual initialization and background subtraction (being usually the prerogative of methods relying on temporal information). Finally, [Ferrari et al., 2008] and [Buehler et al., 2008] build upon the ideas of [Ramanan et al., 2005], [Ramanan, 2006] and [Sigal and Black, 2006] to form a fully automatic body parts detector and tracker in movies (again strongly relying on temporal information).

The human body can be roughly approximated by an articulated 3D model of connected cylinders. However, this approach has several drawbacks including difficulty of handling loose clothing and a large space of unknown parameters to estimate. To reduce the set of estimated parameters, it is a more common practice to use an articulated 2D model of the human body, comprised of a set of connected rectangles [Felzenszwalb & Huttenlocher, 2005; Ramanan, 2006; Ferrari et al., 2008; Buehler et al., 2008; Andriluka et al., 2009; Kumar et al., 2009]. One of the common problems of the 2D approach is with handling some of the body deformations that cannot be explained by 2D motion of the parts, such as foreshortening of the limbs during out-of-plane rotations. The human body interpretation approach proposed in Appendix A addresses this problem by explicitly allowing foreshortening of limbs inside the 2D articulated model. In addition, the chains model (described in [Karlinsky et al., 2010a]) does not require the human body deformation to be explainable by any 2D, 3D or any other parametric model, instead the observed deformation should contain feature chains that were partially observed during training.

Methodology

In this section we will discuss several aspects of the proposed methodology, namely: sequential modeling for the task of body parts interpretation and action recognition and interpretation. This methodology underlies all the published and unpublished results derived in this study that are presented in the Papers section and in the Appendices A and B.

As shown in [Karlinsky et al., 2010b], standard classification methods are insufficient for the task of ‘visual verb phrase’ recognition. Instead we perform a sequence of several stages, which are useful for recognizing many different actions. The first step is body interpretation which includes detecting the people in the image, including their (relevant) body parts. For the case of dynamic images (a sequence of several consecutive movie frames), we also detect the objects participating in the action, including the detection of dynamic events, particularly related to the interaction of the hands and these objects. For the case of single (static) images, we detect the appropriate body configuration, and grasping type, or apply simple non-parametric models conditioned on the results of body parts interpretation (and hence using body anchored features) in order to recognize the action. We briefly describe below these stages and how they are put together for action recognition. We also discuss briefly the general notion of sequential interpretation.

The problem of detecting human body parts in still images is a difficult one and was not yet fully solved. Quoting [Mori et al., 2004], "the problem of recovering human

body configurations in a general setting is arguably the most difficult recognition problem in computer vision". In this report we propose two approaches ([Karlinsky et al., 2010a] and Appendix A) to human body parts detection. Both approaches sequentially apply models for the source part (such as the face) and expand their detections to localize other parts (such as hands or limbs). The sequentiallity is a key aspect of our approach as it allows using arbitrarily complex models for part connections as well as for the part detection itself. For instance, as is explained in more detail in Appendix A, we may use local "contextual" segmentation to detect amorphous parts such as torso and limbs, or represent part-to-part transitions using feature chains partially observed during training and efficiently recognized during testing [Karlinsky et al., 2010a]. In addition, sequential inference becomes necessary when modeling actions involving objects (e.g. transitive actions). Since the same action, like "picking up" may be performed with many types of objects, relevant objects cannot be modeled or identified a-priori, but only sequentially after the necessary parts of the person (such as a hand) have been detected.

Our approach for human body parts detection is summarized in figure 9 and figure 10 provides some taste of the obtained results. The details of the human body interpretation approaches are given in the Papers section and in Appendix A. In [Karlinsky et al., 2010b] and in Appendix B we propose the ways by which a successful human body interpretation might be used to do action recognition, describing both the proposed action recognition algorithms and their obtained results.

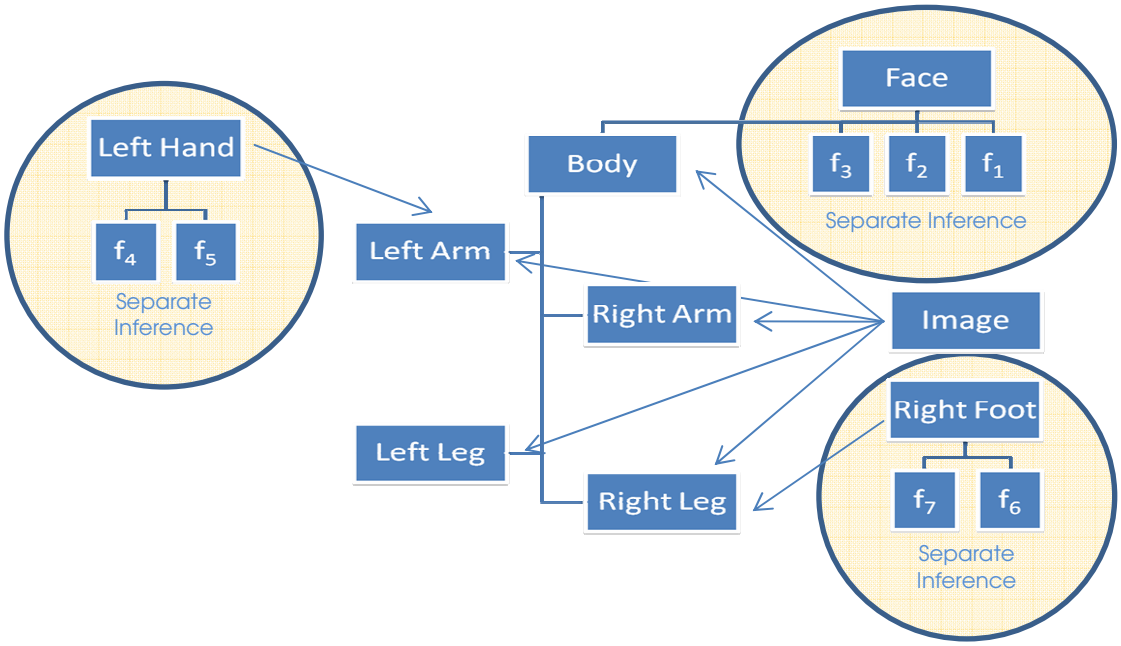


Figure 9: The schematic illustration of the sequential approach for body parts detection. Some parts, such as face and hands are detected first, using a separate inference step, such as face detection by applying UCA or hand detection by applying the chains model [Karlinsky et al., 2010a]. The other, more ambiguous parts, such as arms and legs, are then detected with respect to the found locations of the separately detected parts (Appendix A). The same scheme then extends to the detection of action-related objects and actions themselves in the context of the detected body parts ([Karlinsky et al., 2010b] and Appendix B).

Figure 10: Some examples of frames from different tested datasets, on top the results of full body interpretation presented in Appendix A, at the bottom are results of the hand detection using the chains model described in [Karlinsky et al., 2010a].

Papers

This section summarizes the research papers published in the course of this study and provides additional results not appearing in the papers (where applicable).

Unsupervised Classification and Part Localization by Consistency Amplification (UCA)

(Attached paper [Karlinsky et al., 2008a])

The UCA algorithm can be applied for hand and hand pose detection. The hand is considered as a rigid object, with different hand poses as sub-classes of that object. This approach requires going over different orientations and scales to detect the hand (the UCA is only translation invariant). Figure 11 shows its performance on a hand detection task in a test dataset, and figure 12 shows its performance on the hand pose estimation task.

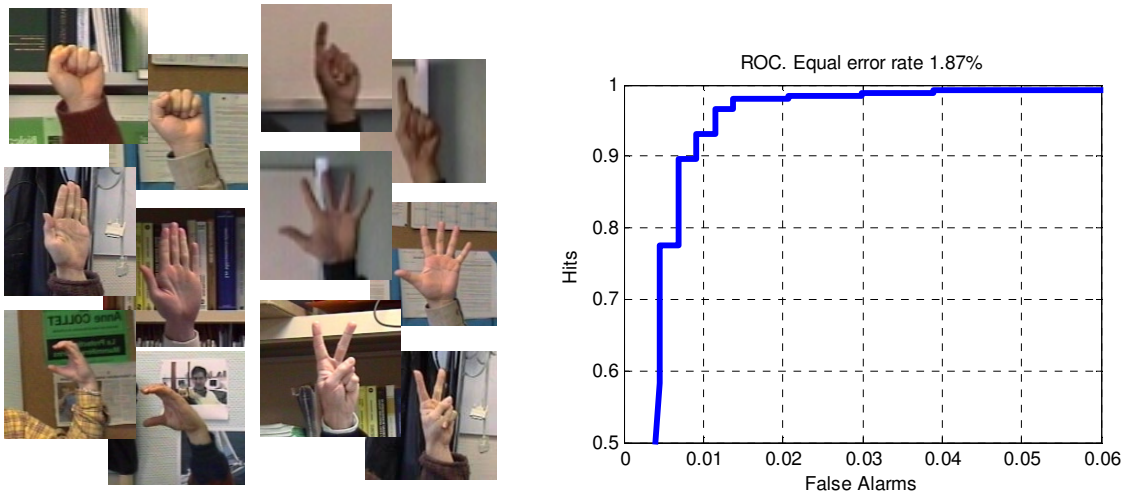


Figure 11: On the left examples of the hand images in the dataset, on the right ROC for the hand-vs.-background classification task against Google backgrounds.

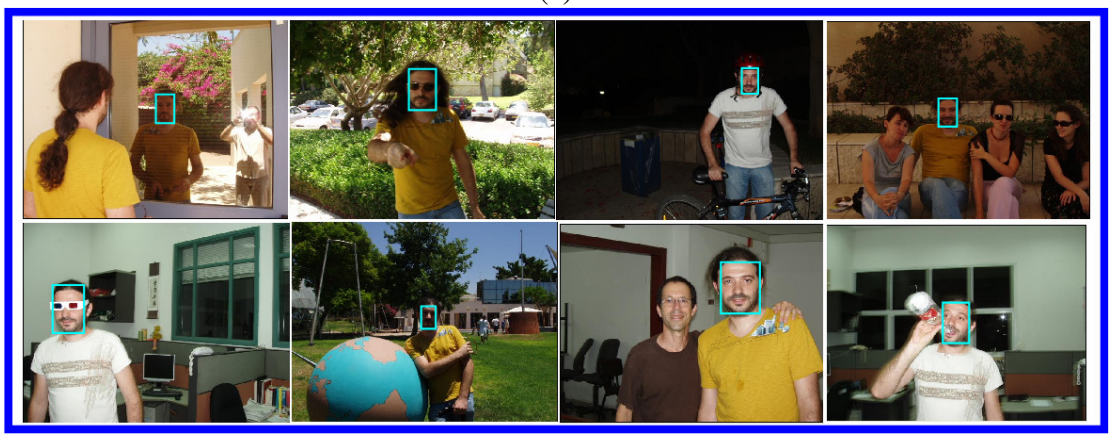
	Fist	Open	Snake	V	Five	Point
Fist	1	0	0	0	0	0
Open	0.2	0.8	0	0	0	0
Snake	0	0	1	0	0	0
V	0	0	0	1	0	0
Five	0	0	0	0	1	0
Point	0	0	0	0	0	1

Figure 12: Confusion matrix for hand pose estimation task distinguishing between 6 different poses. The example poses are shown in figure 11 (on the left).

Combined model for detecting, localizing, interpreting and recognizing faces

(Attached paper [Karlinsky et al., 2008b])

(a)



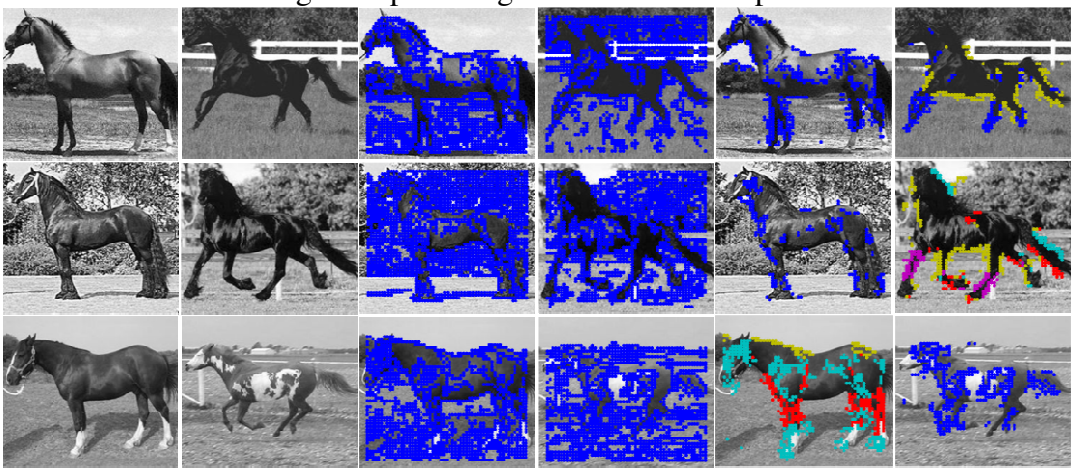
(b)

Figure 13: Demonstrates the face detection & recognition results. (a) Shows some detection examples on two datasets, one collected by us (red box), and the MIT-CMU dataset (green box); (b) Shows some detection + recognition examples on a single person recognition task (see paper for more details).

Unsupervised Feature Optimization (UFO): simultaneous selection of multiple features with their detection parameters

(Attached paper [Karlinsky et al., 2009])

Figure 14 demonstrates that UFO can also be used for extracting the features of a foreground object in an image. In the corresponding experiment, the initial feature pool was comprised of the SIFT descriptors for 20x20 patches around all the Canny edge points in the image. Figure 14c shows some of the feature clusters obtained by the UFO when selecting and optimizing features from this pool.



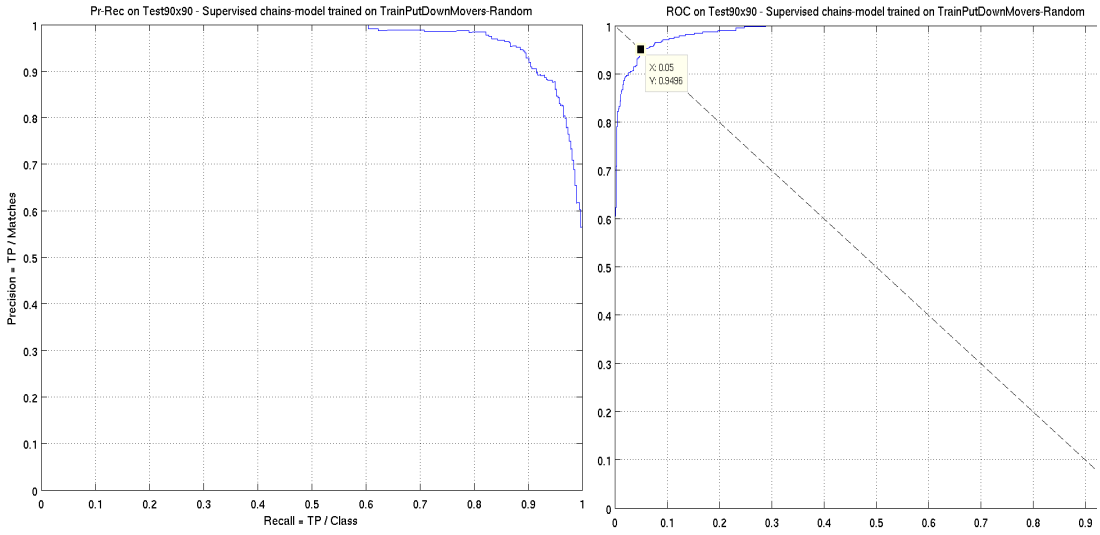
(a) (b) (c)

Figure 14: (a) Original images; (b) Initial feature pool, computed around a sub-sampled set of Canny edge points; (c) Color coded feature clusters obtained by the UFO.

The chains model for detecting parts by their context

(Attached paper [Karlinsky et al., 2010a])

Figure 15 shows additional results of chains model for object detection (no source part) applied to the task of detecting large (40x40 pixel) hands. Figure 16 shows an illustration of the temporal chains model for dynamic object detection.



(a) (b)

(c)

Figure 15: Additional results on the task of large (40x40 pixels) hands detection using the chains model for object detection (without using the source part). Taken from an ongoing project with (lead by) Danny Harari and Nimrod Dorfman. (a) Precision-Recall on hand vs. background classification task; (b) ROC, EER=0.05; (c) Detection examples, the maxima are numbered from highest to lowest and the heat-map on top of the image shows the detection probability mask (red=high).

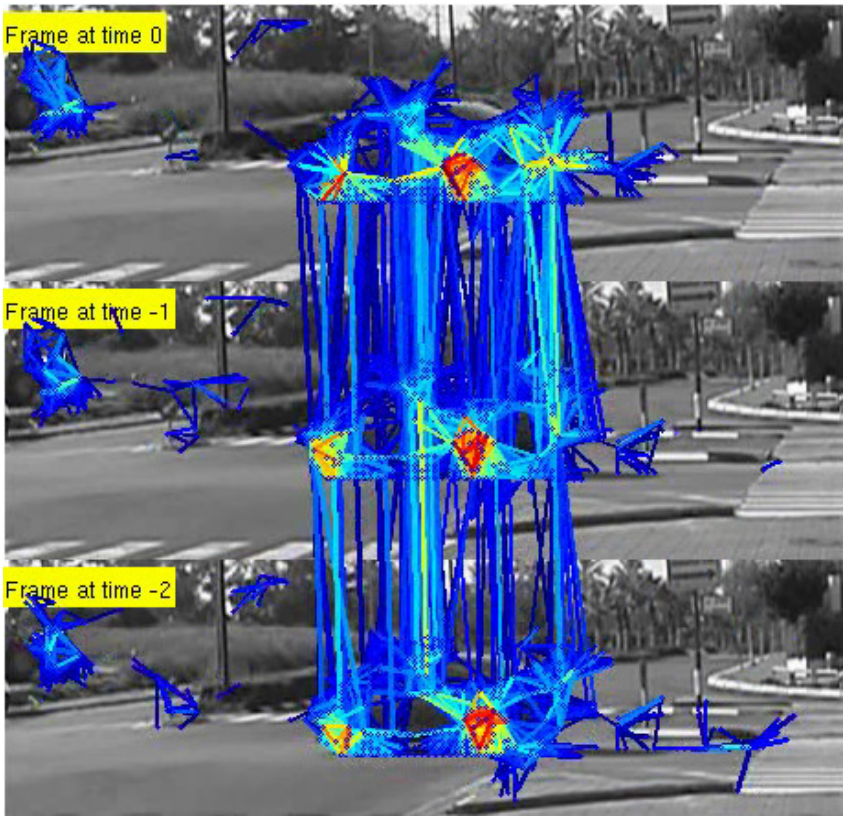


Figure 16: chains for temporal object detection (taken from ongoing project with (lead by) Danny Harari). The figure shows a spatio-temporal chains model "feature graph" for car detection overlayed on three consecutive frames of the test sequence.

Identifying similar transitive actions in single images

(Attached paper [Karlinsky et al., 2010b])

In addition to the static action datasets used in the experiments described in the paper, we have collected one more dataset containing 12 people performing 4 actions, two kinds of 'drinking' (with transparent and opaque glasses) and two kinds of 'talking on the phone' (with regular and cellular phones). Three sequences of each action were used to train the hand detector (sequences of different triplet of people were taken for each action), and the remaining 9 sequences (of each action) were used to test the action recognition. Thus, for this dataset, the static action recognition was tested on frames from 36 sequences of 12 people, each sequence consisting of approximately 400 frames (the method was applied on each frame separately in a 'person leave-one-out' fashion described in the [Karlinsky et al., 2010b] paper). The obtained average

accuracy was 91±10%. In addition, the set of body anchored features used in the approach can be either all the features surrounding the detected hand location, or a refined subset of features detected as more informative to the classification task. In our case, we have experimented with a simple likelihood ratio based feature selection criterion, in which top scoring features according to the ratio $P\left(f_n^m, A=a, o_n^{fh}, o_n^{he}\right) / P\left(f_n^m, A\neq a, o_n^{fh}, o_n^{he}\right)$ where chosen from each training image I_n (see paper for notations and the details of the empirical probability computation). Following the feature selection the average accuracy of the method has improved to 97.5±5%. Figure 17 provides a comparison with the state-of-the-art method for object detection of [Felzenszwalb et al., 2008] that is trained by using images displaying different actions as images having different object labels.

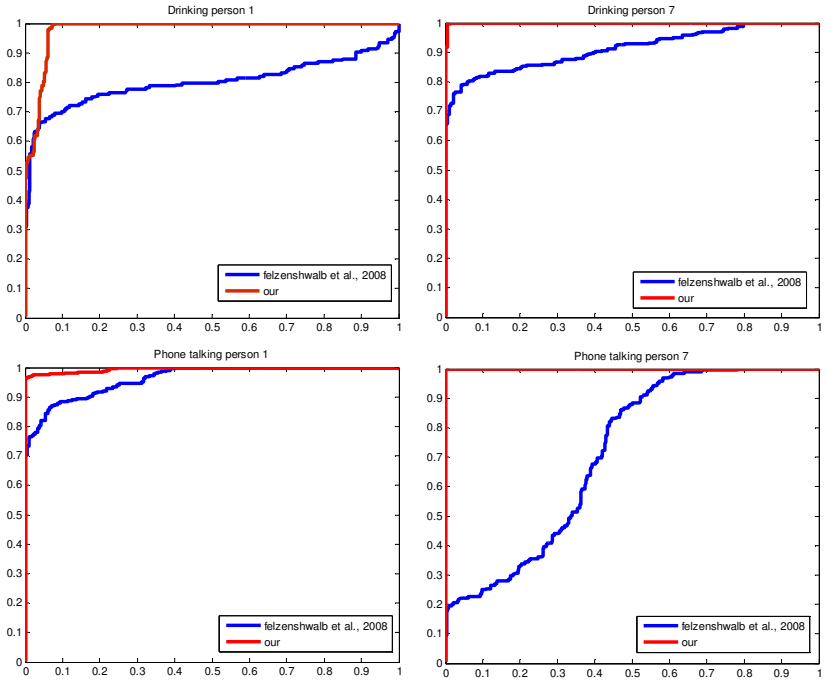


Figure 17: Comparison of our approach with state-of-the-art object detection approach by [Felzenszwalb et al., 2008] on the task of static object detection: drinking vs. talking on the phone. The red line is the ROC of our approach and the blue line is the ROC by [Felzenszwalb et al., 2008] (using their own implementation). Both methods were trained on the same set of examples of 11 people performing the actions and tested on the remaining 12th person. Results for random two people are shown in the left and the right column graphs respectively. Each graph is labeled by the name of the action which is considered positive and the number of the person.

Discussion - Sequential inference and sequential modeling

As noted above, action recognition requires the recognition of multiple parts and their relationships. In this section we describe how the interpretation process can be naturally divided into a sequence of more elementary problems.

Consider an example distribution:

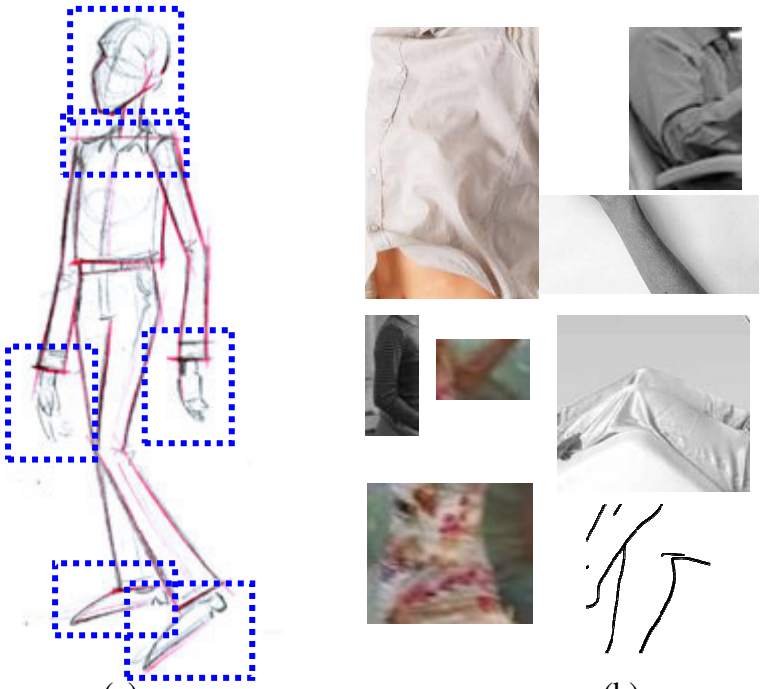
$$P(X_1, X_2, X_3, X_4, X_5) = P(X_1, X_2, X_3) \cdot P(X_4, X_5 | X_1, X_2, X_3)$$

A standard inference question is computing the MAP – Maximum A-Posteriori assignment to the variables X_i: $\arg\max_{X_1, X_2, X_3, X_4, X_5} P(X_1, X_2, X_3, X_4, X_5)$. However, in many cases, especially when dependencies between variables form undirected loops in the graphical representation of the distribution by the Bayesian Network (BN), this inference task is very difficult. A *sequential* approach to inference would be to

decompose the distribution to two or several parts and solve the inference problem sequentially, each time assuming that the results so far are fixed:

$$\arg\max P(X_1,\ldots,X_5) \approx \left\{ \begin{array}{c} \hat{X} = \arg\max P(X_1, X_2, X_3) \\ \arg\max P(X_4, X_5 \,|\, \hat{X}) \end{array} \right\}$$

The intuition behind this kind of approach, especially applied to the problem of human body parts detection is that some parts are both significant and identifiable by themselves even without the context of human body. For instance, as illustrated in figure 18a, the hands, the boots and the face, although out of context, still appear as they are to the human observer and are not considered as noise. In contrast, some parts of the human body, like limbs or torso, are not only hard to detect without the context of the face and / or other separately detectable parts, but also can not be identified on their own by a human observer when seen out of this context (as illustrated in figure 18b).



(a) (b)

Figure 18: (a) Some parts of the human body are identified by human observers even without context (blue boxes); (b) Some other parts need context to be identified, and their structure may be too complex to be defined by a simple distribution.

In addition to computation speed-up provided by the sequential approach, in some cases the model distribution itself is "effectively un-computable" without applying it. To illustrate this point, consider a simple model with two hidden variables: *Face* and *Body* , each representing the data we want to extract about the respective body parts, like image location, outline and etc. In the most general case, the model is defined as:

$$P(Face, Body \mid Image) = P\big(Face \mid f_1(Image)\big) \cdot P\big(Body \mid f_2(Face, Image)\big)$$

where f_1 and f_2 are two functions representing the computations necessary for extracting all the desired information about the *Face* and the *Body* respectively. Since the *Image* is an observed variable, $P\big(Face \mid f_1(Image)\big)$ takes a classical form of feature based computation. The $f_1(Image)$ are the features with respect to which

the distribution of the *Face* is defined. However, if the function $f_2(Face, Image)$ is too complex, such as in the case that the *Body* is defined as an output of a segmentation process applied in a region below the detected face, it is effectively impossible to compute f_2 for every possible *Face* value (such as required by standard message passing techniques, e.g. BP). Hence the only possible inference for this type of model is the sequential one.

Summary

Action recognition deals with image examples which often have some common 'semantic' aspects, but are usually highly variable. This variability is so large that simple classification, which relies on similar arrangements of common parts, is insufficient. This was empirically validated in [Karlinsky et al., 2010b], where we have shown that a state-of-the-art object detection schemes perform poorly on the task of distinguishing between similar static transitive actions like phone talking, drinking from a cup, etc. Before we proceed to action recognition, we therefore need to achieve a full interpretation of relevant body parts, relevant objects, and their relations. In our sequential approach, we propose to start with a set of independently recognizable configurations of features and parts, and then expand the interpretation to other configurations, relevant objects and events. We do the expansion either by directly applying inference on a pre-computed set of possibilities or by intermediate steps of guided image processing. Using the sequential approach enables the action recognition scheme to obtain a high level semantic generalization. For example, seeing an action performed by one hand we can immediately generalize to performing the action using the other hand (e.g. using body anchored features from [Karlinsky et al., 2010b]), although visually the two ways to perform the action are very dissimilar.

The chains model developed in [Karlinsky et al., 2010a] can be extended to provide a framework for the sequential approach. Currently the chains model uses sequences of observed features (small probabilistically detectable elements of the internal context of the object) to connect known parts of the object to the parts being detected. As indicated in [Karlinsky et al., 2010a], the future versions of the same model may include: (1) unobserved variables along the chains indicating 'helper' parts that will allow sharing of training information between partially similar feature chains; (2) a hierarchy of chains that will represent an increasingly general concepts as we go up in the hierarchy, e.g. small rectangular image patch features → rigid sub-parts (e.g. lower arm, upper arm) → deformable parts (e.g. an arm) → deformable object (e.g. a person) or a static action (configuration of parts and relevant objects [Karlinsky et al., 2010b]) → a scene or a group of objects and actions comprising the image (e.g. an office scene); (3) dynamic chains that link a temporal sequence of images and dynamical events (that can again be organized in a hierarchy of increasingly general action or activity related concepts). The proposed extensions provide a principled way of implementing a sequential approach based scheme and integrating the different semantic levels of object, scene, action and activity interpretation within a single unified framework.

In the course of this work several useful learning and interpretation tools were developed. Those include algorithms for top-down (UCA) and bottom-up (UFO) unsupervised learning and feature parameter optimization, the chains model for non-parametric detection of highly deformable parts of non-rigid objects (such as a human

hand), and a novel variant of the human full body interpretation algorithm. We have also shown that these tools provide the interpretation basis necessary for subsequent successful static and dynamic action recognition. We believe that many exciting future object and action recognition applications may benefit from using these tools. Figure 19 schematically illustrates (by example) a sequential scheme in which all the developed tools may be integrated in order to form a sequential scheme for weakly supervised action learning and recognition including the automatic identification of objects and parts relevant for the action.

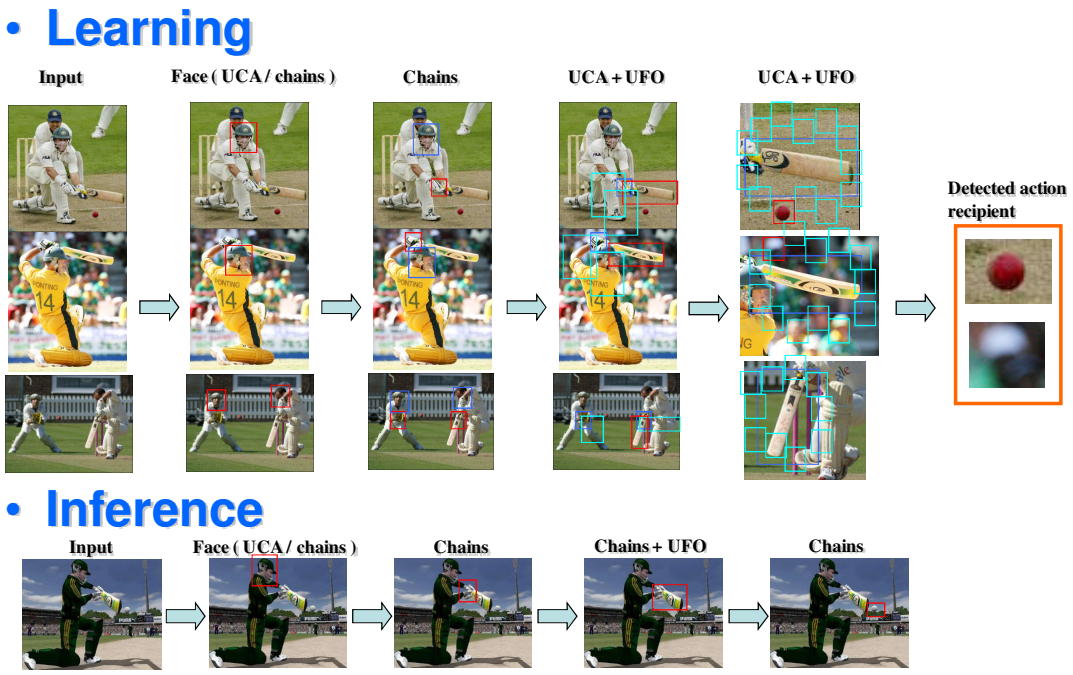


Figure 19: an illustration by example (cricket) outlining a weakly supervised sequential action learning and recognition scheme that uses the different approaches developed in the course of this study. The input to the computation does not provide any details regarding the features / objects / events that are important to the action, and hence sequential training steps are applied to discover them automatically. At each training stage the results of the learning (red boxes) are obtained using the results of the previous stage (blue boxes). Using previous stage results also restricts the candidate search space (cyan boxes) in which we need to look for the objects and parts that are relevant for the current stage. At the first two stages the face and the hand are detected, using the UCA based (for faces) and chains model based schemes (for hands or faces). Subsequently, two successive unsupervised learning stages are applied to detect the tool (cricket bat) and the action recipient (the ball) categories using the developed unsupervised learning techniques UCA and UFO. During inference the sequential approach can be implemented by using the chains model based framework described in the text above.

In addition, several ongoing research projects suggest that the chains model is not only a useful tool for both deformable parts and objects recognition, but can also be applied for salient contour detection and dynamic object recognition (recognition of objects in a temporal sequence of video frames). The chains model inference algorithm involves two simple computations: finding approximate nearest neighbors and a few matrix multiplications. These simple computations are sufficient to perform a range of complex visual tasks indicated above. It is plausible that these computations may also be implemented in the brain. This suggests that another interesting future research direction regarding the chains model may be to look for its neural correlates.

References

[Kruger et al., 2007] Volker Kruger, Danica Kragic, Ales Ude and Christopher Geib. "The Meaning of Action - A review on action recognition and mapping". Advanced Robotics, Volume 21, Number 13, 2007, pp. 1473-1501(29).

[Ramanan, 2006] Deva Ramanan. "Learning to parse images of articulated bodies". NIPS, 2006.

[Ramanan et al., 2005] D. Ramanan, D. A. Forsyth, and A. Zisserman. Strike a pose: Tracking people by finding stylized poses. CVPR, 2005.

[Felzenszwalb& Huttenlocher, 2005] Pedro Felzenszwalb and Daniel Huttenlocher. "Pictorial Structures for Object Recognition". Intl. Journal of Computer Vision, 61(1), pp. 55-79, January 2005.

[Mori et al., 2004] Greg Mori, Xiaofeng Ren, Alexei A. Efros and Jitendra Malik. "Recovering Human Body Configurations: Combining Segmentation and Recognition". CVPR, 2004.

[Balan et al., 2007] Alexandru O. Balan, Leonid Sigal, Michael J. Black, James E. Davis and Horst W. Haussecker. "Detailed Human Shape and Pose from Images". CVPR, 2007.

[Viola & Jones, 2001] Paul Viola and Michael Jones. "Robust Real-time Object Detection". Intl. Journal of Computer Vision, 2002.

[Lucas & Kanade, 1981] Bruce D. Lucas and Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. International Joint Conference on Artificial Intelligence, pages 674-679, 1981.

[Boykov et al., 2001] Yuri Boykov, Olga Veksler, Ramin Zabih. "Efficient Approximate Energy Minimization via Graph Cuts". IEEE transactions on PAMI, vol. 20, no. 12, p. 1222-1239, November 2001.

[Sigal et al., 2004] L. Sigal, S. Bhatia, S. Roth., M. Black, and M. Isard. Tracking loose-limbed people. CVPR, 2004.

[Sigal and Black, 2006] L. Sigal and M. Black. Measure locally, reason globally: Occlusion sensitive articulated pose estimation. CVPR, 2006.

[Ferrari et al., 2008] Vittorio Ferrari, Manuel Marın-Jimenez and Andrew Zisserman. Progressive Search Space Reduction for Human Pose Estimation. CVPR, 2008.

[Buehler et al., 2008] P. Buehler and M. Everingham and D.P. Huttenlocher and A. Zisserman. Long Term Arm and Hand Tracking for Continuous Sign Language TV Broadcasts. BMVC, 2008.

[Andriluka et al., 2009] M. Andriluka and S. Roth and B. Schiele. Pictorial Structures Revisited: People Detection and Articulated Pose Estimation. CVPR, 2009.

[Kumar et al., 2009] M.P. Kumar and A. Zisserman and P.H.S. Torr. Efficient Discriminative Learning of Parts-based Models. ICCV, 2009.

[Shor & Kiryati , 2001] R. Shor and N. Kiryati. Towards Segmentation from Multiple Cues: Symmetry and Color. Multi-Image Analysis, 2001.

[Isard & Blake, 1998] M. Isard and A. Blake. CONDENSATION - conditional density propagation for visual tracking. IJCV, 1998.

[Mount & Arya, 1997] D. Mount and S. Arya. Ann: A library for approximate nearest neighbor searching. CGC 2nd Annual Workshop on Comp. Geometry, 1997.

[Felzenszwalb et al., 2008] P. Felzenszwalb, D. McAllester, D. Ramanan. A Discriminatively Trained, Multiscale, Deformable Part Model. CVPR, 2008.

[Kalrinsky et al., 2008a] L. Karlinsky, M. Dinerstein, D. Levi, and S. Ullman. Unsupervised Classification and Part Localization by Consistency Amplification. ECCV 2008.

[Kalrinsky et al., 2008b] L. Karlinsky, M. Dinerstein, D. Levi, and S. Ullman. Combined Model for Detecting, Localizing, Interpreting, and Recognizing Faces. Faces in Real-Life Images workshop (ECCV 2008).

[Kalrinsky et al., 2009] L. Karlinsky, M. Dinerstein, and S. Ullman. Unsupervised feature optimization (UFO): Simultaneous selection of multiple features with their detection parameters. CVPR 2009.

[Kalrinsky et al., 2010a] L. Karlinsky, M. Dinerstein, D. Harari, and S. Ullman. The chains model for detecting parts by their context. CVPR 2010.

[Kalrinsky et al., 2010b] L. Karlinsky, M. Dinerstein, and S. Ullman. Identifying similar transitive actions in single images. ECCV 2010, submitted.

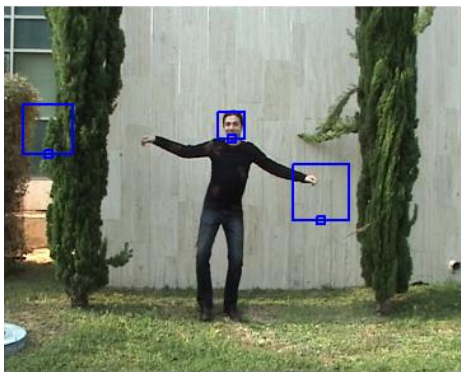
Statement about independent collaboration

I hereby declare that I was a major or equal contributor to all the papers cited in the Papers section, namely [Kalrinsky et al., 2008a], [Kalrinsky et al., 2008b], [Kalrinsky et al., 2009], [Kalrinsky et al., 2010a], [Kalrinsky et al., 2010b].

Appendix A: human body parts interpretation using a cardboard model and local contextual segmentation

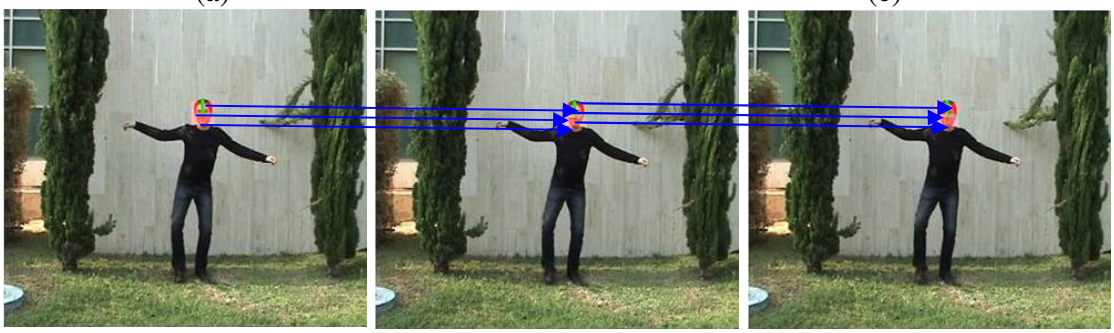
Body Parts detection algorithm

1. *Face detection* – the first step is to extract candidate regions for a face in the image. To this end one might use any of the self face detector, for instance we use Viola & Jones (V&J) face detector [Viola & Jones, 2001] (because of its high speed and need to process large amounts of video data). Since it is necessary to detect the face reliably and under different viewing angles, we can also use the algorithm and methodology we developed in a companion project dealing with unsupervised classification. The name of the project is Unsupervised Consistency Amplification (UCA) and its details are explained in Appendix A. Figure A.1a gives an example of V&J output.
2. *Face detection filtering and completion* – since V&J detections tend to be noisy and miss some of the faces in some of the frames (especially when dealing with small faces) a filtering and completion step is necessary to get rid of the misdetections. We currently perform this filtering using temporal smoothness information by applying KLT tracking [Lucas & Kanade, 1981]. This is the only place in our current implementation that uses temporal information (other then this all processing is applied to individual frames). The usage of temporal information can (and will) be easily circumvented if a more reliable (albeit slower) face detector is used, and that's where UCA will come in near future. The filtering step is illustrated in figure A.1b.
3. *Refining filtered face detections, extracting scale* – even after the face detections are filtered and only the correct detections remain. The detection bounding boxes do not appear tight around the face region and provide only a very rough guess on the real scale of the face.



(a)

(c)

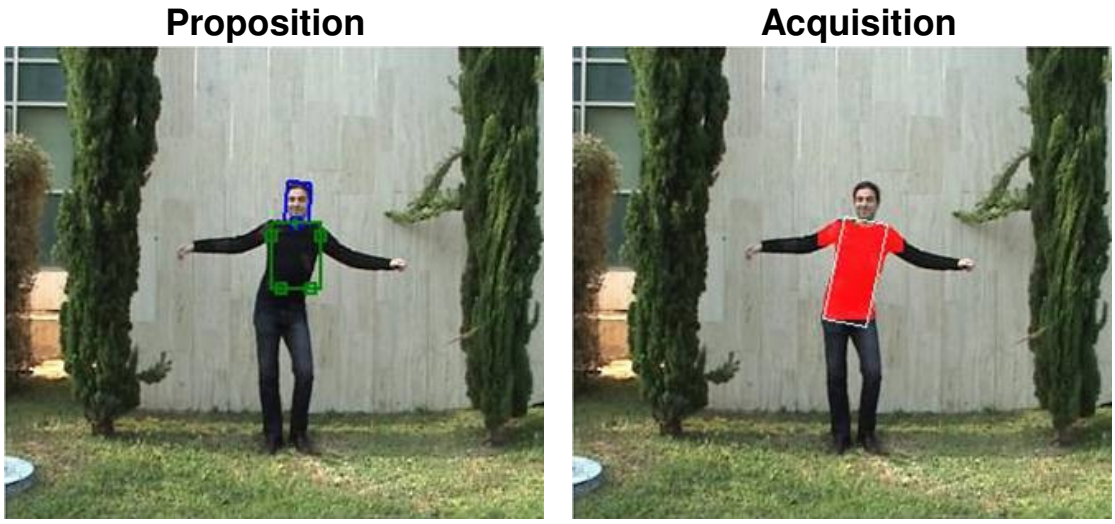


(b)

Figure A.1: (a) Sample output from V&J face detector; (b) Illustration of the face detections filtering process using tracking; (c) Sample output of the local contextual segmentation of the face region.

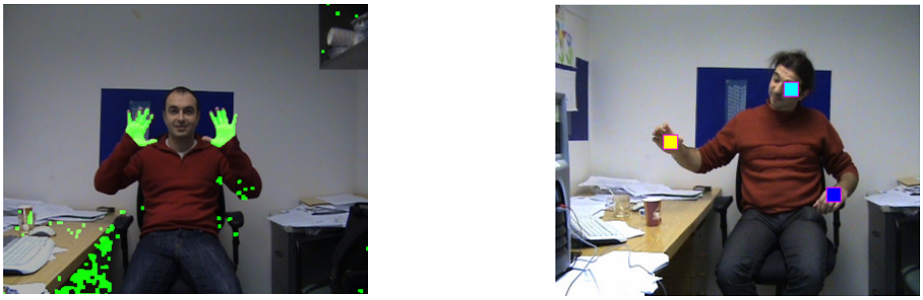
Since, in our sequential processing, we are very interested in the scale and accurate position of the face (as it provides a strong prior for subsequent computations); we apply a process of Local "Contextual" Segmentation (LCS) that is explained in a sub-section below. The LCS is initialized with the ROI of the detection and produces accurate mask of the face region from which a scale estimate for the human figure is extracted. Example of LCS output is depicted in figure A.1c.

4. *Torso detection* – applying the sequential paradigm, following the face detection and refinement, the torso of the person is detected by an LCS process initialized in a region of size and location proposed by the face. The output of the LCS is an accurate estimate of the torso region. An illustration and output example of the body detection stage is given in figure A.2.
5. *Limb termination detection* – the next step in the sequential body parts detection process is detecting the limb terminations (hands, boots, etc.). The hands detection can be done either in the "weak sense" by computing skin color masks or in a "strong sense by applying either the chains method for hand detection or the UCA trained classifier for hand and hand pose detection. Example outputs of skin color based and the UCA based techniques are shown in figure A.3. Multiple examples of hand detection using the chains model can be found in [Karlinsky et al., 2010a]. The skin color models use both a skin prior from [Shor & Kiryati , 2001] and local skin color model computed from the face mask computed as above.



(a) (b)

Figure A.2: (a) Initialization of the LCS process of the torso detection provided by the face; (b) Sample output of the LCS with accurate torso estimate.



(a) (b)

Figure A.3: (a) Output of a skin color detector; (b) Output of a UCA classifier for hands and hand poses.

6. *Limb detection* – our current implementation has two options for limb detection, namely boundary based and LCS based, following the steps described above. An interesting topic for future research is unifying these two options into a single approach exploiting the benefits of both. The conceptual difference between the options is that boundary based option directly applies inference on a set of edge segments pre-computed once for the whole image, while the LCS based option has an intervening stage of additional 'image processing' (in this case LCS), applied prior to the inference (which could not be applied a-priori without guidance obtained from the output of the previous stages of the algorithm).

(a)

(b)

Figure A.4: (a) Illustration of the boundary based option for limb detection; (b) Examples of LCS based option output.

a. *Edge / Boundary based* option is detecting "trains" of parallel edge segment pairs, connecting the estimated joint locations on the torso and the detected hand (or boot) candidates. The relative scale of the limbs is known from scale estimates based on already detected parts, which greatly reduces the search space further exploiting the sequential paradigm. Following the extraction of these limb hypotheses based on parallel edge segments, hypothesis filtering is applied. In the simplest version this filtering is just sorting based on a product of several simple features, such as: joint distance, edge coverage of the hypothesis and visual appearance similarity between respective limbs (assuming symmetry, arms and their upper and lower parts should have similar texture, as well as legs). The appearance similarity is measured by computing the probability of composing one arm's appearance from the other and vice versa. The appearance distribution is modeled by Parzen windows a-parametric approach which is applied following quantization of the image colors to several bins (we use 100 bins) and computing a histogram of clusters over the respective image region:

$$\log P(H_1 \mid H_2) \propto H_1^{\,t} \cdot \log\left(\exp\left(-0.5 \cdot D^2 / \sigma^2\right) \cdot H_2\right)$$

where H_1 is a histogram of left arm part and H_2 is the histogram of the respective part on the right arm. The sample results of the edge/boundary based option is illustrated on figure A.4a.

b. *Local Contextual Segmentation based* option is applying the LCS processes to connect between the estimated joint locations on the torso and the detected limb termination candidates. The most basic implementation of this approach is assuming that the arm is composed from sleeve on one end and skin color on the other end, both giving reasonable estimates for the size and orientation of the respective upper and lower arm parts. Then LCS processes are initialized in the respective ROIs to do the connection. The sleeve is detected by expanding the torso mask with LCS assuming it has similar texture to that of the torso. Examples of applying the LCS based option are shown in figure A.4b.

Any probabilistic detection scheme may err in the ranking of its outputs. For example the correct detection may appear not being the most probable one according to the model (e.g. several of nice detections in examples proposed by [Mori et al., 2004] are not ranked first in their "short-list"). In such cases a nice observation, exploited both in [Ramanan, 2006] and in the UCA method [Karlinsky et al., 2008a], is that one may use the posterior of this ranking to learn additional information about objects and

parts being detected that may be used in subsequent application of the same method (this time empowered by the additional information) to produce better result. Naturally, the better the posterior the better results may be expected. As our method has quite accurate results in many cases even without employing this technique (as illustrated above), it is reasonable to assume its posterior estimates will be quite accurate. Consequently, we feel that integrating these ideas into the method above can make it even more accurate.

Currently, the "limb detection by parallel segment hypotheses" option of the body part detection algorithm provides many possible limb hypotheses that need to be filtered to detect the correct ones. The proposed basic filtering method does not provide sufficiently robust results. Nevertheless, checking against manually labeled ground truth on 300 frames of one of the videos, 97% of the frames had a correct limb candidate option among the hypotheses. One possible way around this issue is using a classifier for filtering correct hypotheses from incorrect ones. Preliminary results with a 90 feature SVM classifier trained for this task show that <2% EER performance is possible and also show acceptable ranking of the hypotheses (the first is almost always the correct one). However, further tests and experiments are needed to properly investigate this point.

Local Contextual Segmentation (LCS)

As commented above, between successive stages in the sequential interpretation scheme there may be an intervening stage of guided image process strongly relying on the output of the previous stages. The LCS is an example of a useful process of this type. The LCS is an iterative local process performing foreground / background segmentation to compute a mask for the object or part of interest. The process of LCS and assumptions employed are explained below.

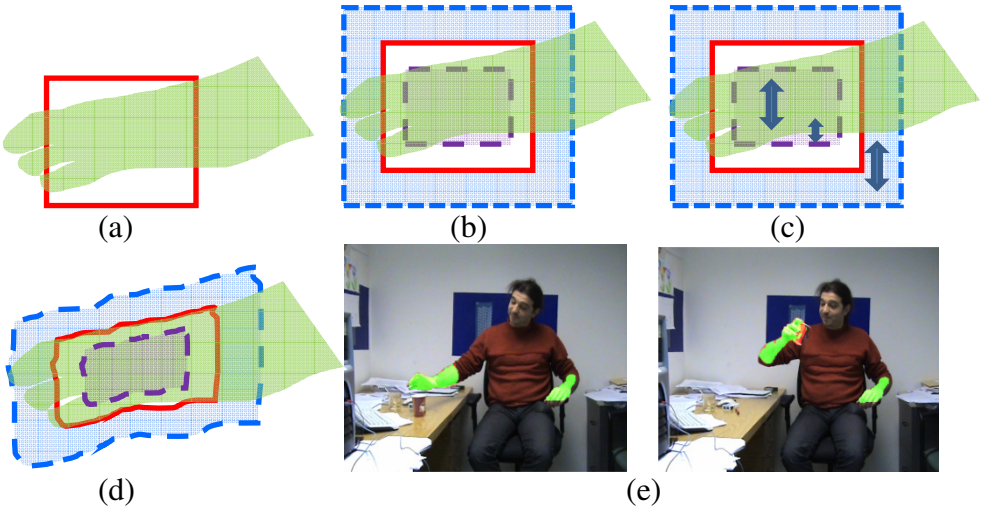


Figure A.5: (a-d) Illustration of different LCS steps, red – current ROI, purple – foreground texture training set, blue – background texture training set; (e) Examples of LCS output for hand and hand-with-glass local segmentation.

1. Assume a region of interest with a foreground object we wish to segment occupying the central area (but not necessarily entirely inside the original ROI), figure A.5a.
2. Hypothesize external and internal regions for the object of interest and approximate background and foreground distributions: (optimized) Parzen

window model of pixels / patches / general descriptors, figure A.5b. The Parzen distributions are computed over histograms of cluster similarity computed for the respective ROIs. The clusters themselves are pre-computed from one or several frames of the video using k-means clustering.

3. Apply the *graph-cut* segmentation [Boykov et al., 2001] with association term provided by the likelihood to be generated by either of the distributions and edges weighted by the gradient strength, figure A.5c.
4. Iterate the process, figure A.5d.

Appendix B: Using body interpretation for dynamic action recognition

After the body parts have been detected, we can use them to extend the interpretation to objects that the person interacts with. For example, we would like to detect some basic 'events', such as whether the hand holds some object or not, or whether an object was added/deleted in a region visited by the hand. These types of hand-object interaction events can be seen as basic building blocks for many different types of actions. In this case we use temporal information. Below is a description of the algorithm we use in order to (sequentially) identify dynamic hand-object interaction events in a video. The algorithm is fairly simple due to the constraints imposed by already detected body parts.

Algorithm outline:

- Apply the body parts detection algorithm (described in Appendix A) to detect body parts masks in every frame.
- Break the frame into cells of fixed size (they may be overlapping). We use 40x40 size cells.
- For each cell record its state (pixels / descriptor) before being visited by the hand and after it left.
- Detect changes by adaptive differencing (ignore changes due to other body parts). The difference of two cells C_1 and C_2 is computed as: $|C_1 - C_2| > t_1 \cdot STD_{noise}$ where the noise model STD_{noise} is computed by an iterative process performing n iterations of throwing away (from the computation) pixels with $|C_1 - C_2| > t_2 \cdot STD(|C_1 - C_2|)$. The parameters n, t_1 and t_2 are all set to 5 in our experiments.
- Track change masks to verify proper hand contact and provide improved annotation of the event that includes the relevant object mask in all frames of interest. We use "local segmentation tracker" that applies LCS process (described in Appendix A) initially on the detected change mask and later on every frame of interest when the initial appearance model and the ROI for the LCS is determined by the previous frame.

Sample results of the above approach for detecting object pick-up and drop-off actions are shown in figure 7.

Unsupervised Classification and Part Localization by Consistency Amplification

Leonid Karlinsky, Michael Dinerstein, Dan Levi, and Shimon Ullman
{leonid.karlinsky,michael.dinerstein,dan.levi,shimon.ullman}@weizmann.ac.il

Weizmann Institute of Science, Rehovot 76100, Israel

Abstract. We present a novel method for unsupervised classification, including the discovery of a new category and precise object and part localization. Given a set of unlabelled images, some of which contain an object of an unknown category, with unknown location and unknown size relative to the background, the method automatically identifies the images that contain the objects, localizes them and their parts, and reliably learns their appearance and geometry for subsequent classification. Current unsupervised methods construct classifiers based on a fixed set of initial features. Instead, we propose a new approach which iteratively extracts new features and re-learns the induced classifier, improving class vs. non-class separation at each iteration. We develop two main tools that allow this iterative combined search. The first is a novel star-like model capable of learning a geometric class representation in the unsupervised setting. The second is learning of "part specific features" that are optimized for parts detection, and which optimally combine different part appearances discovered in the training examples. These novel aspects lead to precise part localization and to improvement in overall classification performance compared with previous methods. We applied our method to multiple object classes from Caltech-101, UIUC and a sub-classification problem from PASCAL. The obtained results are comparable to state-of-the-art supervised classification techniques and superior to state-of-the-art unsupervised approaches previously applied to the same image sets.

1 Introduction

The goal of this paper is unsupervised classification, including discovery of a new category, learning a model of geometric arrangement of object parts and their appearance, and obtaining object and part localization, from a set of unlabeled images, which contains non-class images mixed with some unknown (usually small) percent of class images. The class instances may be uncropped, unaligned and of small size relative to the background.

The problem of unsupervised object classification has gained considerable recent interest [1–12], however, this task is still far from being completely solved. In this study we present a novel methodology to approach the problem. A common approach is to start from some limited, manageable set of initial features

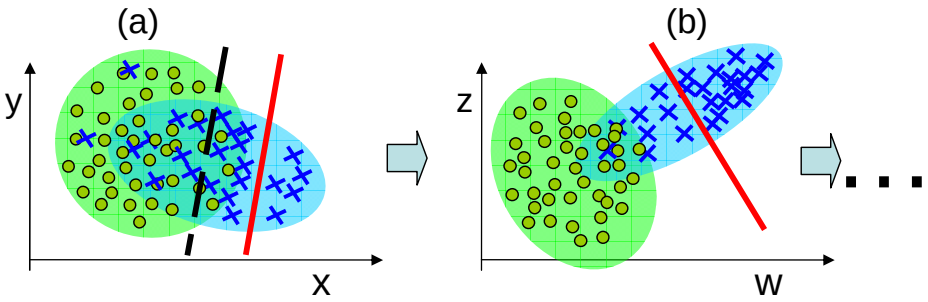


Fig. 1. Illustration of the feature re-extraction approach. (a) In the initial feature space (x-y) it is difficult to separate class (blue crosses) and non class (green circles) examples. In this feature space, the best separating hyperplane (which the unsupervised classification seeks to determine) is marked by the dashed line. Instead, our method identifies a subset of sure class examples separated from the rest (red solid line). (b) Using these examples, the method extracts a new feature set (z-w), in which a larger set of class examples can be identified. The process then continues iteratively. Each iteration uses new features, rather than previous features (or their combinations).

$\mathcal{F}$, for example, a set of local descriptors extracted around image interest points or clusters extracted from such descriptors [1, 2, 11–16, 5–9]. The set of features can be optimized by selecting a subset of the most useful features $\mathcal{F}_1 \subset \mathcal{F}$, or sometimes combinations of features in $\mathcal{F}_1$ are used as new features [2, 8, 14]. However, there is no guarantee that the choice of initial features will in general be sufficient for complete separation. In contrast, we approach the problem as a combined iterative search for features and a classifier. We do not use the initial feature set to obtain the final class separation, but only for identifying a subset of sure class examples which can be reliably separated from the rest (Fig. 1a). This goal is achieved by unsupervised training of a classifier that combines both appearance and part-geometry information. The extracted class examples are used to guide the subsequent extraction of new features, which were not a part of the initial feature set. It is not a-priori clear that this iterative approach will continue to improve classification: if the intermediate classification results are partly incorrect, their use could lead the process astray and cause deteriorating performance. In this work, we demonstrate that in the proposed algorithm, the constructed features become more class-specific as the computation evolves, and the class vs. non-class separation continuously improves (Fig. 1b), reaching a final high level of performance even compared with recent supervised methods.

We develop two main tools that allow the iterative combined search. One is the incremental discovery of part specific features, which combine different part appearances discovered in the training examples. The other is a novel star-like class-geometry model of object parts, which differs from the similar past models [3–5, 15–17, 13] and which can be learned efficiently without supervision in very noisy conditions. These two aspects are described briefly below, and explained in more detail in Sections 2.1 and 2.2.

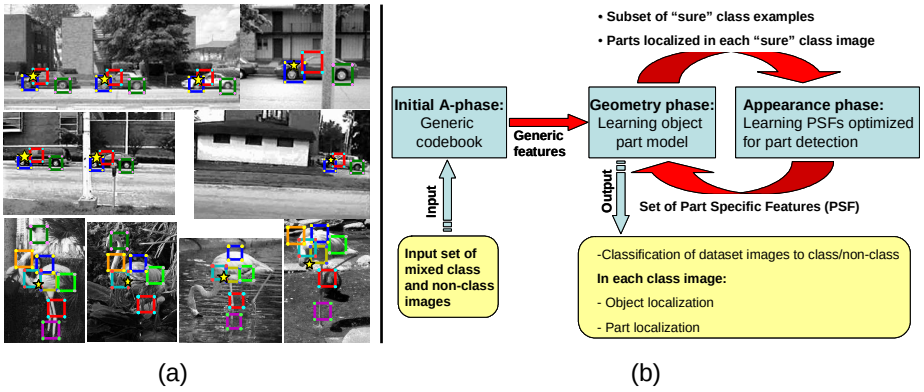


Fig. 2. (a) Example results of unsupervised object and part localization on two datasets (UIUC cars, flamingo). The yellow star is the detected model center location (see text), color coded rectangles are examples of detected object parts (for each object several out of about 150 modeled parts are shown). (b) Schematic diagram of the UCA algorithm.

Feature learning: most unsupervised approaches [1, 2, 5–9, 11, 12], including ours, start from some generic set of features $\mathcal{F}$. During learning, when a particular class is considered, the approaches select a subset $\mathcal{F}_1 \subset \mathcal{F}$ of so-called Class Specific Features (CSF), which coincide better with the class compared with the background or other classes. In contrast, our method extracts and learns a new set of features, termed Part Specific Features (PSF). The PSF are optimized to have higher detection scores at specific locations on the class objects, and at the same time to have lower scores at incorrect locations on the same objects and in non-object detections. Different part specific features have been used successfully in a number of supervised approaches, such as k-fan [17] and semantic hierarchy [18], and were shown to be useful for both object and part localization. Constructing such features in an unsupervised manner is challenging; our method is the first unsupervised method that learns and uses such features, resulting in improved object and part detection and localization.

Geometry learning: Past supervised and unsupervised classification methods can be categorized by their modeling of object geometry. In bag-of-feature methods [2, 9], geometry is ignored. Methods, such as [2, 8, 1, 14], extend the bag-of feature approach by using feature combinations. In [3–5, 11, 13, 15–17] object geometry is modeled by the spatial distribution of each feature in the object reference frame. A geometric part model is useful for classification, but it is challenging to construct such a model in an unsupervised setting. Most previous unsupervised methods therefore do not use a full geometric model [2, 6–9, 11, 12]. Our method uses star-like geometry. It has several differences compared with similar past models. The method is not restricted by a small number of parts as in [3, 4], unlike [13, 15–17] it does not require any supervision, unlike [3, 5, 15, 11] it models distribution of feature locations on the background, unlike [5, 12] it does not rely on non-geometric pLSA [2] for internal supervision, and

unlike [10, 6, 7], it is not based on prior image segmentation. These differences are explained in more detail in Sections 2 and 2.1.

In terms of class vs. background classification performance, our method outperforms the state-of-the-art unsupervised methods [2, 5, 7, 10–12] on 18 classes from the Caltech 101, Weizmann horses and UIUC cars datasets. Surprisingly, the method is also comparable in performance to existing state-of-the-art supervised (and weakly supervised) methods applied to the same datasets. We further demonstrate how our method can be used to separate different object views on the cars class from the PASCAL challenge 2007 dataset. As the method achieves precise object and part localization, it provides a basis for top-down segmentation, as illustrated in supplementary material.

The rest of the paper is organized as follows. Section 2 presents an overview followed by a detailed description of each of the method stages. Section 3 presents results obtained on various datasets, together with an analysis and comparison with previously reported results. Conclusions are discussed in Section 4.

2 The Consistency Amplification Method

Our approach alternates between model learning and data partitioning. Given an image set S, an initial model (learned using initial features) is used to induce an initial partitioning by identifying highly likely class members. The initial partitioning is then used to improve both the appearance and geometrical aspects of the model, and the process is iterated. In this manner the process exploits intermediate classification results at a given stage to guide the next stage. Each stage leads to an improved consistency between the detected features and the model, which is why the process is termed Unsupervised Consistency Amplification (UCA). Each UCA iteration consists of two phases of learning: the feature learning Appearance-phase (A-phase) followed by the part model learning Geometry-phase (G-phase), explained in detail in sections 2.2 and 2.1 respectively. The approach and the order of the phases are summarized in Fig. 2b.

Initial Appearance-phase: In all our experiments, we use a generic codebook of SIFT descriptors of 40×40 patches for the initial (appearance) features. This codebook, denoted by $\mathcal{F}_0$, is computed by a standard technique [19] from all the images in given set S. The codebook descriptors are compared to the descriptors at all points of all the images in S and storing the points of maximal similarity (either one or several, see below) in each image.

Geometry-phase: The detection of parts using the generic features is usually noisy, due to detections in non-class images, and at some incorrect locations in the class images. The goal of the geometric part model learning is to distinguish between the correct and incorrect detections, based on consistent geometric relations between features. This is accomplished by the G-phase of the algorithm, which is also used for the selection of the most useful features and the automatic assignment of each of their detections in every image in S to either object or background model. In contrast with [3–5, 15, 11] that use uniform distribution of features on the background, we model the background by a distribution of the

same family as the class object distribution, which allows to prevent the spurious geometric background consistency from being accounted for by the learned class model. In our experiments we found that modeling the background distribution is better than assuming uniformity with mean performance gain of $12 \pm 7\%$ EER in the first iteration of the UCA that uses initial generic appearance features. The learned background model is then discarded after the learning and is not used for classifying new images. Thus, the model used in the G-phase is a mixture of two stars, one for object and the other for background. It is learned without supervision from all the images in S using a novel graphical model formulation explained in detail in Section 2.1. After the geometric structure has been learned, a subset $H \subset S$ of images which contain class objects with high confidence is selected. In these images the object centers and parts are localized. Unlike [5, 12] that learn the geometric constraints using only a set of objects identified by the non-geometric pLSA [2], our method identifies and localizes objects, and learns their part geometry, jointly and explicitly from the entire data.

Appearance-phase: Each part-specific feature constructed in the A-phase represents an object part by extracting several typical appearance patches of the part, from different images. Part-patches can be extracted, because the locations of the parts in the images of the subset H are already estimated from the previous G-phase. An optimal subset of these part patches is learned by a discriminative model described in section 2.2. The set of all part specific features extracted during the A-phase is denoted by $\mathcal{F}$.

Computing the output: After the G-phase at each iteration, the learned model is applied to produce classification, as well as object and part localization results for either the given dataset or an unseen test set. This is done without introducing any supervision to the system. The way we apply the learned model to test images is described in detail in section 2.3.

2.1 The Geometry Phase

We first describe the G-phase model, and then explain how it is learned from the data. The main goal of the G-phase is to identify the most likely locations of objects and their parts in all images of the given set S and to estimate a subset $H \subset S$ of images which contain class objects with high confidence. The G-phase models the data by a generative probabilistic graphical model depicted in Fig. 3a. Let the image set S have N unlabelled images: $S = \{I_1, I_2, \dots I_N\}$ and the current feature set $\mathcal{F}$ consist of M features: $\mathcal{F} = \{F_1, F_2, \dots F_M\}$. In the G-phase of the first UCA iteration these features are a codebook of generic SIFT descriptors $\mathcal{F}_0$, and in the following UCA iterations these are the learned PSFs. During the G-phase each feature is associated with an object part or the background. Denote the detected location of feature F_m in image I_n by X_m^n (the G-phase uses a single (maximal) detected location per feature in each image, see extension below.) The G-phase model independently generates observed samples: $Data = \{ (F_m, I_n, X_m^n) \,|1 \le n \le N, 1 \le m \le M \}$ The probability of observing a specific image $\Pr(I = I_n)$ is taken to be uniform. The overall observed data likelihood under the G-phase model can be written as:

$$\Pr(Data) \propto \prod_{n=1}^{N} \prod_{m=1}^{M} \sum_{C_m^n=1}^{2} \int_{L_m^n} \Pr(C_m^n \,|I_n\,) \Pr(F = F_m \,|C_m^n\,) \\ \Pr(L_m^n \,|I_n, C_m^n\,) \Pr(L_F = X_m^n \,|F_m, C_m^n, L_m^n\,) dL_m^n \tag{1}$$

The meaning of the product inside the integral in eq. 1 is that each data sample (F_m, I_n, X_m^n) observed in image I_n for the feature F_m is independently generated as follows. First, the latent discrete binary "class" variable C_m^n is drawn with probability $\Pr(C_m^n = k\,|I_n\,) = \alpha_k^n$, independent of the feature F_m. $C_m^n = 1$ means that I_n contains a class object and F_m is generated from the class model. $C_m^n = 2$ means that F_m is generated from the background model, because either I_n does not contain an object or F_m was not detected consistently with the class model. After learning, the value α_k^n is the likelihood of class k (either object or background) in image I_n. Next, the latent location variable L_m^n is drawn from a Gaussian distribution $\Pr(L_m^n\,|I_n, C_m^n = k\,) = N(\mu_k^n, \Sigma_k^n)$. L_m^n represents the image position of the center of the star model (chosen by C_m^n), which generates the feature F_m in image I_n. Note that for every feature detected in image I_n that has chosen the class k, there is a separate variable L_m^n, but all of these variables are generated from the same distribution specific to I_n. Next, the observed feature variable F draws its value F_m from the distribution $\Pr(F = F_m\,|C_m^n = k) = \beta_k^m$ which depends on the chosen class k, but is independent of the image I_n. After learning, the value β_k^m is the likelihood of feature F_m to be consistent with the geometric model of class k. Finally, the observed feature location variable L_F draws its value X_m^n from a linear Gaussian distribution $\Pr(L_F = X_m^n\,|F_m, C_m^n = k, L_m^n\,) = N(L_m^n + \rho_k^m, \Lambda_k^m)$. This distribution models the uncertainty of the offset ρ_k^m of the feature F_m from the L_m^n - center of the star model chosen by C_m^n. It is specific to the feature F_m and the chosen class k and is independent of the specific image I_n.

To summarize, the parameters of the model are α, β, μ, Σ, ρ and Λ, all of them are learned by soft EM as described further below. A schematic drawing illustrating the data generation process and the meaning of the main model parameters is shown in Fig. 3b. The model uses a star-like geometry, but an important difference between the current model and past star model formulations is worth noting. In contrast with [17, 16, 15], that have a single reference point or k-fan per image, in our model there exists a separate reference point (center) random variable for each part, drawn, however, from the same distribution specific to the given image. This allows the features detected in the same image to be updated individually: features assigned to the class update the class star and features assigned to the background update the background star, both the assignments and the updates are soft. Although it may sound technical, it has fundamental importance, since, as we saw in our experiments, in different class images, different subsets of features are geometrically consistent with the object model. It is interesting to note that the transition from the standard star-model to our version is entirely analogous to the transition from Naive Bayes (NB) to pLSA. In the NB there is only a single class node generating the entire feature vector of an image, while in pLSA each feature has a separate topic node gener-

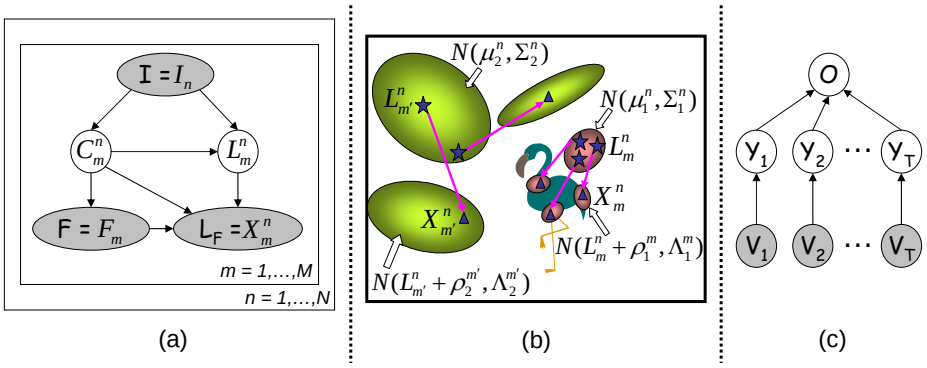


Fig. 3. The probabilistic models used by UCA. Shaded ellipses are observed variables, unfilled are hidden (latent). (a) Graphical representation of the G-phase generative model. (b) Generating the object and background. The object model illustrated in red, the background in green. Each generates centers, denoted by ★ (star) and features denoted by ▲ (triangle). The ellipses denote the uncertainty in position. X_m^n denotes the detected location of feature F_m. Every feature F_m detected on the object is generated using its own star center point L_m^n, but all these L_m^n are generated from the same distribution $N(\mu_1^n, \Sigma_1^n)$ specific to the image. As illustrated, the learned object distributions are tighter than the background distributions. (c) Graphical representation of the "Continuous Noisy OR" discriminative model.

ated from an image specific distribution. The pLSA is more flexible than NB and was found useful for unsupervised classification [2, 9]. Similarly we found that the modified star is useful in modeling feature geometry in the unsupervised setting.

Learning: The model is learned from the data using the soft EM algorithm. The EM update equations are provided in the supplementary material. As mentioned above, the data samples fitted by our model are of the form (F_m, I_n, X_m^n). In order to incorporate the features' detection scores into the learning process, we weight each sample by its score. Namely, the sample (F_m, I_n, X_m^n) is weighted by R_m^n - the similarity of F_m with I_n at location X_m^n. The parameters of the model: α, β, μ, ρ and Λ, are initialized at random and EM is run until convergence. In order to have the object model learned with respect to $C_m^n = 1$, in all our experiments, we initialize $|\Sigma_1^n| << |\Sigma_2^n|$ for every n. During EM iterations, this initialization causes the object feature detections to tend to update the $C_m^n = 1$ model, since they usually appear in more tight and repeatable configurations (i.e. fit a star with smaller center uncertainty Σ_1^n). At the same time, background feature detections, that are usually loosely scattered all over the image, will tend to coincide with the $C_m^n = 2$ model. This method of the initialization of all the parameters (including Σ) was identical throughout all our experiments.

Identifying the set of high-likelihood images: After the EM converges, the value of α_1^n (the image probabilities to belong to the class) shows a strong separation between a subset of class images and all the non-class images, see figure

6a. As a result, by the end of G-phase it becomes possible to identify a subset of high-likelihood class candidate images H. In all our experiments, we marked an image I_n as high likelihood class candidate if $\alpha_1^n > \eta \cdot \max_n(\alpha_1^n)$, where $\eta = 0.85$ was chosen empirically and used throughout all the experiments. Examples of objects automatically identified and localized by the G-phase of the first UCA iteration are shown in Fig. 6b. These are examples of the first G-phase output, obtained using the initial generic features. As can be seen, the localized object model centers appear at similar locations within the object in the different images. In the A-phase, these points are used to extract stacks of corresponding fragments, which are used to construct the part specific features - the CNOR part-detectors, as explained in the next section.

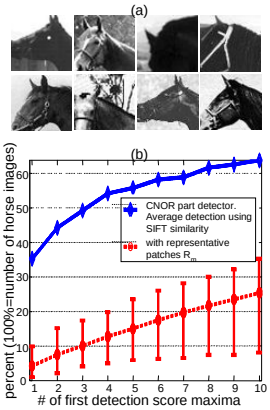
2.2 The Appearance Phase

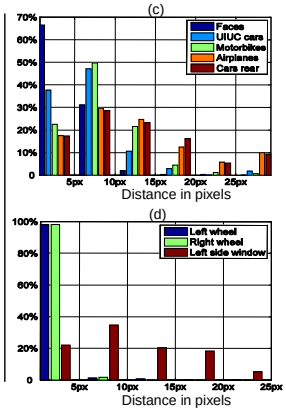
In the G-phase we learned the position of each part relative to the object model center, and detected this center in the images belonging to H. We localize each part in these images by assuming it is located at the learned relative position ρ_1^m from the center located at μ_1^n. In the A-phase we learn for each part a detector trained to distinguish image patches in correct part locations from patches in incorrect ones. The detector is trained using the detected part locations as positive examples and all other locations on the images of H as negative examples. The constructed part detectors form the new feature set $\mathcal{F}$ for the G-phase of subsequent UCA iteration. We next describe the novel probabilistic discriminative model used by the part detector, the Continuous Noisy OR (CNOR), and how this model is trained.

For each part m, corresponding to F_m above, we extract a set of appearances in the following way. In each class candidate image $I_n \in H$, we take the 40x40 image patch at position $\mu_1^n + \rho_1^m$, where μ_1^n is the location of the learned object center in I_n and ρ_1^m is the learned offset of part m from the object center. The accumulated set of image patches is the candidate set of part appearances: $A_m = \{Z_1^m, \ldots, Z_T^m\}$. The next step is to select a subset of appearance representatives $R_m \subseteq A_m$, and learn to optimally combine their detection evidence in order to reliably detect the object part. Both tasks are achieved simultaneously by training the CNOR model, depicted in Fig.3c. Let P be an arbitrary image patch taken from an arbitrary location L in a new image. The binary variable O^P is set to $O^P = 1$ iff L and P are the location and appearance of part m respectively. The probability of O^P is discriminatively modeled as:

$$\Pr(O^P|V;\Theta) = \sum_Y \Pr(O|Y) \cdot \prod_{t=1}^{T} \Pr(Y_t|V_t;\theta_t,R_m) \tag{2}$$

The $\Theta = \{R_m, \theta_1, \ldots, \theta_T\}$ are the learned parameters of the model. $V = \{V_t\}$, where V_t is the output of a continuous SIFT similarity measure between P and $Z_t \in A_m$. Note that we do not explicitly model $\Pr(V)$, which can be a complex distribution. $Y = \{Y_t\}$, where Y_t is a latent binary variable representing the detection of appearance Z_t with:





	Object localization error STD, % of object size
Faces	2.6
UIUC cars	4.6
Motor-bikes	5
Air-planes	8
Cars rear	8

Fig. 4. (I) Example of a CNOR part detector. (a) Selected representative appearances. (b) Cumulative histograms of detections within 15px from ground truth location relative to the number of detection score local maxima used. (II) Evaluation of object and part localizations obtained by the UCA method showing a distribution of localization error in pixels relative to manually marked ground truth. 10px is less then 7% of object size in all the sets in the figure. (c) Object localization, 100% = number of class images. (d) Part localization on UIUC cars, each part was detected in about 95% of objects. In the graph 100% = number of part detections.

$$\Pr(Y_t = 1 | V_t; \theta_t, R_m) = \begin{cases} \frac{1}{1+e^{-\alpha_t(V_t-\tau_t)}} & Z_t \in R_m \\ 0 & Z_t \in A_m \backslash R_m \end{cases} \tag{3}$$

Here $\theta_t = \{\tau_t, \alpha_t\}$ are the parameters of the sigmoid in 3. $Y_t = 1$ becomes likely if patch P exceeds a similarity threshold τ_t with the representative patch $Z_t \in R_m$, with α_t representing the uncertainty of τ_t. If $Z_t \in A_m \backslash R_m$ (meaning Z_t is not a chosen representative) then V_t and Y_t have no effect on $\Pr(O^P | V; \Theta)$. Finally, $\Pr(O^P | Y)$ is a deterministic "or" of Y:

$$\Pr(O^P = 1 | Y) = \begin{cases} 1 & \exists t. Y_t = 1 \\ 0 & otherwise \end{cases} \tag{4}$$

The entire model can intuitively be described as follows: the part is detected ($O_p = 1$) whenever P is "sufficiently" similar to at least one of the part m's representative patches in R_m.

To learn the model parameters, a training set of image patches E is constructed by taking all 40×40 image patches (on a fixed step grid) from all the images in the current class candidate set H. For each patch $P \in E$ the observed data vector is constructed as: $D^P = \langle V^P, O^P \rangle$ where $V^P = \{V_t^P\}$ is computed by measuring similarity between P and A_m patches and $O^P = 1$ iff $P \in A_m$ (and $O^P = 0$ otherwise). Finally the training data for the CNOR model of part m is: $D = \{D^P | P \in E\}$. By treating the correct part appearances (A_m) as positive examples and all other appearances (either other parts of the object or back-

ground patches) as negative examples, the object part detector is trained for correct localization of the part. To limit the number of representatives, the learning objective is to find the Minimum Description Length (MDL) parameters Θ, in other words, to find Θ that maximize a combined score of the model complexity (number of representatives) and model performance (data likelihood). We solve this learning problem using the Structural EM (SEM) algorithm optimizing the Bayesian Information Criterion (BIC) score [20]:

$$BIC = \sum_{P \in E} \log(\Pr(O^P|V^P;\Theta)) - \frac{\log T}{2} \cdot |R_m| \tag{5}$$

The SEM algorithm iterates between two stages. The first stage is, given a set of representatives R_m, to find the optimal values for the $\{\theta_t\}$ parameters. This stage is solved using the EM algorithm. It is computed efficiently, since in our model each EM iteration has linear time complexity. The second stage is, given the current assignment of $\{\theta_t\}$, to estimate an improved R_m. This is achieved by running several iterations of a greedy search over subsets of $R_m \subseteq A_m$, where at every step of the search a current subset R_m is modified by either adding or removing one element.

After learning, the set of part m detections is obtained by identifying first few local maxima of the probability $\Pr(O^P|V;\Theta)$ computed for all patches P in a given image. Selected representatives for an example part are shown in Fig. 4a. The resulting CNOR part detectors for all parts m, are a significantly more reliable set of object features then the initial generic set of features, as demonstrated in Fig. 4b and in section 3, and it provides a general method for reliable part detection for both supervised and unsupervised classification.

2.3 Applying the learned model to classify and localize objects and parts

To compute the classification score for an unseen test image, and to localize the class objects in it, we use the learned ρ_1^m (offsets from object star center) and Λ_1^m (STDs for these offsets) parameters in a voting scheme similar to [13] as follows. For each part detector, a number (five in our experiments) of highest-scoring locations at each image are marked. To identify the object star's center location, each detection X votes for a center location, by placing a Gaussian mask with STD Λ_1^m around the expected location $X - \rho_1^m$. After all the detectors voted, the point with the maximal accumulated vote determines the location of the object star's center in each image (in case there are multiple objects, several local maxima that exceed a global threshold are taken). The accumulated vote value at the detected center point serves as the object detection score in the image. These scores are then used to create the ROC that tests the separation between the class and the non-class images in the results section 3. The object localization results of our method are evaluated in Fig. 4c. The parts are localized by "back-projection" as in [13]. Each part detector that voted into one of the selected object center locations (with one of its five detections) is declared as

'detected' and is marked in the image. The accuracy of our part localization is demonstrated in figures 2a and 5 and evaluated in Fig. 4d. Marking all the detected parts in the image can be used for a top-down segmentation of the detected object (see examples in supplementary material). The details of the top-down segmentation are outside the scope of the current discussion.

3 Results

To test the performance of the UCA method, it was applied to the task of fully unsupervised classification and object and part localization on 18 different object classes. The list of the classes, the parameters of the datasets and ROC EERs obtained by the UCA are summarized in Table 1. The results show that our method obtains superior performance over the existing unsupervised methods in challenging conditions such as small objects relative to the background (e.g. UIUC cars, Caltech101 cars, flamingo), small percent of class images in the set (e.g. schooner, guitars), significant inter-class variability due to non-rigid deformations (e.g. bonsai, horses, crab, flamingo, starfish) and significant lack of alignment (e.g. UIUC cars, faces, PASCAL car views). Examples of the classes and object and part localizations obtained by UCA are shown in figures 2a and 5. Fig. 4c,d shows quantitative evaluation of automatic object and part localization by UCA compared to hand generated ground truth on several dataset. The background images for each dataset were chosen randomly out of Caltech backgrounds set containing 900 images. To challenge our method, we tested it on different class vs. non-class mixes, namely 10%, 20%, 30% and 50%. This is compatible with experimenting with Google data, since manual validation done by [5] showed that on average, above 25% of images returned by Google image search are good examples. For every dataset, increasing percent of class images above the percent reported in the table gives even better results. The UIUC cars dataset contained only the 170 non-cropped and non-aligned test images of the original set (and equal amount of random background images), the training images of the original set are cropped, so to make the task harder they were not used. The Caltech-5 datasets (from [3]) were tested in order to compare with past unsupervised approaches that were tested on the same data, namely [2, 12, 5, 10, 7]. To ensure that the chosen Caltech101 classes are sufficiently hard, 9 of the 11 tested Caltech101 classes are the ones with lowest reported performance by [7] (average of entries for these classes on [7]'s confusion matrix diagonal is 49%). Note that unlike [7], we do not use color information in our scheme. An important characteristic of the UCA is its ability to deal with a low percentage of class images in the dataset. Methods such as [5, 12] that apply the pLSA method of [2] as a pre-processing step to identify and localize class examples may fail on such datasets. This was validated by testing the pLSA method with eight topics (optimal number proposed by [5]) on the schooner dataset that contains only 10% class images. From the N examples with maximal score in the class topic, less then 40% were class examples ($N = 10, 20, \ldots, 100$).

Table 1. Summary of fully unsupervised classification results obtained by the UCA method. For all datasets, the EER STD for UCA was $\leq$ 2% (computed by cross-validation). For motorbikes, airplanes and cars-rear, the class images were randomly chosen from larger sets and remaining images were also used for testing the learned models obtaining 1.3%, 2.3% and 2.8% average EER respectively. The average EER of UCA on the Caltech-5 datasets was 2.65%. Results of other unsupervised methods reported for Caltech-5 were: average EER of 4.08% [12], 7.35% [5] and 11.38% [2] and average multiclass detection rate of 5.4% [7]. Results of leading supervised methods on Caltech-5 are comparable to our unsupervised result: average EER of 2.25% [15] and 1% [21] ([21] did not test on cars-rear class). The object size relative to the background for each dataset was approximated from several characteristic images.

Dataset	Origin	Total number of images	% class images in the set	Object size rel. to bgrnd.	UCA EER, %
Horses	Weizmann	646	50%	35%	2.7
Cars	UIUC	340	50%	3.5%	3.1
Car views	PASCAL	201	50%	40%	10.7
Motorbikes	Caltech-5	900	50%	30%	2.3
Airplanes	Caltech-5	900	50%	20%	2.7
Faces	Caltech-5	900	50%	20%	2.4
Cars-rear	Caltech-5	900	50%	20%	3.2
Bonsai	Caltech101	256	50%	35%	4.1
Ewer	Caltech101	283	30%	34%	2.3

Dataset	Origin	Total number of images	% class images in the set	Object size rel. to bgrnd.	UCA EER, %
Butterfly	Caltech101	303	30%	27%	6
Cars	Caltech101	600	20%	12%	1.1
Crab	Caltech101	365	20%	24%	10.9
Starfish	Caltech101	430	20%	11%	12.6
Laptop	Caltech101	405	20%	32%	4.3
Flamingo	Caltech101	335	20%	13%	7.3
Watch	Caltech101	1139	20%	40%	3.9
Guitars	Caltech101	650	10%	35%	8.2
Schooner	Caltech101	650	10%	27%	6.69

In the PASCAL car views experiment, we tested the ability of UCA to separate related sub-classes. In particular, out of the PASCAL 2007 training images, images depicting frontal and side views of cars were extracted. The scale of the images was normalized by vertical size and large background areas around each car was taken to make the set un-cropped and un-aligned. The UCA was then applied to this set in order to separate the views. Furthermore, when applied on a set of about 700 images containing all the car views, UCA successfully learned the "frontal cars" subclass with similar EER to the two view experiment. Applying pLSA to the same set has yielded high error (32% EER). The ability of our method to separate similar sub-classes and specifically different views of the same class can also be useful in supervised learning applications. If a given training set of images of the same class can be automatically separated into a meaningful set of (inherently similar) subclasses, then it can greatly facilitate the learning task, by allowing the modeling of each subclass separately.

4 Conclusions

The UCA method has a number of basic advantages compared with previous unsupervised classification methods. First, the overall classification results are higher than obtained previously, and remain high even when class examples are sparsely distributed within the dataset. Surprisingly, on the tested classes,

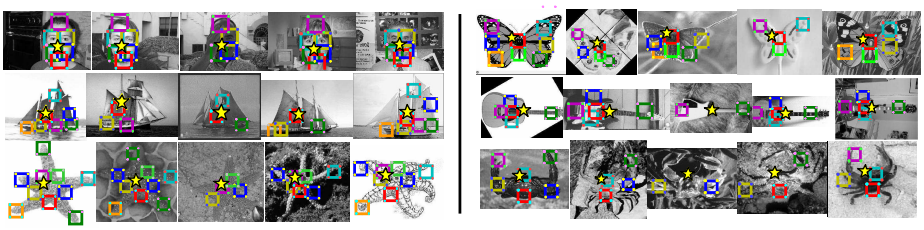


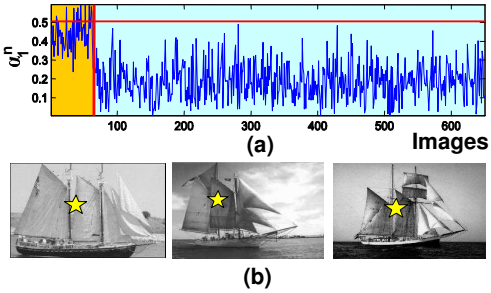
Fig. 5. More examples of unsupervised object and part localizations obtained by the UCA method. See explanation in fig. 2a

results of the unsupervised method are as good as leading supervised methods. Second, the method obtains precise object localization, indicated by a repeatable reference point on each detected object. Third, precise locations of the parts participating in the model are also made available. Fourth, the method is capable of separating similar classes and sub-classes, such as different views of the same class. The main novel aspects of the UCA method are the following. The model is iteratively improved by exploiting intermediate classification results, consistently improving the performance. A novel geometric model is used, which can be efficiently learned from the entire dataset, and therefore improve the methods ability to capture geometric consistencies, even when consistent configurations are sparse. The model uses a part detection scheme, which is trained to detect object parts with diverse appearances in their correct position. The resulting detections are therefore more reliable, providing precise part localization and improved overall performance.

Acknowledgments: This work was supported by EU IST Grant FP6-2005-015803 and ISF Grant 7-0369.

References

1. Fritz, M., Schiele, B.: Towards unsupervised discovery of visual categories. DAGM (2006)
2. Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A., Freeman, W.T.: Discovering objects and their localization in images. ICCV (2005) 370–377
3. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. CVPR (2) (2003) 264–271
4. Fergus, R., Perona, P., Zisserman, A.: A visual category filter for google images. ECCV (2004)
5. Fergus, R., Fei-Fei, L., Perona, P., Zisserman, A.: Learning object categories from google's image search. ICCV (2005) 1816–1823
6. Russell, B.C., Freeman, W.T., Efros, A.A., Sivic, J., Zisserman, A.: Using multiple segmentations to discover objects and their extent in image collections. CVPR (2006)
7. Cao, L., Fei-Fei, L.: Spatially coherent latent topic model for concurrent object segmentation and classification. ICCV (2007)



	Iter. 2 EER %	Iter. 3 EER %	Iter. 4 EER %	Iter. 5 EER %
Starfish	16.3	15.1	12.7	~
Crab	17.8	14.9	12.3	10.9
Flamingo	11.9	7.4	~	~
Guitars	17.4	8.2	~	~
Bonsai	8.6	3.9	~	~

(c)

Fig. 6. Improvements due to consistency amplification. (a) Example of class vs. background separation obtained by the first iteration for the schooners class. Yellow part are the class images and the rest are backgrounds (the ordering is only for illustration purposes). The horizontal line shows the adaptive threshold $\eta \cdot Max$ used to select the set H of high likelihood class examples for the next UCA iteration. (b) Examples of the objects identified and localized. (c) Table of EER improvement with UCA iterations. Only the classes that ran for more then two iterations are shown. The iterations continue until the set H stops growing. The average EER of the first iteration (that used the generic features and not PSFs) was 30% for these classes. This illustrates the low (relative to the PSFs) consistency between the generic features and the class objects.

8. Nowozin, S., Tsuda, K., Uno, T., Kudo, T., Bakir, G.H.: Weighted substructure mining for image analysis. CVPR (2007)
9. Li, L.J., Wang, G., Fei-Fei, L.: Optimol: automatic object picture collection via incremental model learning. CVPR (2007)
10. Ahuja, N., Todorovic, S.: Discovering hierarchical taxonomy of categories and shared subcategories in images. ICCV (2007)
11. Liu, D., Chen, T.: Semantic-shift for unsupervised object detection. CVPR Workshop (2006)
12. Liu, D., T.Chen: Unsupervised image categorization and object localization using topic models and correspondences between images. ICCV (2007)
13. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. ECCV (2004)
14. Quack, T., Ferrari, V., Leibe, B., Gool, L.V.: Efficient mining of frequent and distinctive feature configurations. ICCV (2007)
15. Loeff, N., Arora, H., A.Sorokin, Forsyth, D.: Efficient unsupervised learning for localization and detection in object categories. NIPS (2005)
16. Sudderth, E.B., Torralba, A., Freeman, W.T., Willsky, A.S.: Learning hierarchical models of scenes, objects, and parts. ICCV (2005)
17. Crandall, D.J., Huttenlocher, D.P.: Weakly supervised learning of part-based spatial models for visual object recognition. ECCV (1) (2006) 16–29
18. Epshtein, B., Ullman, S.: Semantic hierarchies for recognizing objects and parts. CVPR (2007)
19. Jurie, F., Triggs, B.: Creating efficient codebooks for visual recognition. ICCV (2005)
20. Friedman, N.: The bayesian structural em algorithm. UAI (1998) 129–138
21. Dorko, G., Schmid, C.: Object class recognition using discriminative local features. INRIA (2005)

Appendix: EM Update equations for the G-phase model

1. G-phase model EM update equations

Following are the auxiliary definitions:

$$\pi_{n,m}^{k} \leftarrow \quad R_m^n \cdot \frac{\alpha_k^n \cdot \beta_k^m \cdot}{\sum_{k=1}^{2} \alpha_k^n \cdot \beta_k^m} \cdot \frac{N(X_m^n; \mu_k^n + \rho_k^m, \Sigma_k^n + \Lambda_k^m)}{N(X_m^n; \mu_k^n + \rho_k^m, \Sigma_k^n + \Lambda_k^m)} \tag{1}$$

Here, we assume for simplicity that both Σ_k^n and Λ_k^m are diagonal matrices and denote by $\Sigma_k^n(j)$ and $\Lambda_k^m(j)$ the j-th element on the diagonal.

$$\tilde{\sigma}_{n,m}^{k}(j) \leftarrow \frac{\Sigma_k^n(j) \cdot \Lambda_k^m(j)}{\Sigma_k^n(j) \cdot \Lambda_k^m(j)} \tag{2}$$

$$\tilde{\mu}_{n,m}^{k} \leftarrow \quad \mu_k^n + \Sigma_k^n \cdot (\Sigma_k^n + \Lambda_k^m)^{-1} \cdot (X_n^m - \mu_k^n - \rho_k^m) \tag{3}$$

Following are the update equations:

$$\alpha_k^n \leftarrow \frac{\sum_{m=1}^{M} \pi_{n,m}^{k}}{\sum_{k=1}^{K} \sum_{m=1}^{M} \pi_{n,m}^{k}} \tag{4}$$

$$\beta_k^m \leftarrow \frac{\sum_{n=1}^{N} \pi_{n,m}^{k}}{\sum_{k=1}^{K} \sum_{n=1}^{N} \pi_{n,m}^{k}} \tag{5}$$

$$\mu_k^n(j) \leftarrow \frac{1}{\sum_{m=1}^{M} \pi_{n,m}^{k}} \cdot \sum_{m=1}^{M} \pi_{n,m}^{k} \cdot \tilde{\mu}_{n,m}^{k}(j) \tag{6}$$

$$\Sigma_k^n(j) \leftarrow \quad -(\mu_n^k(j))^2 + \frac{1}{\sum_{m=1}^{M} \pi_{n,m}^{k}} \cdot \sum_{m=1}^{M} \pi_{n,m}^{k} \cdot [\tilde{\sigma}_{n,m}^{k}(j) + (\tilde{\mu}_{n,m}^{k}(j))^2] \tag{7}$$

$$\rho_k^m(j) \leftarrow \frac{1}{\sum_{n=1}^{N} \pi_{n,m}^{k}} \cdot \sum_{n=1}^{N} \pi_{n,m}^{k} \cdot [X_m^n(j) - \tilde{\mu}_{n,m}^{k}(j)] \tag{8}$$

$$\Lambda_k^m(j) \leftarrow \quad -(\mu_n^k(j))^2 + \frac{1}{\sum_{n=1}^{N} \pi_{n,m}^{k}} \cdot \sum_{n=1}^{N} \pi_{n,m}^{k} \cdot [(X_m^n(j) - \tilde{\mu}_{n,m}^{k}(j))^2 - 2 \cdot (X_m^n(j) - \hat{\mu}_m^k(j)) \cdot (\tilde{\mu}_{n,m}^{k}(j) + \tilde{\sigma}_{n,m}^{k}(j) + (\tilde{\mu}_{n,m}^{k}(j))^2] \tag{9}$$

2. UCA as a basis for top-down segmentation

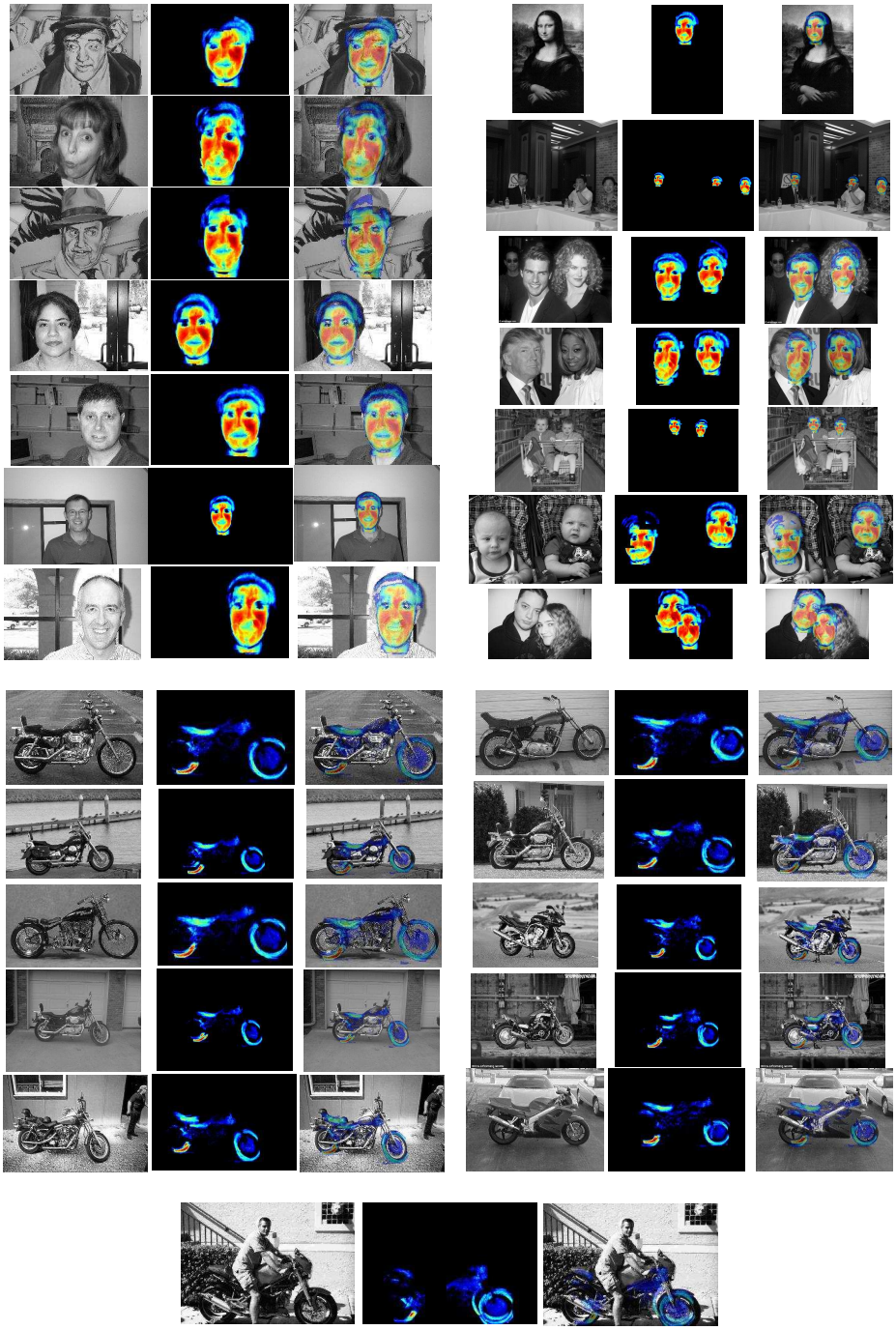
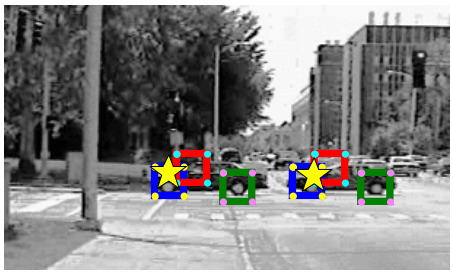
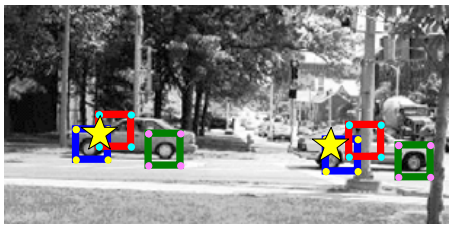
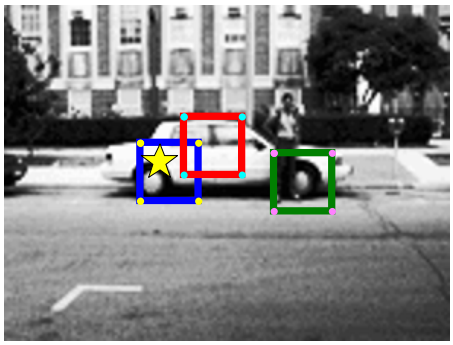
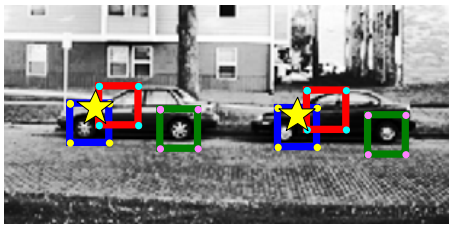
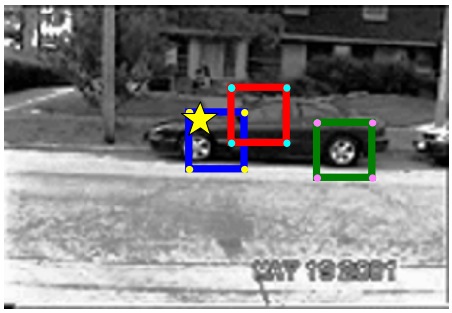
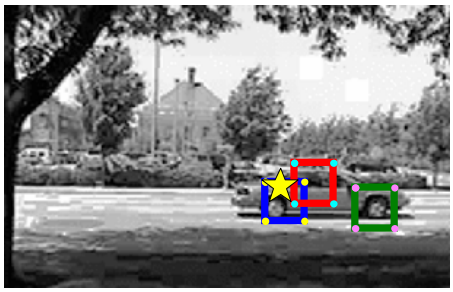
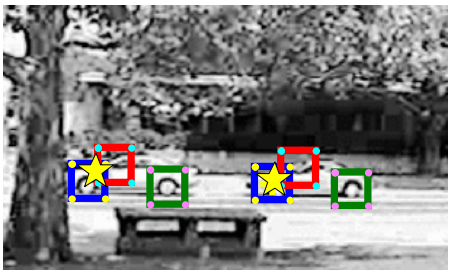
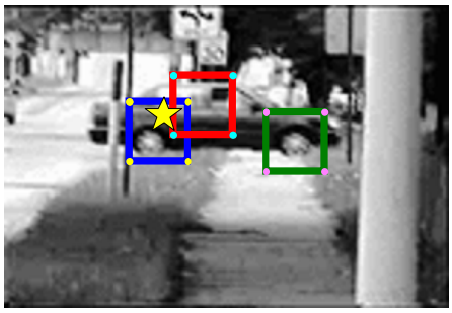


Figure 1. Examples of top-down segmentations produced by back-projecting all the detected CNOR part detector locations consistent with the models learned by UCA. A single global threshold (for each class) was used to produce the depicted segmentation masks. Left column - original images, middle column - computed segmentation masks, right column - overlay.

3. Additional UIUC cars examples

Combined model for detecting, localizing, interpreting and recognizing faces

Leonid Karlinsky, Michael Dinerstein, Dan Levi, and Shimon Ullman
{leonid.karlinsky,michael.dinerstein,dan.levi,shimon.ullman}@weizmann.ac.il

Weizmann Institute of Science, Rehovot 76100, Israel

Abstract. This work describes a method that combines face detection, localization, part interpretation and recognition, and which is capable of learning from very limited data, in a semi-supervised or even fully unsupervised manner. Current state-of-the-art techniques for face detection and recognition are subject to two major limitations: extensive training requirement, often demanding tens of thousands of images, and detecting faces without explicitly detecting relevant facial parts. Both of these limitations hinder the recognition task, since for a specific face the number of available examples is usually small, and because the strongest cues for identity lie in the specific appearance of facial parts. The proposed method alleviates both these limitations by effective learning from a small training set and by detecting the face through, and together with, its main parts. This is obtained by a novel unsupervised training method, which iterates phases of part geometry and part detector learning, to incrementally learn an object category from a set of unlabeled images, containing both class and non-class examples given in unknown order. We tested our method on face detection and localization tasks both in a set of 'real life' images collected from the web as well as in LFW and MIT-CMU databases. We also show promising results of our method when applied to a face recognition task.

1 Introduction

The focus of this work is on a method that combines face detection ('what is it?'), localization ('where is it?'), part interpretation ('where are the facial parts?') and recognition ('who is it?'), and which is capable of learning from very limited data, in a semi-supervised or even fully unsupervised manner. The method was developed for general object detection and localization and in this work we extend and apply it to the task of detecting and localizing faces. In addition, we extend it to perform the recognition task.

The face detection and localization problem has a long history in the literature, but a significant dividing line was the influential work by Viola and Jones [1]. Based on a survey by [2], before [1] the state-of-the-art methods for face detection were divided into several approaches: knowledge-based [3], feature invariant [4], template matching [5], and appearance based [6–8]. Following [1] the focus of face detection approaches turned towards efficient real-time face

detection. In [1], the face is represented by a cascade of linear classifiers each using simple low-level 'Haar features' and trained using boosting methods [9]. Currently, many state-of-the-art face detectors are variants of the approach in [1], with different improvements and extensions. For example, [10] extend the set of Haar features, [11] improve feature computation efficiency and introduce a coarse-to-fine technique, [12] improve the optimization of the cascade, [13] offer an alternative to the boosting technique used in [1] and provide efficient methods for multi-view detection, and [14] suggest an improved set of low-level features and improve the computational efficiency. Currently, the best results were reported by [14].

Despite all the developments listed above, there are still two main limitations shared by current state-of-the-art face detection techniques. These limitations are especially relevant if one wants to extend such techniques to face recognition. The methods based on [1] usually have an extensive training requirement. For instance [13] train on 75,000 face examples, and [14] have around 23,000 faces and 30,000 non-faces in their training set. While general face examples are abundant, this is usually not the case for face recognition tasks, where only a handful of images may be available for a specific individual. The second limitation is non-specificity to facial parts. The Haar-like features based methods detect faces without explicitly detecting facial parts and thus are capable of face detection and localization, but are not suited for recognizing and discriminating between different shapes of face parts such as eyes, nose, ears, mouth, chin and etc. The ability to detect specific facial parts plays a key role in face recognition task, as the distinction between two individuals is often based on fine differences in specific face parts appearance such as types of nose, eyes, hair, etc.

The proposed method addresses both of these limitations. It is capable of learning from a small number of examples, even in the challenging unsupervised setting, in which the object examples appear at unknown locations and only in an unknown subset of the training images. The subset of images containing class examples may even be as small as 10% of the training set. In addition, our method is part-based, meaning that the object is detected through first detecting its parts. The parts and their spatial configuration are learned automatically. Moreover, each part is represented by a (learned) set of its so-called semantic equivalents. For example, the nose part is represented by a set of image fragments representing the nose viewed with varying poses, illuminations, and other types of variations. These semantic equivalents are learned without supervision, and they are combined using an efficient representation (the CNOR model below) to form a robust detector for a corresponding part.

Our method was tested on unconstrained frontal face detection and localization tasks in a challenging set of 'real-life' images, collected randomly from the web. The faces are usually small relative to the images, appear at random locations in the images and exhibit strong scale, lighting, pose and expression variations. The set also contains partially occluded faces (e.g. by sunglasses or other objects) and multiple face instances. We also tested detection performance on the LFW database [15] and detection and localization performance on the

MIT-CMU dataset [16]. Examples of the output of our method on images taken from various datasets appear in figure 7. Finally, we applied our method to the simultaneous object detection, localization and recognition task on a dataset consisting of the set of 'real-life' images mentioned above mixed with an additional set of about 130 images containing a face of a specific person. The person images also exhibit large scale, pose, lighting, location, expression and occlusion variations. Our system showed good performance on this combined detection, localization and recognition task despite being trained on a limited set of only 10 images of the specific individual. Examples of combined person detection and recognition appear at the bottom row of figure 7. In addition, we tested the contribution of various aspects of the proposed method by removing some of them and evaluating the performance on the same detection and localization task.

The rest of the paper is organized as follows. Section 2 describes the method and the learning algorithms used, section 3 provides details of the experimental validation, and section 4 contains the summary and proposes possible future research directions.

2 Method

This section presents our method for combined detection, localization, part interpretation and recognition of human faces. The method builds upon and extends a general method for unsupervised category learning which is presented in a companion paper [17]. The unsupervised learning algorithm, called Unsupervised Consistency Amplification (UCA) is briefly overviewed in section 2.1 and its extensions introduced for face detection and recognition are described in section 2.2. For a more detailed description of the UCA algorithm please refer to [17].

2.1 Unsupervised Consistency Amplification (UCA)

The basic UCA unsupervised training algorithm is described in detail in a companion paper [17], here we give only a brief overview. UCA alternates between model learning and data partitioning. Given an image set S, an initial model (learned using initial generic features) is used to induce an initial partitioning by identifying highly likely class members. The initial partitioning is then used to improve both the appearance and the geometrical aspects of the model, and the process is iterated. In this manner the process exploits intermediate classification results at a given stage to guide the next stage. Each stage leads to an improved consistency between the detected features and the model, which is why the process is termed Unsupervised Consistency Amplification (UCA). Each UCA iteration consists of two phases of learning: the feature learning Appearance-phase (A-phase) followed by the part model learning Geometry-phase (G-phase). The approach and the order of the phases are summarized in Figure 1.
Here we briefly describe the phases of the algorithm:

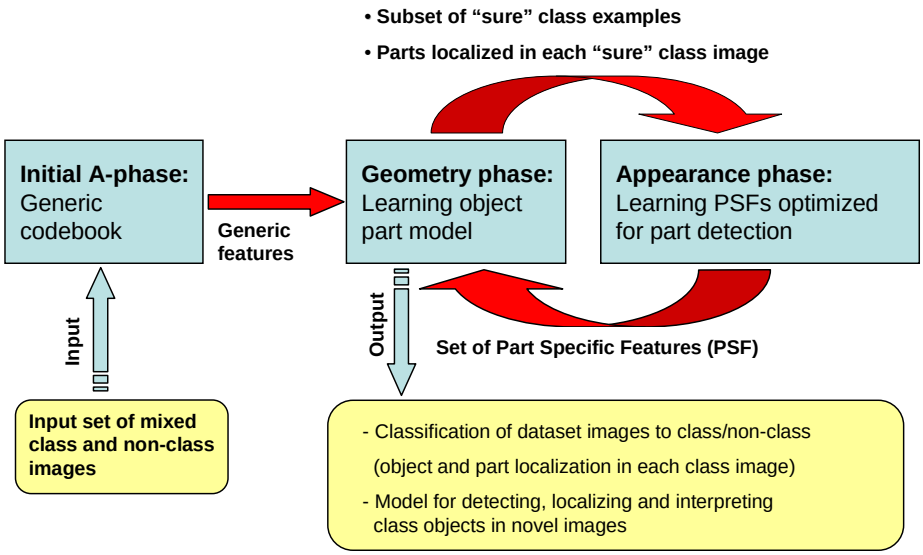


Fig. 1. Schematic diagram of the UCA algorithm

Initial Appearance-phase: We use a generic codebook of quantized SIFT descriptors of 40 x 40 patches for the initial (appearance) features. This codebook is computed by a standard technique [18] from all the images in given set S . The codebook descriptors are compared to the descriptors at all points of all the images in S and storing the points of maximal similarity (either one or several, see below) in each image.

Geometry-phase: The detection of parts using the generic features is usually noisy, due to detections in non-class images, and at some incorrect locations in the class images. The goal of the geometric part model learning is to both distinguish between class and non-class images and between the correct and incorrect part detections, based on consistent geometric relations between the features. This is accomplished by the G-phase of the algorithm, which is also used for the selection of the most useful features and the automatic assignment of each of their detections in every image in S to either object or background model. During training, we model the background by a distribution of the same family as the class object distribution, which allows preventing from the spurious geometric consistency on the background to be accounted for by the learned class model. In our experiments we found that modeling the background distribution is better than assuming it to be uniform. The learned background model is then discarded after the training and is not used for classifying new images. Thus, the generative probabilistic model used in the G-phase is a mixture of two star-models, one for object and the other for background. It is learned without supervision from all the images in S using a novel probabilistic graphical model formulation explained in [17]. After the geometric structure has been learned,

a subset $H \subset S$ of images which contain class objects with high confidence is selected. In these images the object centers and parts are localized.

Appearance-phase: The A-phase constructs the part detectors used as features by the following G-phase. Each part-detector constructed in the A-phase represents an object part by extracting several typical appearance patches of the part, from different images. Part-patches can be extracted, because the locations of the parts in the images of the subset H are already estimated from the previous G-phase. An optimal subset of these part patches is learned by a novel probabilistic discriminative model, the Continuous Noisy OR (CNOR), described in [17]. The model parameters are optimized for part detection in correct location on faces.

Applying the model to new images: As in [17] we apply the model to new images by voting for the object reference point from the locations detected by the learned part detectors. Each part P_i has two learned parameters defining its spatial location relative to the object reference point - offset μ_i and covariance matrix Λ_i defining the uncertainty region of the offset. Both these parameters are used during voting. Following the voting, the parts are detected and localized by back-projection: each part that was detected within one STD from its expected location is declared as 'detected'. In the current paper we extend the voting method used in [17] towards multi-scale and multi-object detection. To handle multiple scales we iterate over candidate horizontal sizes of the face, in our case from 20 to 150 pixels with 10 pixel increments. The search over different scales produces only a few false alarms due to high precision of the UCA classifier (see [17]). We also apply non-maximal suppression to suppress overlapping detections (bounding boxes) with lower scores. To handle multiple objects we replace the voting method used in [17]. In [17] only five highest scoring detections of each part were used in the voting. Since an image may contain more then five faces (as is the case in many of our examples), we replaced this method by voting using the entire set of part detections for each part. Given an image, detector for part P_i is applied to a grid of locations on the image (we used regular grid with a step of 5 pixels) producing a response map R_i , then R_i is shifted by μ_i , and dilated with Λ_i and added to a cumulative voting mask. The local maxima of the voting mask after all part detectors have voted constitute the candidate locations of the faces. Applying non-maximal suppression on voting masks gathered from all the candidate scales and thresholding the result produces the final face detections.

The training of the G-phase and A-phase probabilistic models was achieved by applying Expectation Maximization (EM) algorithm [19] and its structure learning variant [20]. The following section explains how UCA was used in faces detection, localization, part interpretation and recognition tasks.

2.2 UCA-based face detection and recognition

The model for face detection (for brevity detection = detection + localization + part interpretation) and recognition is trained in three stages, each stage training a separate UCA model. In the first stage, M_1 - a model for detection is trained on

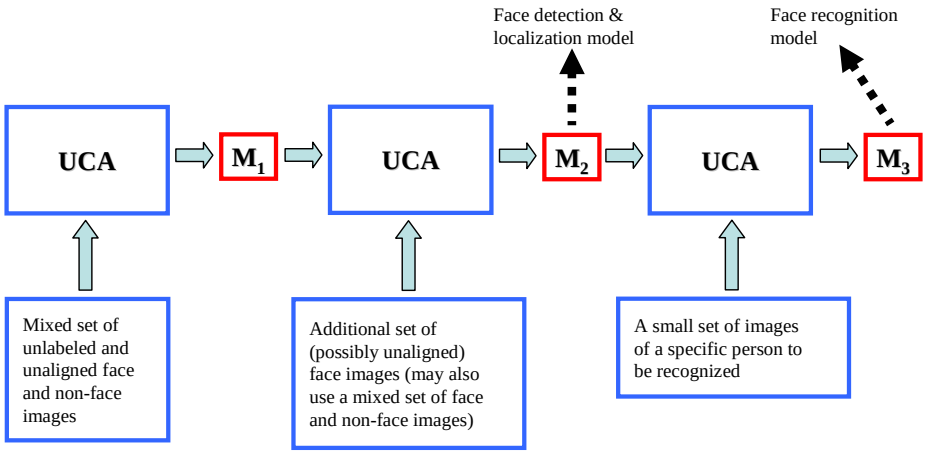


Fig. 2. A schematic illustration of the proposed approach showing the inputs and the outputs of all the training stages

an unlabeled set of mixed face and non-face images in an entirely unsupervised manner. In the second stage an improved model for detection M_2 is trained on an additional training set, in which the faces are detected using M_1. In the third stage, a model for recognition M_3 is trained using several examples of a specific face to be recognized detected and localized, based on the application of M_2.

Stage 1: The first stage is fully unsupervised, its training set consisting of mixed non-face and face images (faces appear at unknown locations). This stage was performed in the experiments of [17] and here we use the model it learned (denoted M_1) for the second stage. Fifty face examples from Caltech faces dataset were present in the unsupervised training set of the first stage (together with 450 non-face images). In general this stage may be used in cases we train on a semi-supervised set (having only images that contain the learned object), but when object examples vary in viewing direction (or other viewing conditions), e.g. if we get a set of mixed frontal and profile faces. Although it is possible to try to capture all the viewing directions by a single model, it is clearly harder and will typically cost in loss of ability to detect facial parts. So in this case it is beneficial to first apply unsupervised learning to separate different views and only later to learn a separate model for each view. Experiments along these lines were performed in [17] for automatically separating different car views.

Stage 2: The second stage is learning an improved model using weak supervision. The goal of this stage is to improve the performance of the model learned during the unsupervised stage by providing it with more object (face) examples. This stage could also be performed by the system in autonomous (unsupervised) online manner by crawling on web images and detecting instances of the object using the M_1 model. In our case we gave the system 300 additional image examples randomly chosen from the LFW database and ran the M_1 model on them to detect and localize faces on these images. The output of M_1 were face bounding

boxes detected in the training images. We then train the model M_2 using the UCA method by assuming the output of M_1 to be the output of the G-phase in one of UCA iterations (see section 2.1 for the definition of the G-phase). The model M_2 is then used to perform the face detection and localization in section 3. The second stage can be seen as additional iterations of the UCA method applied in the first stage. The main difference is that the standard UCA loops over the same image set over and over again, while in M_2 training stage new images are provided either in a weakly supervised or in online unsupervised manner.
Stage 3: The third stage, training a recognition model for a specific face, is performed when we receive a (limited) set of examples of a face of a certain individual (that we wish to recognize in the future) and train a UCA model on these images. The training examples need not be cropped or aligned and can exhibit any scale variation, all we require them to be is 'roughly frontal' (as the example images in Figure 7). The training is organized along the lines of the second stage. The resulting model M_3 can be used by itself for simultaneous detection, localization and recognition of a specific person, but it is better used in conjunction with M_2, since M_3 had only limited training on the few examples provided for the specific person. We investigate both of these options in the section 3.

Figure 2 summarizes the proposed approach. Section 3 describes the experimental validation of our method.

3 Results

To test the proposed method, we applied it to frontal face detection, localization and recognition tasks in several databases, namely: people-containing images gathered from Google image search (denoted WEB database); Labeled Faces in the Wild (LFW) database [15]; MIT-CMU dataset [16]; and a set of unconstrained images of a person taken under different viewing conditions, and at different locations and times (denoted PERSON database). The MIT-CMU dataset combines all the images from tests A, B and C in the dataset description [16]. The statistics of all the datasets are summarized in Table 1.

Name	# images	# frontal faces	image size (rows x columns + STD)	Faces size (min – max)	Has multiple people	Has scale variations	Faces are aligned
WEB	354	477	626×593±197×220	31×20–180×130	+	+	–
LFW	13233	13233	250×250±0×0	135×95	–	–	+
MIT-CMU	130	511	429×440±227×208	21×14–253×168	+	+	–
PERSON	162	162	480×640±0×0	36×28–209×160	–	+	–

Table 1. Provides statistics of the datasets used in our experiments

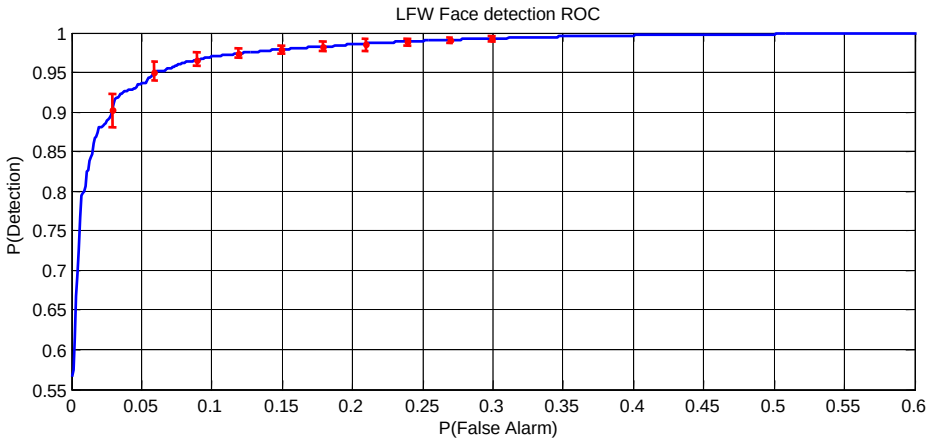


Fig. 3. Summary of the LFW face detection experiments. Although faces in LFW are of the same scale, a full method including scale search was applied both to face-containing and background images

In the first experiment, the model M_2 was applied to the face detection task in the LFW database. Since faces in the LFW are cropped, roughly aligned and equally scaled, there was no reason to test localization on that database, although in several dozens of images that we manually checked the localization was perfect. The LFW database is comprised only of images containing faces detected by Viola & Jones face detector, so in order to test face detection, maximal response of the UCA was computed for all images in the LFW and in additional 916 background images which were a union of the Caltech and Google background sets. These measurements were used to build the ROC curve for face detection depicted on Figure 3. The error bars of the ROC were computed by 10-fold cross validation.

In the second experiment we tested the combined detection and localization performance of our method. To this end, model M_2 was applied to the WEB and MIT-CMU datasets. The results are summarized in ROC curves in Figures 4a and 4b respectively. We also compared our performance on the WEB database with the performance of the OpenCV [21] implementation of Viola & Jones face detector [1] and shown a significant performance gain (see figure 4a). The face was considered correctly localized if the detected bounding box exceeded the standard Jaccard index overlap score of 0.5 with the manually marked ground truth bounding box:

$$Jaccard\ Index\,(BB_1, BB_2) = \frac{|BB_1 \cap BB_2|}{|BB_1 \cup BB_2|} \tag{1}$$

In order to test the contribution of various components of our method to the final performance, Figure 4a also contains ROC curves of performance after removing different components. Specifically, we tested our method without

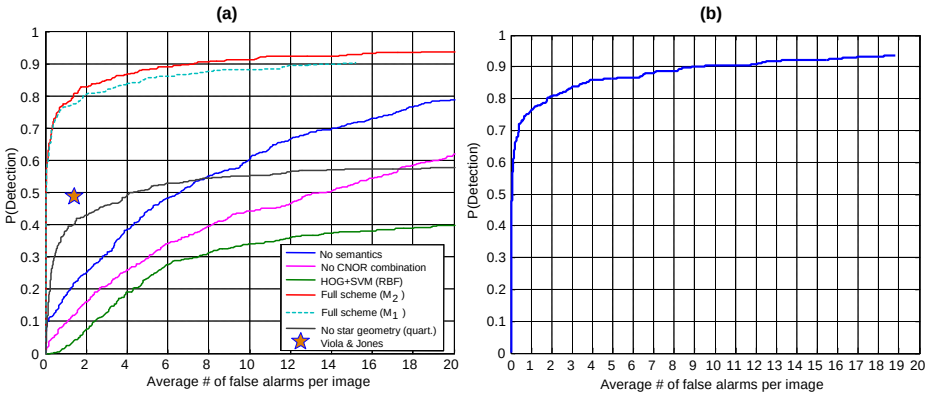


Fig. 4. (a) Summary of the WEB face detection and localization experiments. Additionally provides comparison with OpenCV implementation of [1], HOG+SVM and performance after removing different components of our method. **(b)** Summary of the MIT-CMU face detection and localization experiments, ROC curve

using semantic equivalents for part detection ('No semantics' in the figure), without using the CNOR model for combining the semantic equivalents ('No CNOR combination') and without using the learned geometry of parts ('No star geometry'). In all cases, removal of these components resulted in significant drop in performance. When semantic equivalents were not used for part detection, a single best image fragment was chosen to represent the part. When CNOR model was not used for combining semantic equivalents, they were combined using a simple summation. When part geometry was not used, the parts were split into four groups corresponding to the four quarters of the face. The face was then detected as a combined vote of the four quarter detections, while each of the quarters was detected by the bag-of-features method. Localizing faces directly by a bag-of-features of the whole face (without using this four quarters scheme) failed to produce reasonable results due to the limitation imposed by the overlap score (eq. 1) and the currently used non-maximal suppression scheme.

Additionally, figure 4a contains an ROC curve of HOG + SVM classifier implemented along the lines of [22]. HOG descriptors were computed for all the face bounding boxes detected by M_1 in all the images used to train M_2. For these HOG descriptors RBF kernel SVM was trained against HOG descriptors of maximal score bounding boxes detected by M_1 in background images (Caltech + Google backgrounds).

In the third experiment combined detection, localization and recognition were tested on the combination of the PERSON and the WEB datasets. The results are summarized in figures 5a and 5b. Ten images of a person were used to train the model M_3. The models M_2 and M_3 were used in a cascade-like sequence. First, M_2 was applied to detect candidate faces in the image and then M_3 was applied to search for the face of the specific person at locations and scales close to the ones detected by M_2. We also tested the combined detection, localization

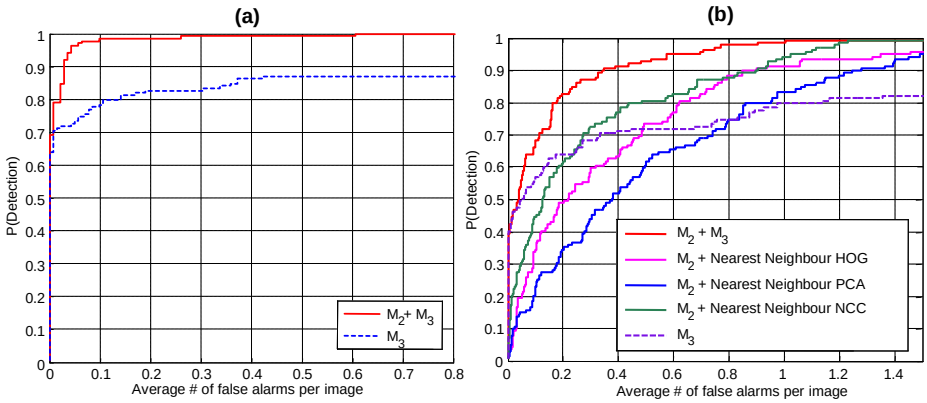


Fig. 5. **(a)** Summary of PERSON face detection and localization experiments. **(b)** Summary of PERSON + WEB combined detection, localization and recognition experiments

and recognition using a single model (M_3). When used by itself, it produces less good detection and localization results (see Fig. 5a) due to the limited training on only 10 images. In figure 5b we also compare our approach to several baseline approaches. Specifically, we tried to replace the M_3 in the $M_2 \rightarrow M_3$ sequence by Nearest Neighbor PCA, NCC or HOG, all trained on the same 10 images we used for training M_3. As can be seen in Figure 5b, the performance of these methods is significantly worse. This can be partially attributed to the fact that these methods try to detect the whole face using a single template, while our method applies part based detection which, as explained in the introduction, is more appropriate for the recognition task. Examples of face interpretation, that is detection of various facial parts, are given in Figure 6.

4 Discussion

This paper presented a method for combined faces detection, localization, part interpretation and recognition in unconstrained images, where faces may appear at any image location, scale and under a variety of difficult viewing conditions. The method shows promising performance in various experiments on different datasets, superior to several standard baseline methods and the standard implementation of the Viola & Jones face detector. The method requires only a few images for training and can be trained in a fully unsupervised manner. The underlying models employed by the method are general and not limited to face detection.

Even in cases of semi-supervised training, the ability to automatically separate different object views in a given training set (containing mixed views) can improve classification performance of the learned models. Moreover, this ability also facilitates learning part based models that are capable of detecting

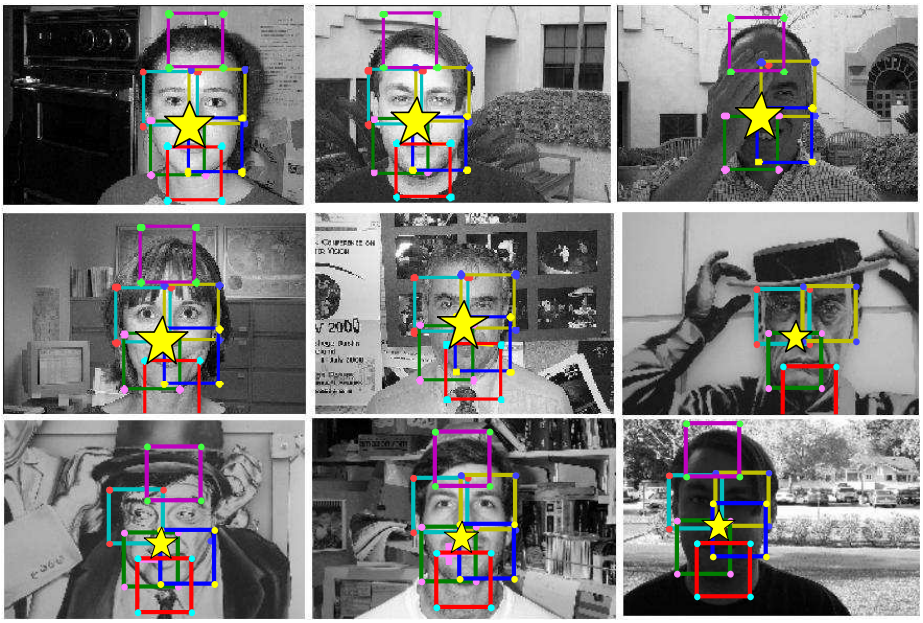


Fig. 6. Examples of detections of several of the (about 50) modeled parts. Yellow star shows the detected object model center location and the colored boxes show the parts that were detected by the model

'meaningful' object parts. Initial experiments performed in [17] for automatic separation of car views from the PASCAL 2007 dataset, show that our method has this ability. An interesting future research direction would be learning viewpoint invariant part based models for face detection from a mixed set of examples of different face views. Additional interesting extension may be linking the models of different views in terms of their detectable parts (such that semantically equivalent parts from different view models are linked) in order to facilitate view-invariant part interpretation.

In its current version, the method does not make full use of the detailed part interpretation obtained during the detection process for the purpose of subsequent individual face identification. Since part detection obtained by our method is usually highly accurate, for each facial part it is possible (in future work) to build a universal dictionary of part appearances, such as semantic equivalent image fragments proposed in [23]. Then, one may learn an association between each of the part appearances and the conditions (lighting, pose, expression, etc.) under which the part is viewed. Given even a single image of a specific individual, her facial parts may be detected (by our model) and categorized according to the learned part appearance dictionaries. Subsequently, when confronted with a new image of the same individual taken under different conditions, the learned association between the conditions and the part appearances in the new image

(known following the detection, localization and part interpretation performed by our model) could be used to recognize the individual. Some ideas along these lines were explored in [24] (but without having a method for reliable part interpretation) and extending it might be an interesting future research direction towards individual recognition under highly varying viewing conditions.

Aknowledgment: This work was supported by EU IST Grant FP6-2005-015803 and ISF Grant 7-0369.

References

1. Jones, M.J., Viola, P.A.: Robust real-time face detection. ICCV (2001)
2. M.-H. Yang, D.K., Ahuja, N.: Detecting faces in images: A survey. PAMI, vol. 24 (2002)
3. Yang, G., Huang, T.S.: Human face detection in complex background. Pattern Recognition, vol. 27 (1994)
4. Yow, K., Cipolla, R.: Feature-based human face detection. Image and Vision Computing, vol. 15 (1997)
5. Sinha, P.: Object recognition via image invariants: A case study. Investigative Ophthalmology and Visual Science, vol. 35 (1994)
6. Rowley, H., Baluja, S., Kanade, T.: Neural network-based face detection. PAMI, vol. 20 (1998)
7. Sung, K.K., Poggio, T.: Example-based learning for view-based human face detection. PAMI, vol. 20 (1998)
8. Schneiderman, H., Kanade, T.: A statistical method for 3d object detection applied to faces and cars. CVPR (2000)
9. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. European Conference on Computational Learning Theory (1995)
10. Lienhart, R., Maydt, J.: An extended set of haar features for rapid object detection. ICIP (2002)
11. Schneiderman, H.: Feature-centric evaluation for efficient cascaded object detection. CVPR (2004)
12. Luo, H.: Optimization design of cascaded classifiers. CVPR (2005)
13. Huang, C., Ai, H., Li, Y., Lao, S.: High-performance rotation invariant multi-view face detection. PAMI (2007)
14. Yan, S., Shan, S., Chen, X., Gao, W.: Locally assembled binary (lab) feature with feature-centric cascade for fast and accurate face detection. CVPR (2008)
15. Huang, G., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. University of Massachusetts, Amherst, Technical Report 07-49 (2007)
16. Online: http://www.vasc.ri.cmu.edu/idb/html/face/frontal_images. CMU VASC Frontal Face Database (1997)
17. Karlinsky, L., Dinerstein, M., Levi, D., Ullman, S.: Unsupervised classification and part localization by consistency amplification. ECCV (2008)
18. Jurie, F., Triggs, B.: Creating efficient codebooks for visual recognition. ICCV (2005)
19. Neal, R., Hinton, G.: A view of the em algorithm that justifies incremental, sparse, and other variants. Learning in Graphical Models (1998)

20. Friedman, N.: The bayesian structural em algorithm. UAI (1998) 129–138
21. Online: http://www.opencvlibrary.sourceforge.net. OpenCV (2006)
22. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. CVPR (2005)
23. Epshtein, B., Ullman, S.: Identifying semantically equivalent object fragments. CVPR (2005)
24. Jurie, F., Triggs, B.: Creating efficient codebooks for visual recognition. ICCV (2005)

Fig. 7. Examples of face detection, localization and recognition obtained by our method. The databases used for experiments are explained in the Results section. Examples inside the red box are from the WEB database, examples in the green box are from the MIT-CMU database and blue box contains examples of combined detection, localization and recognition of a specific person in the PERSON database. The current algorithm does not use color information, all the color images were processed in grayscale by the algorithm

Unsupervised Feature Optimization (UFO): simultaneous selection of multiple features with their detection parameters

Leonid Karlinsky Michael Dinerstein Shimon Ullman
{leonid.karlinsky,michael.dinerstein,shimon.ullman}@weizmann.ac.il
Weizmann Institute of Science, Rehovot 76100, Israel

Abstract

Class learning, both supervised and unsupervised, requires feature selection, which includes two main components. The first is the selection of a discriminative subset of features from a larger pool. The second is the selection of detection parameters for each feature to optimize classification performance. In this paper we present a method for the discovery of multiple classification features, their detection parameters and their consistent configurations, in the fully unsupervised setting. This is achieved by a global optimization of joint consistency between the features as a function of the detection parameters, without assuming any prior parametric model. We demonstrate how the proposed framework can be applied for learning different types of feature parameters, such as detection thresholds and geometric relations, resulting in the unsupervised discovery of informative configurations of objects parts. We test our approach on a wide range of classes and show good results. We also demonstrate how the approach can be used to unsupervisedly separate and learn visually similar subclasses of a single category, such as facial views or hand poses. We use the approach to compare various criteria for feature consistency, including Mutual Information, Suspicious Coincidence, L2 and Jaccard index. Finally, we compare our approach to a parametric consistency optimization technique such as pLSA and show significantly better performance.

1. Introduction

In this paper we consider the problem of unsupervised selection of multiple classification features together with the optimization of their detection parameters and discovery of consistent feature configurations. The input is a mixed set of unlabeled images S, only a small subset of which (20% or lower) contains instances of an unknown class category (which may be uncropped, unaligned and of small size relative to the background), and a large pool of candidate features $\mathcal{F}$ (is of size 1000-3000 in our experiments), which is measured on each image. The feature pool may also be generated directly from all the images in the unsupervised training set (e.g. by applying the method of [9]). As in many classification schemes, each feature $F_i \in \mathcal{F}$ is associated with some parameters θ_i that need to be set. When θ_i are fixed, the feature $F_i\,(I;\theta_i)$ is a function mapping image I to some discrete (usually binary) value. For example, the parameter θ_i may be the similarity threshold and the feature F_i be a binary function with $F_i = 1$ iff the maximal similarity between all descriptors extracted from image I and a descriptor associated with F_i exceeds θ_i. Another example is that the feature is a pair (F_i, F_j) (where F_i and F_j are "maximal similarity + threshold" features from the previous example) and θ_i is an expected spatial offset between the detected locations of the two members of the pair. The combined feature is detected only if the observed offset is close to θ_i. Finally, θ_i may be the linear coefficients for combining similarity measures of several different types of descriptors for a given image patch associated with F_i. Such de-

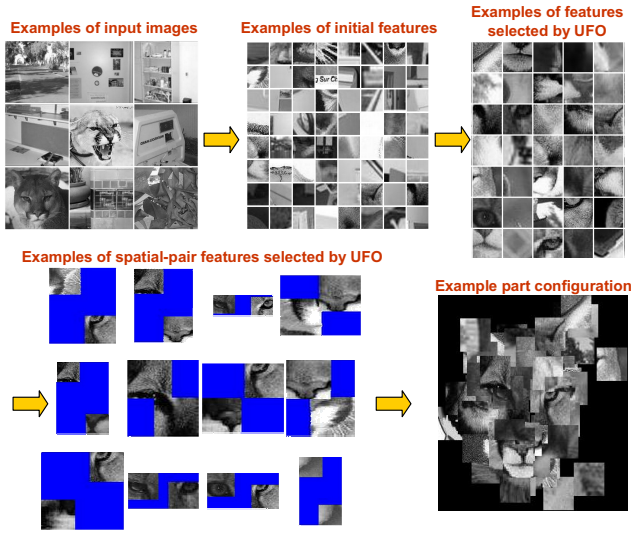


Figure 1. An example of the application of our method to cougar class from Caltech-101.

scriptor combinations were successfully used in [21]. The main goal of the algorithm is to identify a subset of the candidate features that are most discriminative for detecting the category instances together with optimizing parameters of each feature to find their most discriminative values. This is challenging since the image class labels are unknown and cannot be used to identify useful features. Another goal is to use the discovered subset of optimized features to detect and localize a reliable subset of class instances present in the unsupervised training set. The detected class examples can then be used for the subsequent category learning (e.g. as done in [14, 10]). Figure 1 illustrates the input, the various outputs and the intermediate stages of our method.

The problem of unsupervised learning of an unknown category has recently attracted substantial attention [2, 4, 5, 6, 7, 8, 10, 20, 15, 12, 13, 14, 19, 17, 18, 23]. The unsupervised approaches may be further subdivided into weakly supervised - ones that assume that all training images contain uncropped and unaligned instances of the learned object category [6, 15, 18, 23], and fully unsupervised - ones for which the category instances may appear only in a small portion of the training images [8, 20, 7, 5, 19, 4, 17, 12, 2, 13, 14, 10]. The approach proposed in this paper belongs to the latter, fully unsupervised, category. The general idea in all of these approaches is to compensate for the lack of supervision by searching for consistency between the measured features, assuming that this consistency arises primarily from the presence of a class instance in the image. The approaches may be further categorized in terms of how they search for this inter-feature consistency. Some approaches assume a prior parametric model for the features and their detection parameters, and the consistency is detected via unsupervised training of the parameters of this model. The models used by these approaches include the constellation model [6] used by [6, 7], pLSA used by [20, 5, 19, 12, 14], Spatial-LTM [4], ISM [11] used by [8], Semantic-Shift [13], TSI-pLSA [5] and UCA [10]. Other approaches are non-parametric in a sense that no prior model is assumed for the inter-feature consistency and the consistency is detected by some form of a bottom-up agglomerative process. This process builds upon local, usually pairwise (for a given level of agglomeration), consistency relations between the features. Examples of the latter approaches are efficient mining used by [17, 18], joining frequent feature triplets with high suspicious coincidence measure by [23] and combining similar image segments by [2]. Our approach belongs to the latter non-parametric type. However, it does not perform an iterative agglomeration based on the local consistency of the features. Instead, a global optimization is performed, optimizing the consistency between all the potential features as a function of the feature parameters. The output of the global optimization is a consistent setting of feature parameters, maximizing the sum of pairwise consistency values

Figure 2. Examples of part configurations learned by UFO for various object classes. Each configuration displays a set of patch features automatically selected and arranged by the algorithm for each category.

between the relevant features. Clustering the resulting consistency graph between the features gives rise to consistent feature clusters, which are discriminative for the unknown learned category (or several categories in case of a multi-class), as demonstrated in our experiments.

An important consideration is that if we want to compute consistency of feature F_1 with two different features F_2 and F_3, then the parameter (e.g. threshold) of F_1 that attains maximal consistency with F_2 may be different from the one needed for maximal consistency with F_3. Consequently, in order to have maximal joint consistency of many features, the parameter optimization needs to be global. An agglomerative process that optimizes consistency only locally by merging a few (usually 2 or 3) features at a time, may arrive at lower consistency at the higher levels of agglomeration or become inconsistent in the setting of parameters.

As in previous approaches, the proposed framework can be used with different types of consistency measures. Therefore, in our experiments we used the proposed method to compare different consistency measures, such as Suspicious Coincidence (SC), Mutual Information (MI), Jaccard Index (JI) and L2 distance. Surprisingly, the standard SC consistency measure attained lower results in our comparison, significantly inferior to MI and JI measures. In addition, we compare our approach to pLSA - probably the most widely used parametric learning method based on consis-

tency, and show significantly better performance.

Unsupervised learning can be particularly useful in the weakly supervised setting, when all the images contain instances of a given class, but it is beneficial to divide the class into a set of visual sub-classes (such as different object views or different poses), to obtain better recognition of each sub-class individually, instead of recognizing a mixture of sub-classes by a single model. In the experiments below, we demonstrate how the proposed method can be used to obtain unsupervised separation between visually similar sub-classes, such as separating face views and separating hand poses.

Finally, we exploit the fact that the proposed framework may be used with any types of features and their parameters, and demonstrate how our approach can be used in a pipeline that starts from a generic set of quantized image patch descriptors and a set of unlabeled images, and finally arrives at discovering spatial configurations of object parts, which are both discriminative and visually plausible (see Figure 2). Within this pipeline, the method is applied twice: first for learning detection thresholds, and second for learning spatial offset parameters of feature pairs, finally leading to the discovery of full $2D$ consistent feature configurations.

The rest of the paper is organized as follows, section 2 describes the proposed approach, section 3 summarizes the experimental results, and section 4 provides summary and discussion.

2. Method

In this section we describe the Unsupervised Feature Optimization (UFO) algorithm (section 2.1), discuss and analyze the pair-wise consistency measures that can be efficiently used within the algorithm (section 2.2), and show how the UFO algorithm can be applied to select appearance and geometric features, including their respective detection parameters, in order to discover objects and consistent geometric configurations of their parts (section 2.3).

The UFO algorithm receives as input a set of unlabeled images and a feature pool from which it performs the selection. However, in practical applications, such as described in section 2.3, it is possible to work with either an external feature pool provided by the user, or with a feature pool internally generated by the system from all the unsupervised training images (e.g. by applying the method by [9]). A schematic diagram describing the proposed method is given in figure 3 and explained further in section 2.3.

2.1. UFO algorithm

Input: A set of images $S = \{I_n\}_{n=1}^{N}$ and a (large) pool of features together with their respective detection parameters $\mathcal{F}_0 = \{\langle F_i, \theta_i \rangle\}_{i=1}^{M}$, such that each feature is a function: $\langle F_i, \theta_i \rangle : S \rightarrow V$, here V is a discrete set of possible feature values (we use binary values in our experiments), and each θ_i takes a value from a discrete set of possible values that may be specific to each feature $\theta_i \in W_i$. For simplicity, we will use the notation $F_i(\theta_i)$ to indicate a row vector of N values from V, the n-s entry of which will be $\langle F_i, \theta_i \rangle$ measured on image I_n. We will also refer to the feature itself as F_i.

Output: K disjoint subsets of features $\bigcup_{1 \leq k \leq K} \hat{\mathcal{F}}_k = \mathcal{F}_0$, and the respective optimal parameter settings for each feature: $\left\{\hat{\theta}_i\right\}_{i=1}^{M}$. The goal is that one or several of these subsets will be associated with the classes present in the images of set S.

The flow of the algorithm:

1. Compute maximal consistency for each pair of features F_i and F_j: $C_{ij} = \max_{\theta_i, \theta_j} Cons\left[F_i(\theta_i); F_j(\theta_j)\right]$, where the consistency measure, $Cons$, is a function measuring pairwise consistency between two vectors from V^N: $Cons : V^N \times V^N \rightarrow \mathbb{R}$. The consistency measures that we have compared within our algorithm, as well as methods to efficiently compute $Cons\left[F_i(\theta_i); F_j(\theta_j)\right]$ for all i and j and all values of θ_i and θ_j, are described in section 2.2.

2. Transform the resulting $M \times M$ matrix $C = \{C_{ij}\}_{i,j=1}^{M}$ into a graph adjacency matrix A. This is obtained by setting all except the largest L entries of C to zero and the largest L entries to one (C and A are symmetric, we refer to L entries above the diagonal and keep their L respective reflections as well). The resulting graph has M nodes, one node for each feature, and L edges. Throughout all our experiments we used $L = 10,000$. In principle a large value for L is preferred, the only reason to limit its value is to lower the running time of the graphical model based optimization that will be described next. For each edge described by non-zero entry A_{ij} we compute a potential function: $w_{ij} : W_i \times W_j \rightarrow \mathbb{R}$ such that: $w_{ij}(\theta_i, \theta_j) = Cons\left[F_i(\theta_i); F_j(\theta_j)\right]$. The potentials are then used to define an energy function E being the sum of the pairwise consistency measures:

$$E(\theta_1, \ldots, \theta_M) = \sum_{i,j:A_{ij}=1} w_{ij}(\theta_i, \theta_j) \qquad (1)$$

Energy function E is a sparse approximation of the full pairwise consistency measure between all the features.

3. Apply a Loopy-Belief-Propagation (LBP) algorithm to compute (approximate) the maximal assignment to E:

$$\left\{\hat{\theta}_i\right\}_{i=1}^{M} \underset{LBP}{\approx} \arg\max E(\theta_1, \ldots, \theta_M) \qquad (2)$$

4. Finally, compute: $\hat{C}_{ij} = Cons\left[F_i\left(\hat{\theta}_i\right); F_j\left(\hat{\theta}_j\right)\right]$ and cluster the resulting $M \times M$ affinity matrix $\hat{C} = \left\{\hat{C}_{ij}\right\}_{i,j=1}^{M}$ into K clusters (we use $K = 7$ in all our experiments). In our implementation of the UFO algorithm, we use spectral clustering [22] for the last step.

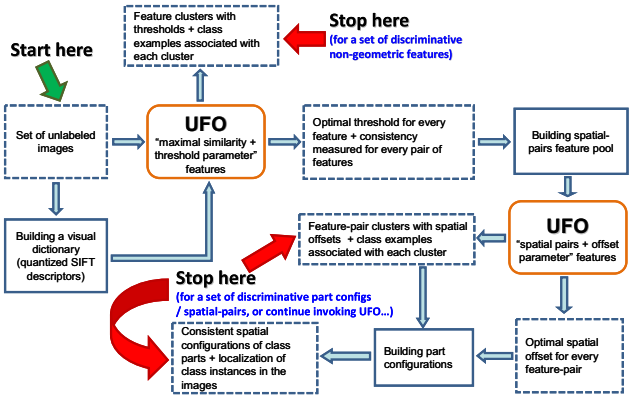


Figure 3. A schematic diagram showing the flow of the method. Details are explained in section 2.3. The UFO algorithm, explained in section 2.1, is applied twice, first for the non-geometric "maximum-similarity" features, and second for the geometric "spatial-pair" features constructed using the results of the first UFO application. The dashed boxes describe input, output and intermediate results. The solid boxes describe assisting algorithms explained in the section 2.3. Entry and optional exit points of the flow are indicated by green and red arrows respectively.

Intuition - Class and feature consistency: As will be demonstrated in the results section 3, the feature clusters obtained by the method are closely associated with the unknown classes. The intuition behind the UFO algorithm is that the main source of consistency between the features is their association with specific classes. For features with high enough consistency with the same class (or sub-class), there will also be high consistency between the features themselves. This is intuitive, but also follows from the lower bounds on feature-to-feature consistency derived analytically for all the consistency measures in the next section 2.2. The bounds become tighter as the consistency between the features and the class (for the optimal setting of detection parameters) increases. Therefore, keeping graph edges with high consistency C_{ij} has a high likelihood to form edges between features consistent with the same class. Moreover, choosing detection parameters $\left\{\hat{\theta}_i\right\}_{i=1}^{M}$ that maximize consistency between the features, will also contribute to increasing the feature-to-class consistency.

The following section describes several natural consistency measures that can be efficiently used within the proposed UFO algorithm, and provides lower bounds for feature-to-feature consistency in terms of the respective feature-to-class consistency for each of the measures.

2.2. Consistency measures and efficient implementation

In our experiments with the UFO algorithm we tested and compared four consistency measures (denoted $Cons$ in the description of the algorithm), namely: Jaccard Index (JI), Mutual Information (MI), Suspicious Coincidence (SC) and Euclidian Distance (L2). As explained in section 2.3, in our experiments we focused on binary features and their parameters. In order to implement the UFO algorithm efficiently, we need to efficiently estimate $Cons\left[F_i\left(\theta_i\right);F_j\left(\theta_j\right)\right]$ for all i and j and all values of θ_i and θ_j. Fortunately, in the binary feature case, the sufficient statistics for computing $Cons\left[F_i\left(\theta_i\right);F_j\left(\theta_j\right)\right]$ for all the listed measures are only five numbers: the number of images for which both $F_i\left(\theta_i\right)$ and $F_j\left(\theta_j\right)$ are equal to one (the inner product $F_i\left(\theta_i\right)\cdot F_j\left(\theta_j\right)$), the number of images for which they are both zero $\bar{F}_i\left(\theta_i\right)\cdot\bar{F}_j\left(\theta_j\right)$, the $\bar{F}_i\left(\theta_i\right)\cdot F_j\left(\theta_j\right)$ (needed for MI), and the numbers of ones in $F_i\left(\theta_i\right)$ and $F_j\left(\theta_j\right)$ respectively ($\left|F_i\left(\theta_i\right)\right|_1$ and $\left|F_j\left(\theta_j\right)\right|_1$). W.l.o.g., assume that the sets of possible values for the detection parameters: W_i are of the same cardinality $|W_i|=T$ for all i. Then the complexity of the full computation of all the required sufficient statistics is $2\cdot T^2$, and in case of MI $3\cdot T^2$, times the complexity of binary matrix multiplication.

In the following text we describe all the tested consistency measures and for each of them provide a lower bound on feature-to-feature consistency $Cons\left[F_i\left(\theta_i\right);F_j\left(\theta_j\right)\right]$ (we will write $Cons\left[F_i;F_j\right]$ for brevity) in terms of respective feature-to-class consistencies: $Cons\left[F_i;C\right]$ and $Cons\left[F_j;C\right]$. The bounds are monotonically increasing in both $Cons\left[F_i;C\right]$ and $Cons\left[F_j;C\right]$, and become tighter as the feature-to-class consistencies approach their maximum. A notable observation regarding these bounds is that they hold for any class labeling C. Therefore, even though two features might have low consistency with the entire class (e.g. faces), they might be very consistent with the same sub-class (e.g. profile faces) and thus have high lower bound on their feature-to-feature consistency (using this sub-class labeling as C). Supporting this claim, we have successfully applied our method for unsupervised separation of visually similar sub-classes of a larger class, such as separating face views and separating hand poses (see section 3 for more details).

Jaccard Index (JI): The JI measures the amount of consistency between two sets by computing the ratio between their intersection and their union. The JI between two binary vectors X and Y is defined as:

$$JI\left(X,Y\right)=\frac{X\cdot Y}{X+Y-X\cdot Y} \tag{3}$$

where $X\cdot Y$ is the dot product. This is exactly the classical JI for sets of indices of ones in the two vectors. In our experiments we use the following symmetric variant of JI:

$$\widehat{JI}\left(X,Y\right)=0.5\cdot\left[JI\left(X,Y\right)+JI\left(\bar{X},\bar{Y}\right)\right] \tag{4}$$

where $\bar{X}=1-X$. This variant is used in order to prevent an artificial tendency for larger consistency due to greater number of ones in the binary vectors.

Claim 2.1

$$JI\left[F_i;F_j\right] \geq \frac{JI\left[F_i;C\right]+JI\left[F_j;C\right]-1}{1/JI\left[F_i;C\right]+1/JI\left[F_j;C\right]-1} \qquad (5)$$

similar result holds for $\widehat{JI}\left[F_i;F_j\right]$.

The proofs of this and following claims are given in the supplementary material. This shows that $\widehat{JI}\left[F_i;F_j\right]$ is increasing in both the JIs: $JI\left[F_i;C\right]$ and $JI\left[F_j;C\right]$, and the bound (5) becomes tight when the JIs attain their maximal values of 1.

Mutual Information (MI): The MI measures the decrease in the entropy of one random variable conditioned on another random variable. For two binary vectors X and Y:

$$MI\left[X;Y\right] = H\left(X\right)+H\left(Y\right)-H\left(X,Y\right) \qquad (6)$$

where H is the Shannon's entropy of a random variable.

Claim 2.2 *Assuming that F_i and F_j are conditionally independent given class C, then:*

$$MI\left[F_i;F_j\right] \geq MI\left[F_i;C\right]+MI\left[F_j;C\right]-H\left(C\right) \qquad (7)$$

Because the maximal value that $MI\left[F_i;F_j\right]$ can attain is: $\min\left[H(F_i),H(F_j)\right]$, maximizing $MI\left[F_i;F_j\right]$ will tend to prefer F_i and F_j that are correlated and non-sparse in terms of both zeros and ones (with higher entropy). Unlike MI, JI can have high values for correlated features that appear very infrequently (these features will have low MI scores, because of their low entropy).

Suspicious Coincidence (SC): The SC measures the ratio between the probability of two events happening simultaneously and an estimate of this probability assuming the two events are independent. For binary vectors X and Y of length N:

$$SC\left[X;Y\right] = \frac{N\cdot\left(X\cdot Y\right)}{\left|X\right|_1\cdot\left|Y\right|_1} \qquad (8)$$

Claim 2.3 *Assume the events $F_i=1$ and $F_j=1$ are conditionally independent given the event $C=1$, then:*

$$SC\left[F_i;F_j\right] \geq SC\left[F_i;C\right]\cdot SC\left[F_j;C\right]\cdot\left|C\right|_1/N \qquad (9)$$

The SC is a commonly used consistency measure, e.g. in [23] it was used for weakly supervised learning of object contours. A drawback of the SC is that taken by itself it may prefer very infrequent events. The reason is that if X has few ones in it, then $SC\left[X;X\right]=1/\left|X\right|_1$ is very large. Therefore, SC is usually used in conjunction with a threshold on minimal frequency of the events (i.e. demanding that $\left|X\right|_1$ is large enough). However, this is problematic in our fully unsupervised setup, since the class instances themselves are infrequent (there may be 20% or less class instances in a set). This inherent drawback of the SC is supported by our experiments (section 3), where SC performs significantly worse then MI or JI.

Euclidian Distance (L2): The bounds for L2 are provided by the triangular inequality. A drawback of L2 is that it is dominated by the larger of $X\cdot Y$ and $\bar{X}\cdot\bar{Y}$ (X and Y defined as above), and hence may disregard the lack of consistency between either zeros or ones in the vectors X and Y, depending which (zeros or ones) are less frequent.

2.3. Implementation Details

The flow of the proposed method is schematically shown in figure 3. In this section we describe the auxiliary steps of the method in more detail. All the constant numbers appearing in the description are fixed parameters of the method and are used throughout the experiments.

Initial feature pool: we apply the method of [9] on the entire image set S (both class and non-class, since labeling is unknown) to build an initial feature pool $\mathcal{F}_0$ of $M_0=1000$ quantized SIFT descriptors of 40×40 image patches. For every image in the set S, we compute the similarity of each feature from the pool at all locations on a dense grid. For each feature $F_i\in\mathcal{F}_0$ and image $I_n\in S$, the five highest similarity local maxima locations are retained. Denote their respective similarity scores by $0\leq\alpha_{i,n}^k\leq 1$ (in decreasing order for each feature) and image locations by $L_{i,n}^k\in\mathbb{R}^2$, where $k\in\{1,\ldots,5\}$.

Unsupervised optimization of threshold parameters: we apply the UFO algorithm (section 2.1) to learn individual optimal threshold parameters θ_i^{thr} for each $F_i\in\mathcal{F}_0$. We use only the maximal similarity scores $\alpha_{i,n}^1$ in this optimization. For each of the optimized threshold parameters θ_i^{thr} we set a range of 20 candidate values W_i^{thr}, by taking 20 values with equal spacing in the index from the sorted list $sort\left(\alpha_{i,1}^1,\alpha_{i,2}^1,\ldots,\alpha_{i,N}^1\right)$. This ensures that each feature has an adequate set of candidate values for its threshold for any density of its continuous similarity values. As explained in section 2.1, following threshold setting, the UFO also produces clusters of features, which are tested for their classification performance in the results section 3.

Building spatial-pairs feature pool: an additional output of the UFO (section 2.1) is the feature-to-feature consistency matrix $\hat{C}=\left\{\hat{C}_{ij}\right\}_{i,j=1}^{M_0}$ computed for the chosen optimal threshold parameters. We next build spatial-pair features from $M_1=3000$ pairs of $\mathcal{F}_0$ features that have maximal pair-wise consistency (i.e. maximal entries in the matrix $\hat{C}$). The spatial-pair features are pairs of $\mathcal{F}_0$ features that have a Gaussian model for the spatial offset between their detected locations. For each spatial-pair feature in the resulting feature pool, denoted $\mathcal{F}_1$, we train a separate Gaussian Mixture Model (GMM) producing five competing Gaussian spatial offset models. The GMM for a spatial-pair feature $(F_i,F_j)\in\mathcal{F}_1$ is trained on a set $O_{i,j}$ of offsets between F_i and F_j in all their detected locations (from all the images)

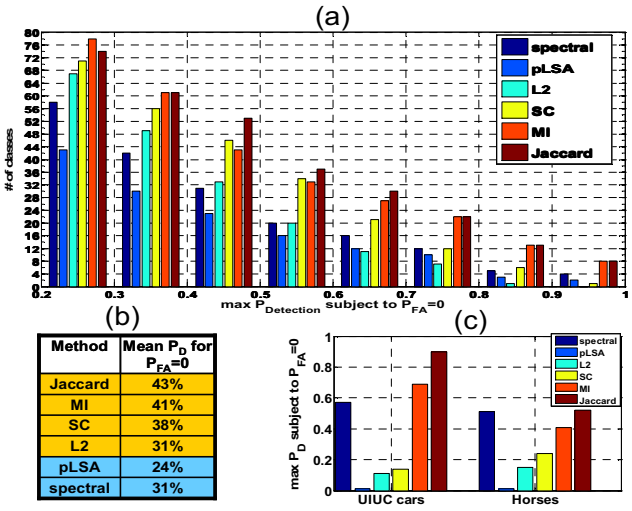
that passed the optimal thresholds (computed by UFO) $\hat{\theta}_i^{thr}$ and $\hat{\theta}_j^{thr}$ respectively: $O_{i,j} = \left\{ L_{i,n}^{k_1} - L_{j,n}^{k_2} \mid 1 \le n \le N, \right.$ $\left. 1 \le k_1, k_2 \le 5,\ \alpha_{i,n}^{k_1} > \hat{\theta}_i^{thr},\ \alpha_{j,n}^{k_2} > \hat{\theta}_j^{thr} \right\}$. Denote the Gaussian components of the learned mixture: $G_{i,j}^k = N\left(\mu_{i,j}^k, \Sigma_{i,j}^k\right)$, where $k \in \{1, \ldots, 5\}$. The detection parameter for the spatial-pair feature $(F_i, F_j) \in \mathcal{F}_1$ is $\theta_{i,j}^{offs} \in \{1, \ldots, 5\}$, and the pair-feature is considered detected in image $I_n \in S$ with $\theta_{i,j}^{offs} = k$, iff there exists an offset in $O_{i,j}$ that originates from image I_n and for that offset the $G_{i,j}^k$ component has maximum likelihood among all the GMM components.

Unsupervised optimization of offset parameters: we apply the UFO algorithm again, this time for choosing the correct Gaussian offset model among the five individual candidates for each spatial-pair feature. The outputs of the algorithm are the optimal offset models $\hat{G}_{i,j} = N\left(\hat{\mu}_{i,j}, \hat{\Sigma}_{i,j}\right)$ for the spatial-pair features $(F_i, F_j) \in \mathcal{F}_1$, clusters of these features, and pair-wise consistency measured between each two spatial-pair features.

Building consistent configurations of object parts: The next goal is to derive for each cluster of the spatial-pair features (computed by the UFO) 2-D feature configurations that will be as consistent as possible with the pair-wise offsets found in the previous step. Example configurations are shown in figure 2. They are usually visually plausible in the sense that they correspond to a repeating part configuration in the object. Since each feature in a configuration is associated with some object part, we call them "part configurations". Let $\{(F_{i_1}, F_{j_1}), \ldots, (F_{i_L}, F_{j_L})\} \subset \mathcal{F}_1$ be a cluster of spatial-pair features, and let $\left\{ \hat{G}_{i_1,j_1}, \ldots, \hat{G}_{i_L,j_L} \right\}$ their respective optimal offset models selected by the UFO. First, we solve a 2D placement problem, where we define unknowns (x_i, y_i) for each feature F_i belonging to at least one of the pairs in the cluster, and solve the following system of linear equations, two equations for each pair in the cluster:

$$x_{i_l} - x_{j_l} = \hat{\mu}_{i_l,j_l}(1), \quad y_{i_l} - y_{j_l} = \hat{\mu}_{i_l,j_l}(2) \qquad (10)$$

where the $\hat{\mu}_{i_l,j_l}(1)$ and $\hat{\mu}_{i_l,j_l}(2)$ are the x and y components of the mean. This linear system is solved by weighted least squares with outlier rejection. The two equations of each pair are weighted by the amount of uncertainty of the respective optimal offset model, measured by the area of the covariance: $\left|\hat{\Sigma}_{i_l,j_l}\right|$. We also add two auxiliary equations $x_{i_1} = 0$ and $y_{i_1} = 0$ with large weight, as the system (10) is independent of the choice of the origin. The outlier rejection is implemented by removing pairs of equations corresponding to maximal error, until the bound of at most one pixel error is reached. The part configurations are obtained as connected components (CC) of a graph whose nodes are the features belonging to at least one of the spatial-pairs from the cluster and whose edges are the pairs themselves. The resulting configurations may then by used as 'hyper-features' for either detecting the learned objects in new images (e.g. using the ISM voting scheme proposed by [11]) or as input to subsequent invocations of the UFO in order to build even more complex features.



Method	Mean P_D for P_{FA}=0
Jaccard	43%
MI	41%
SC	38%
L2	31%
pLSA	24%
spectral	31%

Figure 4. **(a)** Summary of the comparison of UFO and baseline methods (pLSA, spectral) on the entire Caltech-101 dataset. For each method the bar-plots show a cumulative histogram of detection probability P_D at the point $P_{FA} = 0$ (no False Alarms) on the ROC. For example, bars placed between 0.2 and 0.3 count the number of classes for which the tested methods achieved at least 20% detection with no false alarms; the pLSA achieved this for 42 of the 101 classes, compared with 78 classes for UFO with MI consistency measure. We also analyzed the 27 classes with $P_D < 20\%$ for the JI consistency measure (the rightmost dark-red bar). Only 13 classes produced < 5 class examples as top scoring ones (getting 5 top examples at random in our 20% class setup has probability < 0.0003); **(b)** Mean P_D for $P_{FA} = 0$ on all Caltech-101 classes for each method; **(c)** Comparison on UIUC cars and horses datasets.

3. Experimental Results

To test the proposed approach we applied it on a wide range of classes from several datasets, including the entire Caltech-101, UIUC cars [1], horses [3], hand poses [16], a dataset of similar face views, and a dataset obtained from querying Google's image search. For every class from the Caltech-101, UIUC cars, and horses datasets, the experimental protocol was to take all the class images mixed with random selection of four times more background images from the Caltech backgrounds set. Thus in every unsupervised set there were only 20% class images. Figure 4(a,b) summarizes the comparison of the UFO algorithm with baseline approaches on the entire Caltech-101 dataset. The comparison of the different consistency measures dis-

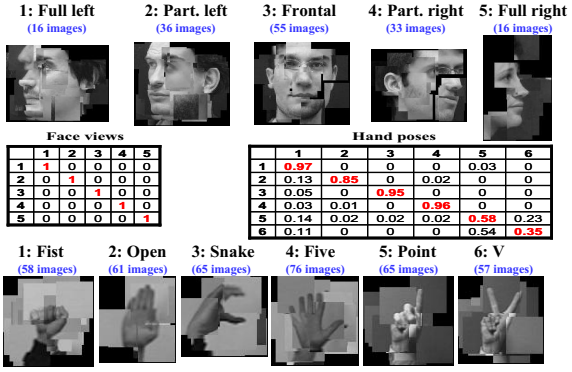


Figure 5. Summary of the multi-sub-class unsupervised separation experiment. The proposed method was applied for unsupervised multi-class separation of face views and hand poses datasets. The pLSA and spectral clustering baseline approaches performed significantly worse with at least 20% difference in mean accuracy. The images show examples of part configurations obtained for each sub-class.

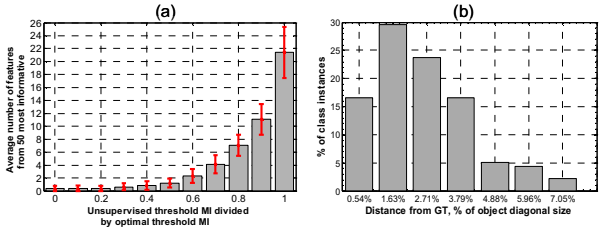


Figure 6. **(a)** Comparison of unsupervisedly learned parameters (using MI consistency measure) with optimal parameters obtained in supervised setting with respect to MI. The mean histogram was computed for 50 most informative features in 78 Caltech-101 classes for which the UFO succeeded ($P_D > 20\%$ for $P_{FA} = 0$). The rightmost bin corresponds to perfect match. The density remains almost unchanged when the histogram is computed for either 2, 10, and 100 most informative features; **(b)** Evaluation of car localization using the part configurations learned on UIUC cars dataset. The histogram shows the error distribution of a detected reference point with respect to manually marked ground truth.

cussed in section 2.2 is also included in the figure 4(a,b). The baseline methods are pLSA with 7 topics, and spectral clustering (to 7 clusters) of the normalized cross correlation of feature responses without parameter optimization. In all the experiments UFO also computed 7 clusters. To make a fair comparison, the UFO was applied with the selection of threshold parameters but not geometry, since both baseline approaches do not use geometry. Each of the compared approaches generates feature clusters (pLSA topics are soft feature clusters). The test for a good unsupervised feature cluster is the extent to which it corresponds to the unknown class. The score we have chosen to compare the different approaches, is the detection probability P_D at the point $P_{FA} = 0$ (no False Alarms) on the ROC. The reason is that applications that will use the proposed approach (or a baseline) for unsupervised extraction of class examples, will take top scoring examples and any errors in them will hinder subsequent performance. We also got very similar results in terms of the relative differences between the methods for the point of 80% precision. For each method and each class, the cluster with highest P_D for $P_{FA} = 0$ was taken into the comparison. The ROC for each pLSA topic was generated according to the topic probability in each image. For both the UFO and the spectral approach, the ROC for each cluster was generated by simply summing up the response vectors of features belonging to the cluster. For UFO, the response vectors were thresholded using the computed optimal detection thresholds (one for each feature) prior to the summation.

On figure 4c we show the results of the same comparison on the UIUC cars and the horses datasets. Only the 170 uncropped and unaligned test images from the UIUC cars dataset (mixed with random background images) were used in the experiment. Figure 6b illustrates the performance of car localization using the part configurations learned on the UIUC cars dataset. The part configurations were combined using ISM [11] and voted for a single reference point.

In addition, we have compared the percent of features originating in class images in the feature set returned by the UFO with the percent of such features in codebooks returned by the standard unsupervised feature selection and quantization methods, namely k-means and [9], that are extensively used in many current image classification approaches. The comparison showed that on our mix of 20% class 80% background images, the standard methods produce codebooks containing on average $22 \pm 8\%$ class features, while the UFO produces a set of features with $68 \pm 15\%$ class features (statistics computed on all Caltech-101 classes). Moreover, as can be seen in figure 2, the features selected by the UFO mostly come from the class object region of the class images, while it is likely that it is not so for the standard algorithms (mostly the class objects occupy less then 50% of the class images).

On two datasets, the hands (382 images, 6 poses, proposed by [16]) and the faces (156 images, 5 views), the proposed method was applied in an unsupervised multi-class manner in order to simultaneously separate and learn all the sub-classes. From the hand poses dataset only the test images were used. The resulting confusion matrices and the computed part configurations for the relevant clusters are shown in figure 5, showing perfect separation for face views and high performance for hand poses.

Figure 6a shows comparison of the parameters learned in the fully unsupervised setting by UFO with MI consistency measure, with ground truth most informative (maximal MI) parameters obtained with supervision. The comparison is

Figure 7. Summary of the experiment on 200 first images returned by "jaguar" Google's image search query. 30 of them contained (jaguar) animals the rest (jaguar) cars and noise. All processing was done in grayscale. All images were normalized to same vertical size. The green and the red boxes display first 20 images of two of the clusters returned by the UFO. The cluster associated with animals (green box), had 67% recall with 95% precision, and 83% recall with 72% precision in animal jaguar detection.

performed on the 78 Caltech-101 classes for which the UFO (with MI) exceeded 20% detection with no false alarms. Surprisingly, the unsupervised method learns globally optimal parameters for more then 40%, and near optimal parameters (MI ratio of $\geq$ 0.8) for about 80%, of the most discriminative features. Finally, figure 7 summarizes the results of an experiment of unsupervised clustering of images obtained from a query "jaguar" to Google's image search.

4. Discussion

The main novelty of the proposed approach is the discovery of multiple classification features and their detection parameters in the fully unsupervised setting by a global optimization of their joint consistency as a function of the detection parameters. The method can be applied to both unsupervised and (weakly) supervised learning tasks, as a method for reducing the dimensionality of the feature space by selecting and optimizing most discriminative features. It also proved useful as a method for unsupervised sub-class discovery. Automatic separation into meaningful sub-classes is an important tool for boosting object recognition performance, as it allows to model each sub-class individually. Future work includes extending the proposed method for optimizing descriptor combination parameters, testing additional consistency measures, and extending to the temporal domain for unsupervised discovery of consistent motion patterns, that could be applied to both action recognition and motion based object recognition.

Acknowledgment: This work was supported by EU IST Grant FP6-2005-015803.

References

[1] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *PAMI*, 2004.

[2] N. Ahuja and S. Todorovic. Discovering hierarchical taxonomy of categories and shared subcategories in images. *ICCV*, 2007.

[3] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. *ECCV*, 2002.

[4] L. Cao and L. Fei-Fei. Spatially coherent latent topic model for concurrent object segmentation and classification. *ICCV*, 2007.

[5] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. *ICCV*, pages 1816–1823, 2005.

[6] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. *CVPR (2)*, pages 264–271, 2003.

[7] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. *ECCV*, 2004.

[8] M. Fritz and B. Schiele. Towards unsupervised discovery of visual categories. *DAGM*, 2006.

[9] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. *ICCV*, 2005.

[10] L. Karlinsky, M. Dinerstein, D. Levi, and S. Ullman. Unsupervised classification and part localization by consistency amplification. *ECCV*, 2008.

[11] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. *ECCV*, 2004.

[12] L.-J. Li, G. Wang, and L. Fei-Fei. Optimol: automatic object picture collection via incremental model learning. *CVPR*, 2007.

[13] D. Liu and T. Chen. Semantic-shift for unsupervised object detection. *CVPR Workshop*, 2006.

[14] D. Liu and T.Chen. Unsupervised image categorization and object localization using topic models and correspondences between images. *ICCV*, 2007.

[15] N. Loeff, H. Arora, A.Sorokin, and D. Forsyth. Efficient unsupervised learning for localization and detection in object categories. *NIPS*, 2005.

[16] S. Marcel. Hand posture recognition in a body-face centered space. *Conference on Human Factors in Computer Systems (CHI)*, 1999.

[17] S. Nowozin, K. Tsuda, T. Uno, T. Kudo, and G. H. Bakir. Weighted substructure mining for image analysis. *CVPR*, 2007.

[18] T. Quack, V. Ferrari, B. Leibe, and L. V. Gool. Efficient mining of frequent and distinctive feature configurations. *ICCV*, 2007.

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

[20] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their localization in images. *ICCV*, pages 370–377, 2005.

[21] M. Varma and D. Ray. Learning the discriminative powerinvariance trade-off. *ICCV*, 2007.

[22] Y. Weiss. Segmentation using eigenvectors: A unifying view. *ICCV*, 1999.

[23] L. Zhu, C. Lin, H. Huang, Y. Chen, and A. Yuille. Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion. *ECCV*, 2008.

UFO: Supplementary material

1. Consistency measures and their bounds

In this section we provide the proofs of the claims, stated in the paper, concerning the lower bounds for consistencies measured between pairs of features. The proofs are given following the formulations of corresponding claims for three out of the four consistency measures, considered in the paper, namely: Jaccard Index, Mutual Information and Suspicious coincidence. The bound for the case of L2 obviously follows from the triangular inequality.

1.1. Jaccard Index (JI)

Claim 2.1

$$JI\left[F_i;F_j\right] \geq \frac{JI\left[F_i;C\right]+JI\left[F_j;C\right]-1}{\frac{1}{JI[F_i;C]}+\frac{1}{JI[F_j;C]}-1} \tag{1}$$

Proof.
Here for convenience we treat F_i, F_j and C (binary row vectors of length N in the paper) as sets of indices of ones in them. Bar above a set means its complement.
Consider:

$$\begin{aligned} |F_i \cap C| &= JI[F_i,C]\cdot|F_i \cup C| \geq JI[F_i,C]\cdot|C| \\ |F_j \cap C| &= JI[F_j,C]\cdot|F_j \cup C| \geq JI[F_j,C]\cdot|C| \end{aligned} \tag{2}$$

Hence:

$$|\bar{F_j} \cap C| = |C \setminus (F_j \cap C)| = |C| - |F_j \cap C| \leq |C| - JI[F_j,C]\cdot|C| \tag{3}$$

And thus:

$$\begin{aligned} |F_i \cap F_j| &\geq |(F_i \cap C)\cap(F_j \cap C)| = |(F_i \cap C)\setminus(\bar{F_j}\cap C)| \geq \\ &\geq |F_i \cap C| - |\bar{F_j}\cap C| \geq JI[F_i,C]\cdot C - (|C| - JI[F_j,C]\cdot|C|) = \\ &= (JI[F_i,C]+JI[F_j,C]-1)\cdot|C| \end{aligned} \tag{4}$$

Moreover:

$$|C| \geq |F_i \cap C| = JI[F_i,C]\cdot|F_i \cup C| \geq JI[F_i,C]\cdot|F_i| \tag{5}$$

which means:

$$|F_i| \leq \frac{|C|}{JI[F_i,C]} \tag{6}$$

Thus:

$$|F_i \setminus C| \leq |F_i| - |C| \leq \frac{|C|}{JI[F_i,C]} - |C| \tag{7}$$

Similarly:

$$|F_j \setminus C| \leq |F_j| - |C| \leq \frac{|C|}{JI[F_j, C]} - |C| \tag{8}$$

Hence:

$$\begin{aligned} |F_i \cup F_j| &\leq |C \cup (F_i \setminus C) \cup (F_j \setminus C)| \leq |C| + |F_i \setminus C| + |F_j \setminus C| \leq \\ &\leq |C| + \frac{|C|}{JI[F_i, C]} - |C| + \frac{|C|}{JI[F_j, C]} - |C| = \\ &= \left(\frac{1}{JI[F_i, C]} + \frac{1}{JI[F_j, C]} - 1 \right) \cdot |C| \end{aligned} \tag{9}$$

Combining all the above expressions we finally get:

$$\begin{aligned} JI\left[F_i; F_j\right] &= \frac{|F_i \cap F_j|}{|F_i \cup F_j|} \geq \\ &\geq \frac{(JI[F_i, C] + JI[F_j, C] - 1) \cdot |C|}{\left(\frac{1}{JI[F_i, C]} + \frac{1}{JI[F_j, C]} - 1 \right) \cdot |C|} = \\ &= \frac{JI[F_i, C] + JI[F_j, C] - 1}{\frac{1}{JI[F_i, C]} + \frac{1}{JI[F_j, C]} - 1} \end{aligned} \tag{10}$$

■

1.2. Mutual Information (MI)

Claim 2.2
Assuming that F_i and F_j are conditionally independent given class C, then:

$$MI\left[F_i; F_j\right] \geq MI\left[F_i; C\right] + MI\left[F_j; C\right] - H\left(C\right) \tag{11}$$

Proof.
Here for convenience we treat F_i, F_j and C (binary row vectors of length N in the paper) as binary random variables with joint distribution given by empirical distribution computed on the vectors (this is the ML approximation).
Consider:

$$\begin{aligned} H(F_i, F_j) \leq H(F_i, F_j, C) &= -\sum_{F_i, F_j, C} P(F_i, F_j, C) \log(P(F_i, F_j, C)) = \\ &= -\sum_{F_i, F_j, C} P(F_i, F_j, C) \log(P(F_i, F_j | C) \cdot P(C)) = \\ &= -\sum_{F_i, F_j, C} P(F_i, F_j, C) \log(P(F_i | C) P(F_j | C) \cdot P(C)) = \\ &= -\sum_{F_i, F_j, C} P(F_i, F_j, C) \log(P(F_i | C)) - \sum_{F_i, F_j, C} P(F_i, F_j, C) \log(P(F_j | C)) - \sum_{F_i, F_j, C} P(F_i, F_j, C) \log(P(C)) = \\ &= -\sum_{F_i, C} P(F_i, C) \log(P(F_i | C)) - \sum_{F_j, C} P(F_j, C) \log(P(F_j | C)) - \sum_{C} P(C) \log(P(C)) = \\ &= H(F_i | C) + H(F_j | C) - H(C) \end{aligned} \tag{12}$$

Thus:

$$\begin{aligned} MI(F_i,F_j) &= H(F_i)+H(F_j)-H(F_i,F_j) \geq H(F_i)+H(F_j)-[H(F_i|C)+H(F_j|C)-H(C)] = \\ &= [H(F_i)-H(F_i|C)]+[H(F_j)-H(F_j|C)]-H(C) = \\ &= MI(F_i,C)+MI(F_j,C)-H(C) \end{aligned} \tag{13}$$

■

1.3. Suspicious Coincidence (SC)

Claim 2.3

Assume the events $F_i = 1$ and $F_j = 1$ are conditionally independent given the event $C = 1$, then:

$$SC\left[F_i;F_j\right] \geq SC\left[F_i;C\right] \cdot SC\left[F_j;C\right] \cdot P(C=1) \tag{14}$$

Proof.

Here for convenience we treat F_i, F_j and C (binary row vectors of length N in the paper) as binary random variables with joint distribution given by empirical distribution computed on the vectors (this is the ML approximation).

$$\begin{aligned} SC\left[F_i;F_j\right] &= \frac{P(F_i=1,F_j=1)}{P(F_i=1)P(F_j=1)} = \\ &= \frac{P(F_i=1,F_j=1|C=1)P(C=1)+P(F_i=1,F_j=1|C=0)P(C=0)}{P(F_i=1)P(F_j=1)} \geq \\ &\geq \frac{P(F_i=1,F_j=1|C=1)P(C=1)}{P(F_i=1)P(F_j=1)} = \\ &= \frac{P(F_i=1,C=1)P(F_j=1,C=1)P(C=1)}{P(F_i=1)P(C=1)P(F_j=1)P(C=1)} = \\ &= SC\left[F_i;C\right] \cdot SC\left[F_j;C\right] \cdot P(C=1) \end{aligned} \tag{15}$$

■

The chains model for detecting parts by their context

Leonid Karlinsky Michael Dinerstein Daniel Harari Shimon Ullman
{leonid.karlinsky,michael.dinerstein,danny.harari,shimon.ullman}@weizmann.ac.il
Weizmann Institute of Science, Rehovot 76100, Israel

Abstract

Detecting an object part relies on two sources of information - the appearance of the part itself, and the context supplied by surrounding parts. In this paper we consider problems in which a target part cannot be recognized reliably using its own appearance, such as detecting low-resolution hands, and must be recognized using the context of surrounding parts. We develop the 'chains model' which can locate parts of interest in a robust and precise manner, even when the surrounding context is highly variable and deformable. In the proposed model, the relation between context features and the target part is modeled in a non-parametric manner using an ensemble of feature chains leading from parts in the context to the detection target. The method uses the configuration of the features in the image directly rather than through fitting an articulated 3-D model of the object. In addition, the chains are composable, meaning that new chains observed in the test image can be composed of sub-chains seen during training. Consequently, the model is capable of handling object poses which are infrequent, even non-existent, during training. We test the approach in different settings, including object parts detection, as well as complete object detection. The results show the advantages of the chains model for detecting and localizing parts of complex deformable objects.

1. Introduction

Two main sources of information can be used for detecting a part of interest of an object: the appearance of the part itself, and the context supplied by surrounding parts. In the current paper we focus on detecting the location of a part of a deformable object using the context supplied by other parts of the object (denoted 'internal context' below). We also show that the same approach can be used to detect complete objects using the internal context of their parts. The detection of object parts and objects are interrelated tasks, since object parts can be used for detecting the objects, and the internal context of the object can be used to detect parts. Below, we briefly review existing recognition systems from the point of view of how they specify the relative locations of object parts, and their ability to detect a part of interest specified to the system, using the surrounding context.

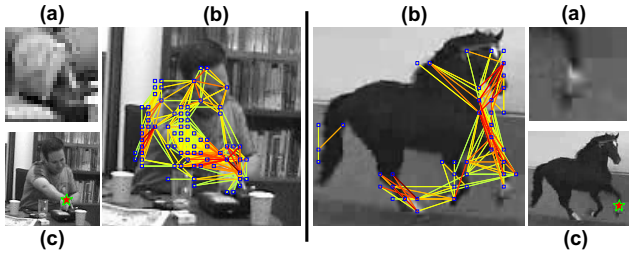


Figure 1. Overview of the proposed approach. **(a)** The goal is to detect object parts (e.g., hand, hoof) of deformable objects (e.g., person, horse) in still images, when the parts of interest are not recognizable on their own; **(b)** During testing, the inference graph of the chains model is constructed for each test image and used to compute the chains model posterior for the detected part. The edges are color-coded according to their weights (red=high). **(c)** Star marks the global maximum of the posterior.

Existing approaches can be divided into several groups depending on the way they model geometric relationships between parts. The simplest methods do not explicitly model feature geometry; for example, the so called bag-of-features approaches, such as [25, 3]. Another popular approach models feature geometry by a star model that allows each object part to have a region of tolerance relative to some object reference point (star center) [16, 8, 13]. The constellation model [9] allows more flexible joint Gaussian relationships between object parts. Finally, there are models that try to capture more complex feature geometry, such as feature hierarchy [6, 23], articulated spring models [7, 21, 10, 2, 14, 5] and level-set methods [22]. In terms of object and part localization, the bag-of-features methods provide only a rough estimate of object location as a cluster of relevant features. The star and the constellation models provide more specific object and part localization, but they are particularly suitable for semi-rigid objects and they do not focus on the detection of specific parts of interest. Feature hierarchies provide more flexible geometry, but are much more complex to learn, and in current approaches

[6, 23] are still restricted to semi-rigid objects. Finally, articulated models and level-set methods assume a known apriori model of object deformation (with the exception of [21]). Such models may encounter problems when the deformations are complex and only partially covered by the training examples. Moreover, most existing approaches (except [3]) train a relatively small feature codebook (e.g., obtained through quantization of randomly sampled patch descriptors), and all geometric approaches train a fixed model for the spatial arrangement of these features. One problem with using a feature codebook and a fixed spatial model is that they are trained to fit well the majority of the training examples, while infrequent object poses in the training set are usually underrepresented. This is reflected both in the lack of features for these poses, as well as in the spatial model that does not fit well the infrequent poses. This problem becomes particularly significant when the task is the detection of parts of a highly deformable object, such as hand detection, since many of the possible object poses are infrequent in the training set.

The goal of the proposed approach is to obtain precise localization of parts of highly deformable objects. An example goal would be detecting the hand location in images of people captured from a distance or in a pose that does not allow detecting the hand reliably based on its own appearance. The approach is different from past approaches in several respects. First, the model focuses on detecting parts of interest, whereas many recognition models focus on the detection of the object itself, and ambiguous parts may be missed during recognition. Second, the way it captures geometric relations between features is neither through a semi-rigid model, such as a star or a constellation, nor through an articulated model fitted to the majority of the training set. Instead, we propose a new non-parametric probabilistic model (the chains model) and its associated inference algorithm, which can efficiently handle complex non-rigid feature configurations. This is obtained by using multiple feature chains leading from potential source features to the target part. Third, instead of using a relatively small feature codebook, all of the relevant features extracted from the training images are retained and efficiently used during the testing. Consequently, the model is able to represent information from all training images, including images containing rare poses. Fourth, all relative pair-wise spatial configurations of the relevant features extracted from the training images are retained and used to build the geometric model for the features on the test image. As a result, even rare poses are not under-represented, both in terms of the features as well as in terms of their geometric model. An overview of the approach is given in figure 1.

The remainder of the paper is organized as follows. Section 2 describes the proposed approach and its implementation details. Section 3 describes the experimental validation. Summary and discussion are provided in section 4.

2. Method

This section describes the proposed 'chains model', the associated feature selection approach and the implementation details of the method. For clarity, the hand detection task will be used for illustrating the proposed method. In addition, we describe the use of the method for full object detection in section 2.3.

Intuitively, the chains model describes an assembly of feature chains of arbitrary length that start at some known reference point on the object, such as an automatically detected face, and terminate at the detection target, such as the hand. The chains model is a generative model for the set of features extracted from the test image. It generates some of the features by creating a feature chain between the source (face) and the target (hand) parts. The rest of the features are generated independently out of the 'world' distribution, i.e. marginal distribution over all images. During inference, we marginalize over all possible feature chains between the face and the hand. This is shown to be equivalent to summing over simple paths in a graph connecting the features extracted from the test image, with edge weights that are related to the so-called suspicious coincidence between the neighboring features. We show how we can approximate the solution of the inference problem by a simple computation using the adjacency matrix of the constructed graph.

In the algorithm, the method for estimating different empirical probabilities of the features is similar to the one used by [3], and is based on Kernel Density Estimation (KDE) over neighbors computed using the Approximate Nearest Neighbor (ANN) technique. The details of this computation are described in section 2.5.

2.1. Feature extraction

For each image I_n (training or test), the face location is detected using an off-the-shelf method (we used [13]). Denote the location of the detected face by $\ell_n^f = \left(x_n^f, y_n^f\right)$ and its horizontal size of by s_n. All other distances, offsets and patch sizes are computed relative to s_n, eliminating the dependence on the scale of the person. We extract a dense set of features centered on all edge points of I_n (detected by [18]) sampled with a spacing of $\min(0.2 \cdot s_n, 4)$ pixels. Denote the set of extracted features by $F_n = \left\{ f_n^j = \left\langle SIFT_n^j, X_n^j \right\rangle | j = 1, 2, \ldots, k_n \right\}$, where k_n is the total number of features extracted from image I_n, $SIFT_n^j$ is the SIFT descriptor (128-D row vector) [17] of feature j from image I_n, computed for $s_n \times s_n$ sized patch centered at location X_n^j (2-D row vector). For training images only, we also denote the marked hand location in image I_n by $\ell_n^h = \left(x_n^h, y_n^h\right)$. For training and quantitative evaluation, ground truth hand locations were semi-automatically iden-

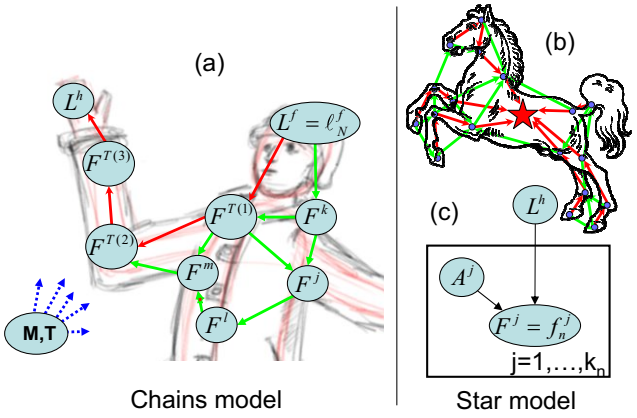


Figure 2. **(a)** Chains model applied for part detection. The unobserved variable L^h is the location of the target part (e.g. hand). $L^f = \ell_N^f$ is the observed location of the reference part (e.g. face) in image I_N. The edges symbolize the chains 'feature graph' constructed over the set of observed features $\left\{F^j = f_N^j\right\}$. Unobserved variables T and M (affecting all the nodes) select a simple path T (red) of length M on the graph. Features not on this path are generated from their respective 'world' distributions. During inference we marginalize over T and M, summing over paths on the graph going from the reference part to each candidate location of the target part. **(b)** Chains model applied for full object detection as an extended star model. **(c)** Star model. Unobserved binary variables A^j (one per feature), control whether the feature is generated with respect to the star center or from its 'world' distribution.

tified in some of the hand detection experiments. In these experiments a colored glove was used for fast and simple hand tracking, using a color blob tracker with manual initialization. Although color was used to obtain the ground truth, both training and testing of the chains model were performed on grayscale images in all of the experiments. In addition, experiments on external datasets (of [5] and [10]) verified the proposed approach on people without gloves.

2.2. Feature selection for target (hand) detection

The training stage proceeds as follows. The algorithm receives a set of training frames in which a person performs various hand movements. At each frame, both the hand and the face locations are marked by a single point each, as described in Section 2.1. No motion information is used. The feature selection procedure then retains 100 features with highest likelihood ratio from each training image. The ratio Λ_n^j for feature f_n^j in image I_n is computed as: $\Lambda_n^j = \frac{P\left(L^h=\ell_n^h, F^j=f_n^j, L^f=\ell_n^f\right)}{P\left(L^h\neq\ell_n^h, F^j=f_n^j, L^f=\ell_n^f\right)}$ where the random variables L^h and L^f designate the hand and face locations respectively, and F^j is a random variable for the j-th observed feature. Throughout the paper we will use a shorthand form of variable assignments, e.g., $P\left(\ell_n^f\right)$ instead of $P\left(L^f = \ell_n^f\right)$.

As we found in our experiments, this feature selection criterion favors features whose presence correlates with the target part location (e.g., many arm features were automatically selected for hand detection). The probabilities used to compute Λ_n^j are estimated as described in section 2.5.

2.3. The chains model

Model definition. The chains model is a generative model for the features extracted from the test image I_N. The observed variables of the model are the extracted features $F_N = \left\{F^j = f_N^j = \left\langle SIFT_N^j, X_N^j \right\rangle\right\}$ and the detected face (source part) location $L^f = \ell_N^f$. The upper index j of the features is according to some arbitrary order of the feature extractor (the order is immaterial for the subsequent computation). The unobserved variables are the location of the hand (target part) denoted by L^h, an arbitrary length M and an ordered list of feature indices denoted by $T = \langle T(1), \ldots, T(M)\rangle$, where $1 \leq T(i) \leq k_N$. k_N is the number of features extracted from image I_N. The list T defines a simple chain of features (i.e., without repetitions). The joint distribution of all these variables is taken to be:

$$P_{CH}\left(M, T, L^h, \ell_N^f, \left\{f_N^j\right\}\right) = P(M,T) \cdot P\left(L^h \left| f_N^{T(M)}\right.\right) \cdot \prod_{m=1}^{M-1} P\left(f_N^{T(m+1)} \left| f_N^{T(m)}\right.\right) \cdot P\left(f_N^{T(1)} \left| \ell_N^f\right.\right) \cdot P\left(\ell_N^f\right) \cdot \prod_{j\notin T} P_W\left(f_N^j\right) \quad (1)$$

In this expression, along the chain we use the conditional probability of each feature given its chain predecessor, and for features outside the chain we take the independent product of their 'world probabilities'. The distribution $P(M,T)$ is taken to be uniform over simple chains of length up to $M = 4$ and such that locations of neighboring chain features are neither too close nor too far apart (i.e for any $1 \leq m < M$: $\frac{1}{2}s_N \leq \left\|X_N^{T(m+1)} - X_N^{T(m)}\right\| \leq \frac{3}{2}s_N$). $P_W\left(f_N^j\right)$ is the 'world' probability for generating the feature f_N^j, i.e., the probability to observe f_N^j either on the object or in the background. $P_W\left(f_N^j\right)$ is used to generate all of the features that lie outside the chain T. $P\left(f_N^{T(1)} \left| \ell_N^f\right.\right)$ is the conditional probability for transition from the face to the first feature on the chain, and $P\left(L^h \left| f_N^{T(M)}\right.\right)$ is the probability of transition from the last feature on the chain to the hand. All of these probabilities will be explicitly defined in section 2.5. We next divide the joint P_{CH} by the term $\prod_{f_N^j \in F_N} P_W\left(f_N^j\right)$, which is fixed for a given image I_N, thus rewriting the joint in an equivalent form as:

$$P_{CH}\left(M,T,L^h,\ldots\right) \propto P\left(M,T\right)\cdot\frac{P\left(f_N^{T(1)},\ell_N^f\right)}{P_W\left(f_N^{T(1)}\right)}\cdot$$
$$P\left(L^h\left|f_N^{T(M)}\right.\right)\cdot\prod_{m=1}^{M-1}\frac{P\left(f_N^{T(m+1)},f_N^{T(m)}\right)}{P_W\left(f_N^{T(m+1)}\right)\cdot P\left(f_N^{T(m)}\right)} \quad (2)$$

Here, $P\left(f_N^{T(m)}\right)$ is the marginal of the pairwise probability: $P\left(f_N^{T(m+1)},f_N^{T(m)}\right)$. The chains model is schematically illustrated in figure 2a.

Inference. The goal of the inference is to compute the posterior distribution over the hand locations:

$$P_{CH}\left(L^h \mid \ell_N^f,\left\{f_N^j\right\}\right) \propto \sum_{M,T} P_{CH}\left(M,T,L^h,\ell_N^f,\left\{f_N^j\right\}\right)$$

To compute this posterior, we define a directed weighted 'feature graph' G_N. The nodes of G_N are $\left\{f_N^j\right\}$, and edges connect all pairs of features f_N^j and f_N^k for which $\frac{1}{2}s_N \leq \left\|X_N^k - X_N^j\right\| \leq \frac{3}{2}s_N$. The weight on the directed edge from node f_N^k to node f_N^j is $w_{jk} = \frac{P\left(f_N^j,f_N^k\right)}{P_W\left(f_N^j\right)\cdot P\left(f_N^k\right)}$ (a ratio similar to 'suspicious coincidence'). The adjacency matrix A_N of G_N is a matrix in which entry jk is equal to w_{jk} if the directed edge from f_N^k to f_N^j exists in G_N and is zero otherwise. Using this notation, the marginal over chains of length smaller or equal to M can be computed as:

$$P_{CH}\left(L^h\left|\ell_N^f,\left\{f_N^j\right\}\right.\right) \propto \sum_{M,T} P_{CH}\left(M,T,L^h,\ldots\right) =$$
$$D\cdot C^*\cdot S \approx D\cdot C\cdot S,\ \ where\ \ C=\sum_{m=0}^{M-1}\left(A_N\right)^m;$$
$$D=\left[P\left(L^h\left|f_N^1\right.\right),\cdots,P\left(L^h\left|f_N^{k_N}\right.\right)\right];$$
$$S=\left[\frac{P\left(f_N^1,l_N^f\right)}{P_W\left(f_N^1\right)},\cdots,\frac{P\left(f_N^{k_N},\ell_N^f\right)}{P_W\left(f_N^{k_N}\right)}\right]^T \quad (3)$$

where rectangular brackets denote row vectors. Here S is the source vector containing edge weights between the source part and its neighbors; C^* is the chains matrix, where entry jk is the sum of products of weights on all simple paths going from node k to node j in the graph G_N; and D is the target vector containing the edge weights between the target part and its neighbors. The matrix C is used as an easily computable approximation of C^*, obtained by summing over all paths of length $\leq M$, and not only over simple paths. The reason to use an approximation is that computing k best simple paths is costly [12], and in our experiments we have found the simple approximation to be sufficient. Entry jk of the adjacency matrix $\left(A_N\right)^t$ is the sum of the weights of all directed paths of length t going from node k to node j, where the weight of a path is a product of weights of edges on it. Consequently, finding L^h that maximizes eq. 3 is equivalent to max-marginal inference over the posterior of the chains model. Thus the inference in the model is simple and straightforward, involving only a few matrix multiplications. Interestingly, it is also possible to compute in a similar manner the Maximum A-Posteriori (MAP) inference for the chains model, using the observation that $\sqrt[r]{a^r+b^r} \underset{r\to\infty}{\to} \max\left(a,b\right)$. Raising each entry of D, A_N and S to a sufficiently large power r, obtaining the respective D_r, A_N^r, S_r, and $C_r = \sum_{m=0}^{M-1}\left(A_N^r\right)^m$, and maximizing $\sqrt[r]{D_r\cdot C_r\cdot S_r}$ becomes equivalent to a MAP inference for the chains model. We have experimentaly observed that using a combination of MAP and max-marginal inference attains the best performance. We maximize $D\cdot\sqrt[r]{C_r\cdot S_r}$, where the r-th root here is taken entry-wise. This is equivalent to maximizing a sum over all of the features in terms of their votes for possible hand locations, where each feature vote is determined by the single maximal chain reaching it from the source part. We use this method with $r=10$ in all our experiments. Finally, it is also possible to use the same set of selected features for detecting multiple object parts by replacing the vector D with a matrix in which each row corresponds to a part being detected.

Star as a special case. Noteworthy, setting $C=I$ in eq. 3 reduces the chains model to a variant of the popular star model illustrated in figure 2c. This variant allows each feature to independently choose to be generated either by the object or the world distributions. The proof of this reduction and the detailed description of the star model variant are beyond the scope of the current paper and are provided in supplementary material. In section 3 we quantitatively show the advantage of the chains model over the star model for highly deformable part detection.

2.4. Full object detection

The chains model can also be applied with some modifications to the task of full object detection. Similar to part detection, the model can use the test image features to point at the object location, either directly, or via intermediate feature chains (see figure 2b). In this section we describe how to apply the chains model to object detection. In the final discussion we consider the integration of objects and object parts detection within the chains model.

For detection at the object level, we eliminate the detection of the chain's source part, thereby allowing an arbitrary source location. As we no longer have a reference scale (obtained from the source size), the features' SIFT descriptors are computed for patches of a fixed predefined size (20×20 pixels). Instead, multi-scale support is achieved by testing each image at multiple scales. The training stage runs on a set of positive images (with objects roughly aligned in lo-

Probability ($KDE(V,S)$)	Query vector (V)	Queried set (S)
$P_W\left(f_N^j\right)$	$SIFT_N^j$	$\left\{SIFT_{t_i}^k \mid 1 \le i \le R,\ all\ k\right\}$ [1,2,9]
$P\left(f_N^j\right)$	$SIFT_N^j$	$\left\{SIFT_{t_i}^k \mid 1 \le i \le R,\ selected\ k\right\}$ [1,2,9]
$P\left(O, f_n^j\right)$	$SIFT_N^j$	$\left\{SIFT_{t_i}^k \mid 1 \le i \le R,\ all\ k\ ,\ O_{t_i} = O\right\}$ [1,3,9]
$P\left(\ell_N^f, f_N^j\right)$	$\left[SIFT_N^j, \alpha_N \cdot \left(X_N^j - \ell_N^f\right)\right]$ [4]	$\left\{\left[SIFT_{t_i}^k, \alpha_{t_i} \cdot \left(X_{t_i}^k - \ell_{t_i}^f\right)\right] \mid 1 \le i \le R,\ selected\ k\right\}$ [1,4,5,9]
$P\left(f_N^k, f_N^j\right)$	$\left[SIFT_N^k, SIFT_N^j, \alpha_N \cdot \left(X_N^k - X_N^j\right)\right]$ [4]	$\left\{\left[SIFT_{t_i}^m, SIFT_{t_i}^l, \alpha_{t_i} \cdot \left(X_{t_i}^m - X_{t_i}^l\right)\right] \mid 1 \le i \le R,\ selected\ m\ and\ l,\ \frac{1}{2}s_{t_i} \le \left\|X_{t_i}^m - X_{t_i}^l\right\| \le \frac{3}{2}s_{t_i}\right\}$ [1,4,9]
$P\left(L^h, f_N^j\right)$	$\left[SIFT_N^j, \alpha_N \cdot \left(L^h - X_N^j\right)\right]$ [4]	$\left\{\left[SIFT_{t_i}^k, \alpha_{t_i} \cdot \left(\ell_{t_i}^h - X_{t_i}^k\right)\right] \mid 1 \le i \le R,\ selected\ k\right\}$ [1,4,6,7,9]
(1) $P\left(\ell_n^h, f_n^j, \ell_n^f\right)$ (2) $P\left(L^h \neq \ell_n^h, f_n^j, \ell_n^f\right)$	$\left[SIFT_n^j, \alpha_n \cdot \left(X_n^j - \ell_n^f\right)\right]$ [4]	$\left\{\left[SIFT_{t_i}^k, \alpha_{t_i} \cdot \left(X_{t_i}^k - \ell_{t_i}^f\right)\right] \mid 1 \le i \le R,\ all\ k\right\}$ [1,4,8,9]

[1] 'all k' - all of the features extracted from the training images; 'selected k' - only the selected features; 'selected m and l' - pairs of selected features.
A selected feature $f_{t_i}^k$ is a valid neighbour of a query feature f_N^j iff f_N^j is within $\le \theta_{t_i}^k$ from $f_{t_i}^k$, where the threshold $\theta_{t_i}^k$ is such that $P\left(L^h \neq \ell_{t_i}^h, f_{t_i}^k, \ell_{t_i}^f \middle| \text{computed using only neighbors that passed } \theta_{t_i}^k\right) = 0$. Features f_N^j without any valid neighbors are not considered.

[2] For both $P_W\left(f_N^j\right)$ and $P\left(f_N^j\right)$, the location component of the feature, i.e. X_N^j, is assumed to be distributed uniformly.

[3] O is binary and $O_{t_i} = true$ means that I_{t_i} is a positive training image.

[4] $\alpha_* = 1/s_*$ is a scale factor, where s_* is the horizontal size of the face in image I_*. $(*)$ stands for n, N, t_i

[5] To limit the set of features reachable directly from the source part, we set $P\left(\ell_N^f, f_N^j\right) = 0$ when $\alpha_N \cdot \left(X_N^j - \ell_N^f\right) > 4$.

[6] To limit the set of features that can vote for the target part, we set $P\left(L^h \middle| f_N^j\right) = 0$ when $\alpha_N \cdot \left(L^h - X_N^j\right) > 1.5$.

[7] For speed, we query using only $SIFT_N^j$ and add the offset component directly to the KDE.

[8] Define $HandDist_{t_i}^k = \left\|\alpha_{t_i} \cdot \left(\ell_{t_i}^h - X_{t_i}^k\right) - \alpha_n \cdot \left(L^h - X_n^j\right)\right\|$. Restrict to neighbors with $\begin{cases} HandDist_{t_i}^k \le 0.75, \text{for (1)} \\ HandDist_{t_i}^k > 0.75, \text{for (2)} \end{cases}$

[9] $\{I_{t_1}, \ldots, I_{t_R}\}$ denotes the set of the training images.

Table 1. Computations of all the required model probabilities using ANN queries. A query for a vector V in a set S returns a set of k neighbors, from which the estimated probability is computed by Gaussian Kernel Density Estimation - $KDE(V,S)$.

cation and scale), and negative background images. In the feature selection stage, features are selected by their log-likelihood ratio defined as: $\Lambda_n^j = \log \frac{P\left(O=true, f_n^j\right)}{P\left(O=false, f_n^j\right)}$ where O is a binary variable designating the presence of the object in an image. We retain up to 100 features with the highest positive log-likelihood ratio from each positive training image. By allowing chains to originate at any node (not just in ℓ_n^f), i.e. substituting $P\left(f_n^{k_n}, \ell_n^f\right)$ with $P\left(f_n^{k_n}\right)$ in eq. 3, detection of the full object and its parts can be obtained without requiring a pre-detected reference part.

2.5. Probability estimates by ANN-based KDE

We next show how to efficiently compute, from the training data, all of the probabilities used in the model and during the feature selection. These probabilities are computed using Kernel Density Estimation (KDE). Given a set of samples from a distribution of interest $\{Y_1, \ldots, Y_R\}$, a Gaussian KDE estimate $P(Y)$ for the probability of a new sample Y can be approximated as:

$$P(Y) \propto \frac{1}{R} \cdot \sum_{r=1}^{R} \exp\left(-\|Y - Y_r\|^2 \Big/ 2\sigma^2\right) \approx \frac{1}{R} \cdot \sum_{Y_r \in NN} \exp\left(-\|Y - Y_r\|^2 \Big/ 2\sigma^2\right) \quad (4)$$

where NN is the set of nearest neighbors of Y within the given set of samples. When the number of samples R is large, brute-force search for the NN set becomes infeasible. Therefore, we use Approximate Nearest Neighbor (ANN) search (in the implementation of [19]) to compute the KDE. For a given training image I_n or test image I_N, table 1 describes the computation of all the required model probabilities. In each case, the computation is based on an ANN query. The query returns a set of K nearest neighbors, and the Gaussian KDE is applied to this set to get the estimated probability. The accuracy of all of the computations depends on the size of the NN set. In our experiments we found that it is sufficient to use $K = 25$ (which was used throughout). We used $\sigma = \begin{cases} 0.5\,, & for\ P\left(f_n^k, f_n^j\right) \\ 0.35\,, & otherwise \end{cases}$. Different value for $P\left(f_n^k, f_n^j\right)$ is due to the higher dimensionality of the associated ANN query, which leads to accuracy loss of the ANN implementation of [19].

3. Results

For the task of hand detection, the chains model was compared with two types of models: a star model (described in the supplementary) and state of the art methods specialized for human pose estimation [10, 2, 14, 5]. The star model we used is a high performance method, which gives state of the art results on several standard classification datasets. The comparisons were performed on several datasets including three collected by us, two 'sign language' datasets collected by [5], and the 'Buffy' dataset collected

	Chains				Star			
Scenario + Dataset	1 max	2 max	3 max	4 max	1 max	2 max	3 max	4 max
self trained (ST)	87	93.9	95.2	95.9	46.9	65	75.7	81.2
self trained (Horse)	63.1	90.3	93.4	94.6	50.2	61.6	68	74
person cross-generalization (PCG)	86.9	93.5	95.4	96.1	57.9	74.1	81.9	85.5
full generalization (ST)	73.2	86.6	90	92.2	37.6	52	66.9	74.7

Table 2. Comparison of hand detection results of the chains model and the star model applied on ST, PCG and Horse datasets in various test scenarios. The columns titled '1 max', '2 max', etc., measure the detection performance within the top k local maxima after non-maximal suppression with the radius of 0.25 face width. In all of the scenarios, the standard deviation of the chains performance is about 4%. In the full generalization scenario the performance of [2] is 38.6%. In the horse hoof detection experiment, the reason for the 63% in the '1 max' vs. 90.3% in '2 max', is that the system had a hard time distinguishing between the front-left and right hoofs. This distinction was found confusing to humans as well.

by [10]. Details of the datasets are described in section 3.1. All the compared approaches, except [5], were applied to each frame independently, without using any tracking or motion information. Our method was applied to grayscale versions of the images, while most of the compared approaches [10, 14, 5] use color information. In all of the experiments, the chains model detection of the hand was considered correct if it was within one hand width ($1/2$ face width) from the ground truth location. On our datasets the hand detection accuracy was evaluated for the right hand only. On all of the external datasets (not created by us), hand detection accuracy was evaluated for both right and left hands. For the external datasets, the compared methods' results were taken out of the original publications. We also compared with [2] on our own dataset using the original code kindly provided by the authors. The hand detection experiments are described in section 3.1 and the results are summarized in tables 2 and 3.

Evaluation of the full object detection was conducted on two datasets: UIUC cars [1] and Weizmann Horses [4]. We compare our performance to the star model (supplementary) and to state of the art methods [20, 16, 15, 11, 24] on these datasets. The experiments are described in section 3.2 and the results are summarized in table 4. Qualitative results for both hand and object detection are shown in figure 3 and in the supplementary material.

3.1. Hand Detection

Our datasets. For our experiments, we collected three datasets denoted by 'ST', 'PCG', and 'Horse'. The ST dataset consists of four movies (about 3000 frames each) of four different people performing various hand movements. The PCG dataset has 12 movies of 12 different people (about 600 frames each). The Horse dataset consists of a single movie of a running horse (1331 frames). For the

	Chains				[5]	[14]	[10]	[2]
Setting	1 max	2 max	3 max	4 max	1 max	1 max	1 max	1 max
BBC news	96.8	99	99.5	99.6	95.6	86.37[1]	-	-
5 signers	84.9	92.8	95.4	96.7	-	78.95[1]	-	-
Buffy	47	53.9	56.7	59.3	-	39.2[1]	56[1,2]	46.1

Table 3. Comparison of hand detection results of the chains model and past methods applied on sign language and Buffy datasets. (1) [10, 14] reported average of detections of all 6 upper body parts. We believe this can be considered an upper bound on the hand detection percentage since the method by Andriluka et al. [2] that achieves 46.1% correct lower arm detection obtained 73.5% average detection for all the 6 body parts. (2) Result of [10] was obtained on 88% of frames in the dataset using foreground segmentation, 41% - without using foreground segmentation.

Horse dataset the ground truth was obtained by manually marking the front left hoof and the head locations. Experiment on the Horse dataset demonstrates that, unlike methods tailored to the human body, exactly the same chains model algorithm (using the same parameters) can be used to detect deformable parts of both humans and animals. The experiments on our datasets were divided into three different test scenarios. The ST and the Horse datasets were used for testing in the 'self-trained' scenario. The first 1000-1500 frames of each movie sequence in the dataset were used for training and the rest were used for testing. The PCG dataset was used to test cross-generalization between different people in different clothing. The experiment was performed in a leave-one-out manner: each of the twelve movie sequences present in the dataset was used for testing while the remaining sequences were used for training. Finally, the 'full-generalization' scenario included training on all frames of the PCG dataset, and testing on the ST dataset training frames (having the widest variety of poses). This scenario tested both the generalization to new people and clothes, as well as to new unseen background. Two of the four people in ST dataset were present (in different clothing) in the PCG dataset and were excluded from the PCG training in their respective tests. The method of [2] attained 38.5% hand detection rate on the ST dataset. Its performance is to be compared with the results of the 'full-generalization' scenario of our method since it was not trained on any of our training data. Although [2] was not given the detected face locations, it detected the face correctly in all but a very few test frames.

Sign language datasets. These two datasets include a 'BBC news' dataset containing a full BBC news sequence with a single signer, and a '5-signers' dataset - a collection of frames from 5 news sequences, with different signers. Both datasets were collected by [5]. On the BBC news dataset the experiment was performed in a 'self-trained' fashion. We have manually marked the ground truth hand locations for the first 3115 frames. The first 600 frames

(a)

Method	[20]	[16]	[11, 15]	Chains model (UIUC / Caltech-101 trained)	Star model (UIUC / Caltech-101 trained)
Result	99.9%	97.5%	98.5%	99.5% / 93.4%	98% / 89.2%

(b)

Method	[24]	[11]	Chains model	Star model
Result	95.7%	93.9%	97.5 ± 1%	87.8 ± 3%

Table 4. Comparison between different methods by precision-recall at EER: (a) UIUC car testset; (b) Weizmann Horses dataset, the larger advantage of the chains may be related to fact that horses are somewhat more flexible than rigid 3-D objects, such as cars.

were used for training and the remaining 2515 frames were used for testing. This is compatible with the experiment of [5], who also trained and tested on the same sequence. Notably, [5] used color and tracking cues, and quantitatively tested only on 296 frames. On the 5-signers dataset, the experiment was in the 'full-generalization' mode. All 195 frames of the 5-signers dataset were used for testing the model trained on the first 600 frames of the BBC news dataset. We consider our setting to be more challenging than the one used in [14], who trained and tested on the same signers in their 5-signers dataset experiment.

Buffy dataset. The 'Buffy' dataset was collected by [10]. It contains frames selected from 4 episodes of the popular series 'Buffy the Vampire Slayer'. The methods we compared with used frames from the first 3 episodes for testing (308 images) and the 4th episode is reserved for training. In our experiment on this dataset, we have trained the chains model on our PCG dataset plus 5 additional short movie sequences of standing people we shot in our lab. The reason for using additional movies is that the PCG dataset contained only sitting people, while 'Buffy' dataset consists mostly of standing people. Ground truth face locations were used. We do not consider this to be a significant advantage since the top competing method [2] achieved 96% correct head detection and thereby mostly had the correct head locations as well. Since most of the Buffy frames are very dark and of low contrast, histogram equalization was applied to all of the frames.

3.2. Full object detection

Car Detection. The UIUC car dataset contains side-view images with one or more cars. Using 550 positive and 500 negative training images and 200 cars in 170 test images of the UIUC single scale dataset, the chains model achieved the recall of 99.5% at precision-recall Equal Error Rate (EER). In addition, to test cross-dataset generalization, we replaced the original UIUC training set with Caltech-101 cars, still achieving a competitive recall of 93.4% at EER.

Horse Detection. The Weizmann Horses dataset consists of 323 side-on horse images in natural environments. The horses vary in breeds, textures, and articulations. The chains model achieved $97.5 \pm 1\%$ recall at EER computed by a 3-fold cross-validation (with 500 negative images).

Figure 3. Qualitative examples, see supplementary for more examples and movies. Demonstrates hand detection for people in various environments, poses and clothing. Object detection is illustrated by horse detection, chains posterior is overlayed on the image (red = high posterior), star marks the global maximum.

4. Discussion

The chains model described in this paper locates parts of interest in a robust and precise manner, even when the surrounding context is highly variable and deformable. Three properties of the chains model make it particularly useful for this task. First, evidence is combined in a flexible and comprehensive manner: for a given part, information from all other parts which can supply evidence regarding the part's location, either directly or indirectly via other parts,

is used. Second, the computation efficiently combines evidence in parallel from a large number of chains and subchains. Finally, it is a non-parametric model, in which the required probabilities are derived for a new image using all the information stored during training. Moreover, the probabilities are estimated using nearest neighbors, which can handle effectively feature configurations that are relatively infrequent in the training data. Another useful property of the model is that it uses image features directly, without relying on domain-specific parametric models, such as models that use articulated skeletons of humans and animals.

Recognition at the full object level is naturally included in the scheme, since chains in the model lead to a target, which can be either a part or the full object. At the object level, the chains model can be viewed as an extended star model, and consequently the performance in object detection is competitive with similar top-performing schemes in the domains tested. The chains model can then use partial detection obtained during the first recognition phase as sources for the recognition of additional, difficult to detect, parts, embedded in deformable context. The detection of ambiguous parts by the chains model can also be naturally combined with other recognition schemes: partial results obtained by other recognition schemes can be augmented by serving as sources for disambiguating chains.

The training of the chains model for the part detection task requires training images with labeled reference and target parts. In practice, these can often be obtained automatically; for example, for hand detection, we used a face detector and a colored glove tracking. More generally, the model can be trained to generalize from cases in which the target part is recognizable on its own, to cases in which it is not.

Regarding future directions, the chains model may be extended to provide a useful component for scene recognition at multiple levels: chains can connect parts and objects, as well as related objects in a scene, providing a vehicle for the use of context in the recognition of scenes, objects and parts. Finally, it would be interesting to extend the chains model to deal with video input by developing spatio-temporal chains, traversing features within a frame as well as across nearby frames in a video sequence.

References

[1] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *PAMI*, 2004.

[2] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. *CVPR*, 2009.

[3] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. *CVPR*, 2008.

[4] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. *ECCV*, 2002.

[5] P. Buehler, M. Everingham, D. Huttenlocher, and A. Zisserman. Long term arm and hand tracking for continuous sign language tv broadcasts. *BMVC*, 2008.

[6] B. Epshtein and S. Ullman. Semantic hierarchies for recognizing objects and parts. *CVPR*, 2007.

[7] P. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures. *CVPR*, 2000.

[8] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. *CVPR*, 2008.

[9] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. *CVPR*, 2003.

[10] V. Ferrari, M. Marin, and A. Zisserman. Progressive search space reduction for human pose estimation. *CVPR*, 2008.

[11] J. Gall and V. Lempitsky. Class-specific hough forests for object detection. *CVPR*, 2009.

[12] J. Hershberger, M. Maxely, and S. Suri. Finding the k shortest simple paths: A new algorithm and its implementation. *ACM Trans. Algorithms*, 2007.

[13] L. Karlinsky, M. Dinerstein, D. Levi, and S. Ullman. Unsupervised classification and part localization by consistency amplification. *ECCV*, 2008.

[14] M. Kumar, A. Zisserman, and P. Torr. Efficient discriminative learning of parts-based models. *ICCV*, 2009.

[15] C. Lampert, M. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. *CVPR*, 2008.

[16] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *IJCV*, 2008.

[17] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[18] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 2004.

[19] D. Mount and S. Arya. Ann: A library for approximate nearest neighbor searching. *CGC 2nd Annual Workshop on Comp. Geometry*, 1997.

[20] J. Mutch and D. Lowe. Multiclass object recognition with sparse, localized features. *CVPR*, 2006.

[21] D. Ramanan, D. A. Forsyth, and K. Barnard. Building models of animals from video. *PAMI*, 2006.

[22] T. Schoenemann and D. Cremers. Matching non-rigidly deformable shapes across images: A globally optimal solution. *CVPR*, 2008.

[23] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. *CVPR*, 2005.

[24] J. Shotton, A. Blake, and R. Cipolla. Multiscale categorical object recognition using contour fragments. *PAMI*, 2008.

[25] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their localization in images. *ICCV*, 2005.

Supplementary for 'The chains model for detecting parts by their context'

Leonid Karlinsky Michael Dinerstein Danny Harari Shimon Ullman
{leonid.karlinsky,michael.dinerstein,danny.harari,shimon.ullman}@weizmann.ac.il
Weizmann Institute of Science, Rehovot 76100, Israel

This manuscript contains the supplementary material for the paper: 'The chains model for detecting parts by their context'. Section 1 describes the movies, provided within the current supplementary, showing examples of deformable part detection on various datasets and scenarios. Section 2 describes in details the star model being a special case of the chains model and provides the proof of the corresponding reduction. And finally, figures 2, 3, and 4 depict more additional examples of hand and car detection on '5-signers', 'Buffy', and UIUC cars datasets respectively. The aforementioned datasets are described in the paper.

1. Movies

Five movies are provided together with this supplementary material. They contain sample results of different deformable part detection experiments. The quantative evaluation of the results for all of these experiments can be found in the paper. To view the movies it is necessary to have the Xvid codec installed. The codec can be found within K-lite codec pack:

http://www.codecguide.com/download_kl.htm

The movies illustrating the results of various experiments performed on our datasets are:

1. **'self_trained_ST_Xvid.avi'** - shows a sample result of the self-trained experiment on the ST dataset.
2. **'person_cross_generalization_PCG_Xvid.avi'** - contains a sample result of the person-cross-generalization experiment on the PCG dataset.
3. **'full_generalization_ST_Xvid.avi'** - provides example results of the full-generalization experiment on the ST dataset.

In these movies, only the right hand of the person is being detected. A green star marks the detected location of the right hand. Red color inside the star means that the star shows the location of the first maximum of the chains model posterior, and green color means that the star depicts the location of the second maximum. The frame-rate of the movies was increased in order to meet the size restrictions of the supplementary submission regulations.

The movie **'self_trained_Horse_Xvid.avi'** shows the result of the horse hoof detection experiment. The green star marks the detected location of the front left hoof of the horse (meaning front left from the point of view of the horse). Again red color inside the star means 1^{st} maximum, green - the 2^{nd} maximum, and (in case of the horse only) no color inside the star means the 3^{rd} maximum.

Finally, the movie **'self_trained_BBC_news_Xvid.avi'** demonstrates results of the self-trained experiment on the 'BBC news' dataset. Both right and left hands are being detected. The detected location of the right hand is marked as above, and the detection of the left hand is marked by a yellow star. The colors inside the stars are as above, red means the 1^{st} maximum, and green (or yellow) - the 2^{nd} maximum.

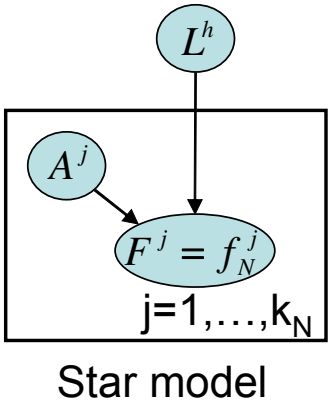


Figure 1. **Star model.** The observed variables $\left\{F^j = f_N^j\right\}$ are the features extracted from the test image I_N. The unobserved variable L^h is the location of the star center (the detection target). Unobserved binary variables A^j (one per feature), control whether the feature is generated with respect to the star center or from its 'world' distribution.

2. Appendix: Star as a special case of the chains model

We show here that a variant of the popular star model is in fact a special case of the chains model presented in the paper. This variant allows each feature to independently choose to be generated by the object or the world distributions. Consider the following generative star model, also illustrated graphically in figure 1:

$$\begin{aligned} &P\left(L^h, \left\{F^j = f_N^j\right\}, \left\{A^j\right\}\right) = \\ &P\left(L^h\right) \cdot \prod_{j=1}^{k_N} P\left(A^j\right) \cdot \prod_{j=1}^{k_N} P\left(F^j = f_N^j \left| A^j, L^h\right.\right) \end{aligned} \tag{1}$$

where L^h is the unobserved star center (detection target) location and we assume $P\left(L^h\right)$ is uniform, $\left\{f_N^j \mid 1 \le j \le k_N\right\}$ are the features extracted from the test image I_N, and A^j is an unobserved binary variable with $P\left(A^j = 1\right) = \alpha$ that controls whether the j-th feature F^j is generated by the object or the world distributions, namely:

$$P\left(F^j = f_N^j \left| A^j, L^h\right.\right) = \begin{cases} P^*\left(F^j = f_N^j \left| L^h\right.\right) & \text{if } A^j = 1 \\ P_W\left(F^j = f_N^j\right) & \text{if } A^j = 0 \end{cases}$$

where $P_W\left(F^j = f_N^j\right)$ is the 'world' probability of the feature f_N^j that is defined in the paper, and the probability P^* is defined as:

$$\begin{aligned} &P^*\left(F^j = f_N^j \left| L^h\right.\right) \propto P^*\left(F^j = f_N^j, L^h\right) = \\ &P\left(L^h \left| F^j = f_N^j\right.\right) \cdot P_{obj}\left(F^j = f_N^j\right) \end{aligned} \tag{2}$$

where $P\left(L^h \left| F^j = f_N^j\right.\right)$ is defined as above and $P_{obj}\left(F^j = f_N^j\right)$ is the 'object' distribution of the feature f_N^j. It is proportional to $P\left(F^j = f_N^j\right)$ defined in the paper when there is no observed source part L^f, and to $P\left(F^j = f_N^j, L^f = \ell_N^f\right)$ when there is one (see paper section 2.5 and table 1 for details).

We show below that this star model is equivalent to using the chains model with chains of length one. In the star model, inference for the posterior of the center location involves the following marginalization over unobserved

variables A^j (here equivalence $\sim$ is in terms of maximizing the posterior):

$$\begin{array}{l}
\log P\left(L^h \left| \left\{F^j = f_N^j\right\}\right.\right) = \\
\log\left[P\left(L^h\right) \cdot \prod_{j=1}^{k_N} \sum_{A^j=0}^{1} P\left(A^j\right) \cdot P\left(F^j = f_N^j \left| A^j, L^h\right.\right)\right] \sim \\
\sum_{j=1}^{k_N} \log\left[\sum_{A^j=0}^{1} P\left(A^j\right) \cdot P\left(F^j = f_N^j \left| A^j, L^h\right.\right)\right] = \\
\sum_{j=1}^{k_N} \log\left[\alpha \cdot P^*\left(F^j = f_N^j \left| L^h\right.\right) + (1-\alpha) \cdot P_W\left(F^j = f_N^j\right)\right] = \\
\sum_{j=1}^{k_N} \log\left[\frac{P^*\left(F^j = f_N^j \left| L^h\right.\right)}{\gamma \cdot P_W\left(F^j = f_N^j\right)} + 1\right] + \sum_{j=1}^{k_N} \log\left[(1-\alpha) \cdot P_W\left(F^j = f_N^j\right)\right] \sim \\
\sum_{j=1}^{k_N} \log\left[\frac{P^*\left(F^j = f_N^j \left| L^h\right.\right)}{\gamma \cdot P_W\left(F^j = f_N^j\right)} + 1\right] \underset{(*)}{\approx} \sum_{j=1}^{k_N} \frac{P^*\left(F^j = f_N^j \left| L^h\right.\right)}{\gamma \cdot P_W\left(F^j = f_N^j\right)} \sim \\
\sum_{j=1}^{k_N} P\left(L^h \left| F^j = f_N^j\right.\right) \cdot \frac{P_{obj}\left(F^j = f_N^j\right)}{P_W\left(F^j = f_N^j\right)} = P_{CH}\left(L^h \left| \left\{F^j = f_N^j\right\}\right.\right)
\end{array} \tag{3}$$

where $\sum_{j=1}^{k_N} \log\left[(1-\alpha) \cdot P_W\left(F^j = f_N^j\right)\right]$ is a constant independent of L^h and $\gamma = (1-\alpha)/\alpha$. The transition $(*)$ follows from $\log(1+\varepsilon) \approx \varepsilon$ for $\varepsilon << 1$, and from

$$\frac{P^*\left(F^j = f_N^j \left| L^h\right.\right)}{\gamma \cdot P_W\left(F^j = f_N^j\right)} << 1 \tag{4}$$

since we can take γ to be large, reflecting the fact that most features are generated from the background distribution. The distribution $\mathrm{P}_{CH}\left(L^h \left| \left\{F^j = f_N^j\right\}\right.\right)$ is exactly the marginal distribution of the chains model restricted to chains of length one. The approximate chains model posterior distribution of L^h is $D \cdot C \cdot S$ (see the paper, page 4, eq. 3). For a star model $C = I$, therefore the star model posterior is equal $D \cdot S$. In the paper we quantitatively show the advantage of the chains model over the star model for highly deformable part (e.g., hand) detection.

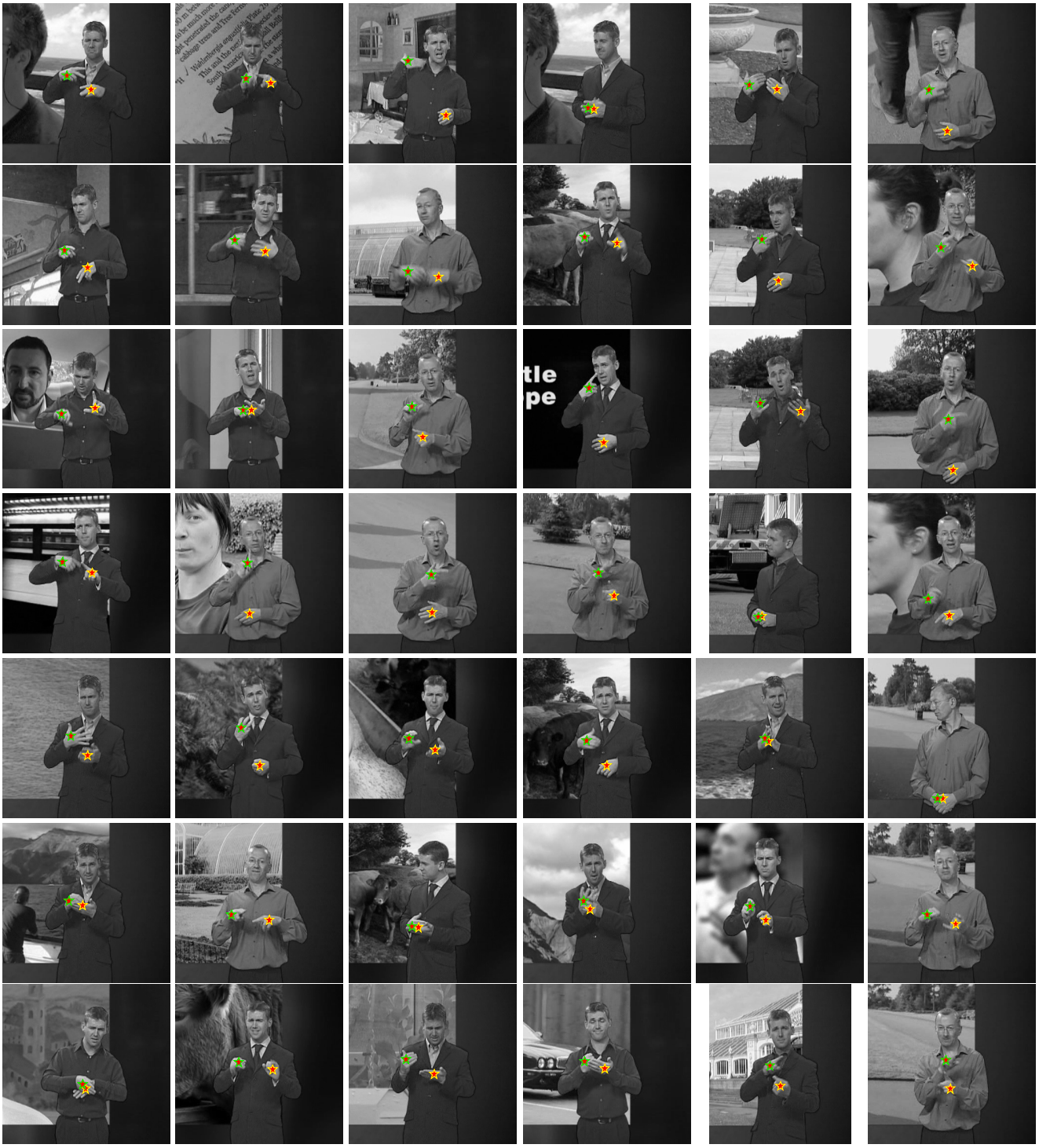


Figure 2. Additional examples of hand detection on 5-signers dataset. A green star marks the detected location of the right hand of a person, the detected location of the left hand is marked by a yellow star.

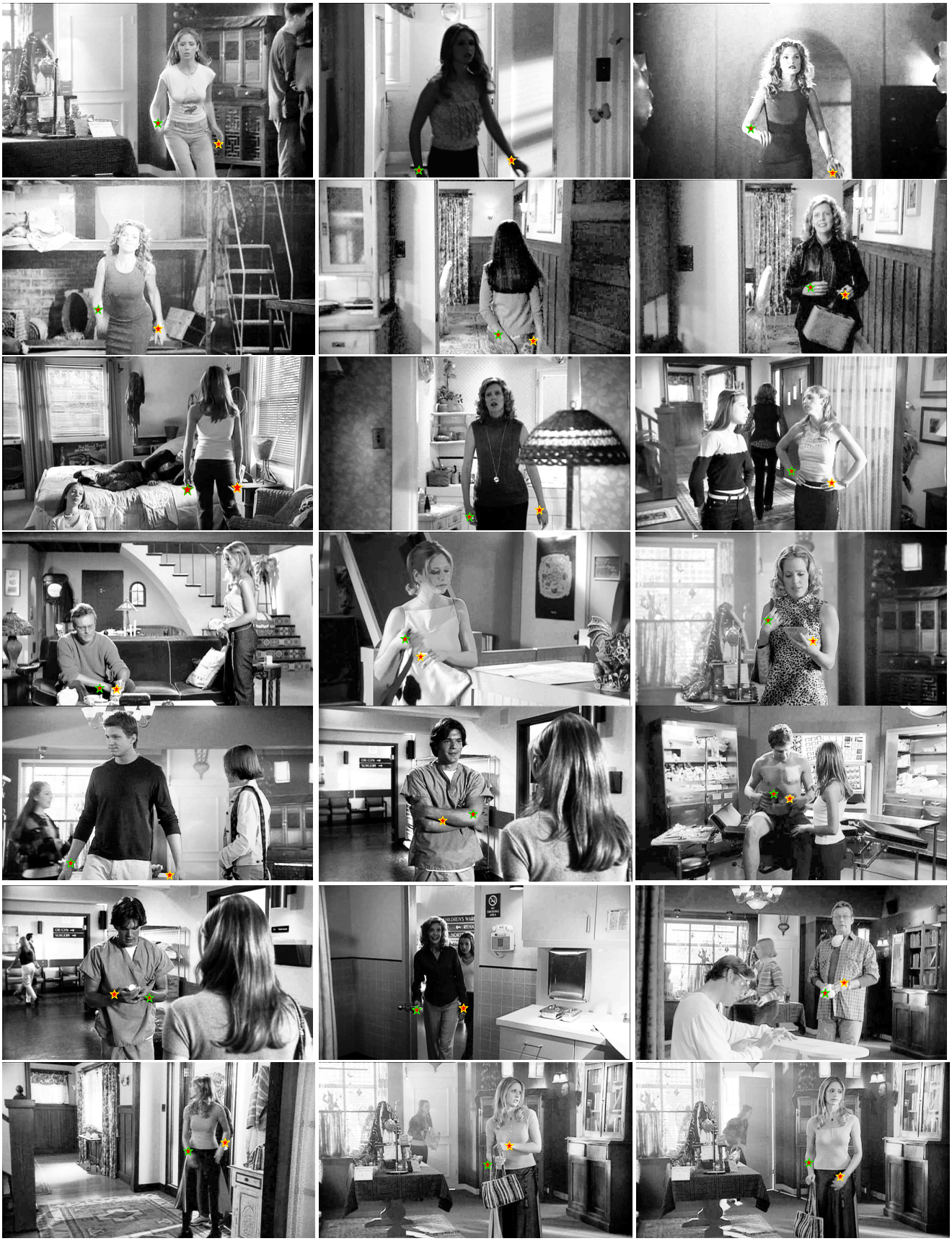


Figure 3. Additional examples of hand detection on Buffy dataset. A green star marks the detected location of the right hand of a person, the detected location of the left hand is marked by a yellow star.

Figure 4. Additional examples of car detection on UIUC cars dataset. A yellow box marks the detected location of the car, the ground truth location is marked by a red box.

Identifying similar transitive actions in single images

Anonymous ECCV submission

Paper ID 105

Abstract. This paper presents an approach for the recognition of similar actions, which include an actor, a tool and in some cases a recipient of the action, using only single images as input. Examples in the current study include drinking, toasting, talking on the phone, wearing glasses, etc. These actions are similar in terms of body pose, the objects involved are typically small and partially occluded, and the background scene can be arbitrary. Due to these factors, it is difficult to apply standard approaches for object recognition to recognize these actions as objects. We propose an approach that applies a two-stage interpretation procedure to each training or test image. The first stage produces accurate detection and localization of the relevant body parts of the actor. The second stage extracts features from locations anchored to the detected body parts, and uses these features together with the relative configuration of these parts in order to recognize the action. During training, this allows the automatic discovery and construction of implicit non-parametric models for different important aspects of the actions, such as relevant objects, types of grasping, and hand-object and body part configurations. During testing, this allows the approach to extract distinctive features specifically from restricted image regions that contain the relevant information for recognizing the action. As a result, we eliminate the need to have a priori models for relevant objects, and become capable of learning from action labels only, without labeling of the body parts and relevant objects. Focusing in a body-anchored manner on the relevant regions makes it possible to automatically learn the distinctions between highly similar actions, increases efficiency, and considerably enhances recognition results.

1 Introduction

This paper deals with the problem of recognizing transitive actions in single images. *Transitive actions* are actions involving an actor, a tool, and in some cases, a recipient of the action. Some simple examples are drinking from a glass, talking on the phone, eating from a cup with a spoon, and brushing teeth. The more popular type of actions, covered in the majority of computer vision action recognition literature [1], are *intransitive actions*, which typically involve only an actor who performs some (repetitive) sequence of movements, such as walking, jumping, and sitting down. Such intransitive actions are characterized

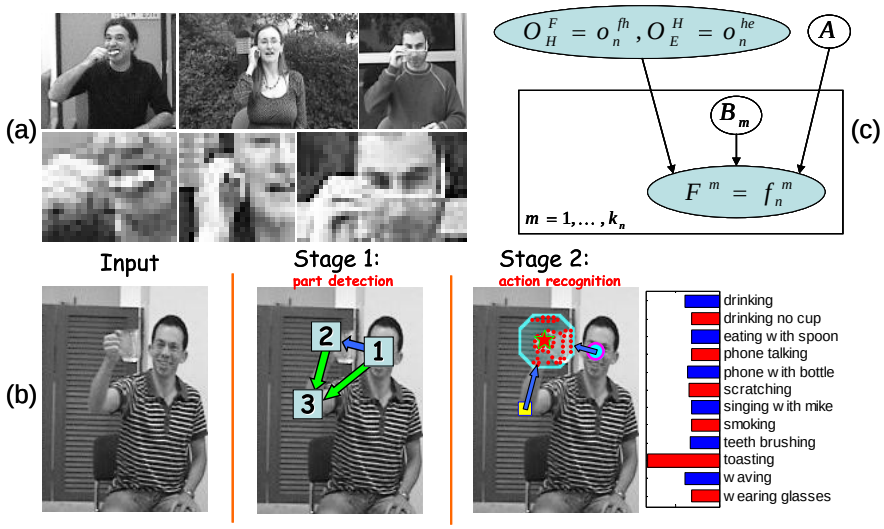


Fig. 1. (a) Examples of similar transitive actions identifiable by humans from single images (brushing teeth, talking on a cell phone and wearing glasses). (b) Illustration of a single run of the proposed two-stage approach. In the first stage the parts are detected in the face→hand→elbow order. In the second stage we apply both action learning and action recognition using the configuration of the detected parts and the features anchored to the hand region; the bar graph on the right shows relative log-posterior estimates for the different actions. (c) Graphical representation (in plate notation) of the proposed probabilistic model for action recognition (see section 2.2 for details).

by movement and therefore temporal information (a short video sequence) is required in order to recognize them [2–5]. In contrast, the transitive actions are more characterized by the pose of the actor, the tool she/he is holding, and the presence of the action recipient, and thus in many cases such actions can be readily identified by human observers from only a single image (see figure 1a).

A number of approaches have been developed to deal with dynamic action recognition from short video sequences. Different approaches in this category used different levels of specificity of the spatio-temporal information recovered from the dynamic scene including bag of spatio-temporal features, e.g. [2–4], a spatio-temporal pyramid [6], space-time shapes [5] tracking body parts of the actor and objects [1, 7], and different types of grasping dynamics [8].

Only a few approaches to date have dealt with action recognition in single images, or the so called static action recognition. [9] studied the recognition of sports actions using the pose of the actor. [10] used scene interpretation in terms of objects and their relative configuration to distinguish between different sporting events such as badminton and sailing. [11] recognized static intransitive actions such as walking and jumping based on a human body pose represented by a variant of the HOG descriptor. Finally, the most detailed static scheme to date [7] recognized static transitive sports actions, such as the tennis forehand and the volleyball smash, by using a full body mask, bag of features for describing scene

context, and the detection of the objects relevant for the action, such as bats, balls, etc. [7] used GrabCut [12] to extract the body mask and fully supervised training for the a priori known relevant objects and scenes.

In this paper we consider the task of differentiating between similar types of transitive actions, such as smoking a cigarette, drinking from a cup, eating from a cup with a spoon, talking on the phone, etc., given only a single image as input. The lack of temporal information precludes the use of approaches relying on the scene dynamics [2–6, 8]. The similarity between the body poses in such actions creates a difficulty for approaches that rely on pose analysis [9, 11, 7]. The relevant differences between similar actions in terms of the actor body configuration can be at a fine level of detail. Therefore, one cannot rely on a fixed pre-determined number of configuration types [9, 11, 7]; rather, one needs to be able to make as fine discriminations as required by the task. In many cases, it is necessary to detect fine details of information such as the accurate relative offset between the hand and the face, the presence and type of the object in the hand, type of grasping, face orientation and expression, etc. (figure 1a). Recovering details at this level requires accurate localization of body parts, such as faces, hands and limbs. However, all current action recognition approaches which employ body-parts detection use temporal information (tracking) in order to detect the parts [1], and the approach we propose is the first to employ flexible body parts (hands, elbows) detection in single images.

Crucial information for action identification may be sublte and difficult to detect. Objects participating in different actions may be very small, occupying only a few pixels in a low resolution image (spoon, cigarette). In addition, these objects may be unknown a priori, such as in the natural case when the learning is weakly supervised, i.e. we know only the action label of the training images, while the body parts and participating objects are not annotated and cannot be independently learned as in [10, 7]. Similarly, the relation between the object and relevant body parts is important for identifying the performed action. This was demonstrated by [7], who describe the position of the object relative to the observer using rough distance and direction from the body's centroid. However, the object position relative to the parts may differ only slightly between different actions (e.g. smoking and brushing teeth differ only in the type of grasping). In such cases, very accurate detection of relevant body parts (such as hands) and of grasping types are necessary to make the correct distinction between the actions. In addition, the background scene, used by [7, 10] to recognize sports actions and events, is irrelevant for most types of transitive actions of interest, and cannot be directly utilized.

In terms of datasets, most of the aforementioned action recognition approaches were applied to datasets of distinct intransitive actions, such as the KTH [13] and Weizmann actions [5] datasets. Such actions do not require a high degree of interpretation detail and may be recognized by detecting representative human poses specific to the action, or by a bag-of-features collection of spatio-temporal features. Only recently, a Hollywood movie actions dataset was introduced by [6]. Although it still contains mostly distinct intransitive actions,

it is more challenging than its predecessors because of high variability due to the uncontrolled and unstructured nature of the video sequences from which it was constructed. Hence, it will be difficult to apply the fine-detail body part interpretation based approaches to this dataset, as it will require the detection of human parts and objects from a very diverse set of scales and viewing angles. Handling this variability is beyond the capability of the current state-of-the-art body interpretation techniques [14–16]. Finally, [7] proposed a set of distinct transitive actions shot over a static roughly uniform background.

The main contribution of this paper is an approach for recognizing and distinguishing between similar transitive actions in single images, while trained using weak supervision of action labels only (without labeling of the body parts and relevant objects). In both the learning and the test settings, the approach applies two-stage interpretation steps to each (training or testing) image. In general, the first stage produces accurate detection and localization of body parts, and the second extracts and uses features from locations anchored to body parts. In the implementation of the first stage, the face is detected first, and its detection is extended to accurately detect the locations of the elbow and the hand of the actor. In the second stage, the relative part locations and the hand region are analyzed for action related learning/recognition. During training, this allows the automatic discovery and construction of implicit non-parametric models for different important aspects of the actions, such as accurate relative part locations, relevant objects, types of grasping, and accurate hand-object configuration models. During testing, this allows the approach to focus on the small image regions that contain all of the relevant information for recognizing the action. As a result, we eliminate the need to have a priori models for relevant objects and become capable of weakly supervised learning of the actions. Focusing in a body-anchored manner on the relevant regions not only increases efficiency, but also considerably enhances recognition results. The approach is visually summarized in figure 1b. In the results section 3 we show that employing standard state-of-the-art object recognition techniques in order to recognize and distinguish static actions without focusing on the relevant parts such as the hand and the face may be problematic. An additional contribution is a dataset for static action recognition containing 10 people performing 12 similar hand-near-face transitive actions, namely drinking, talking on the phone, scratching, toasting, waving, brushing teeth, smoking, wearing glasses, eating with a spoon, singing with a microphone, and also drinking without cup and making a phone call with a bottle. All actors appear in natural settings against different cluttered backgrounds containing multiple unrelated changes, such as people passing by.

The rest of the paper is organized as follows. Section 2 describes the proposed approach and its implementation details. Section 3 describes the experimental validation. Summary and discussion are provided in section 4.

2 Method

In this section we present our approach to the recognition of similar transitive actions from single images. We focus on a group of similar hand-near-face actions,

Fig. 2. (a) Examples of the computed binary masks (cyan) for searching for elbow location given the detected hand and face marked by a red-green star and magenta rectangle respectively. The yellow square marks the detected elbow; (b) Additional examples of parts detection.

in which the hand is sufficiently close to the face and performs some activity, e.g. drinking, smoking or eating with a spoon.

As discussed above, the approach proceeds in two main stages. The first stage is body interpretation, which is by itself a sequential process. First, the person is detected by detecting her/his face which can be viewed from different angles, including both in-plane and out-of-plane rotation (up to full profile view). Next, the face detection is extended to detect the hands and elbows of the person. This is achieved in a non-parametric manner by following chains of features connecting the face to the part of interest (hand, elbow), by an extension of [16]. In the second stage, features gathered from the hand region and the relative locations of the hand, face and elbow, are used to model and recognize the static action of interest. The first stage of the process, dealing with the face, hand and elbow detection, is described in section 2.1. The static action modeling and recognition is described in section 2.2 and additional implementation details are provided in section 2.3.

2.1 Body parts detection

Body parts detection in static images is a challenging problem, which has recently been addressed by several groups [14–18]. The hardest parts to detect are the most flexible parts of the body - the lower arms and the hands. This is due to large pose and appearance variability and the low resolution typical to these parts. In our approach, we have adopted an extension of the non-parametric method for the detection of general parts of deformable objects recently proposed by [16]. The method of [16] can operate in two modes. The first mode is used for the independent detection of sufficiently large and rigid objects and object parts, such as the face. The second mode allows for the extention of some existing detections to additional detection of other parts that are more difficult to detect independently, such as hands and elbows. The method extends the star

model by allowing features to vote for the detection target either directly, or indirectly, via features participating in feature-chains going towards the target. In the independent detection mode, these feature chains may start anywhere in the image, whereas in the extension mode these chains must originate from already detected parts. The method employs a non-parametric generative probabilistic model, which can efficiently learn to detect any part from a collection of training sequences with marked target (e.g., hand) and source (e.g., face) parts (or only the target parts in the independent detection mode). The mathematical details of this model are described in [16]. In our approach, the method of [16] is used to detect the face in its independent detection mode, and to detect the hand and the elbow by chains-extension from the face detection (treated as the source part). The method is trained using a collection of short video sequences, each having the face, the hand and the elbow marked by three points. The code for the method was kindly provided by the authors of [16] and extended to allow restricted detection of dependent parts, such as hand and elbow.

In some cases, the elbow is more difficult to detect than the hand, as it has less structure. For each (training or test) image I_n, we therefore constrain the elbow detection by a binary mask M_n of possible elbow locations given the detected hand location x_n^h. Denoting the relative hand-elbow offset in image I_k by o_k^{he}, and the hand-face offset by o_k^{fh}, from all the training images $\{I_k\}$ we retrieve the set $\left\{ o_k^{he} \;\middle|\; \left| o_k^{fh} - o_n^{fh} \right| \leq r_1 \right\}$ of all the hand-elbow offsets for which the hand-face offsets are sufficiently close to the hand-face offset o_n^{fh} detected in the test image I_n ($r_1 = 0.25$ face width in our implementation) . The retrieved offsets are then combined to form the binary mask M_n, by assigning 'one' to all pixels $\{o_k^{he} + x_n^h\}$ and dilating the resulting mask by a disk with a small radius r_2 ($r_2 = 0.5$ face width in our implementation). Some examples of the detected faces, hands and elbows on various test images are shown in figure 2b. Figure 2a shows some conditional elbow-given-hand masks for some of the poses.

2.2 Modeling and recognition of static actions

Given an image I_n (training or test), we first introduce the following notation (lower index refers to the image, upper indices to parts). Denote the instance of the action contained in I_n by a_n (known for training and unknown for test images). Denote the detected locations of the face by x_n^f, the hand by x_n^h, and the elbow by x_n^e. Also denote the width of the detected face by s_n. Throughout the paper, we will define the values of various size and distance parameters in s_n units, in order to eliminate the dependence on the scale of the person performing the action. For many actions, and especially for the set of 'hand-near-face' actions defined above, most of the discriminating information about the action resides in the hand region. We represent this information by a set of rectangular patch features extracted from this region. Each feature is represented by a corresponding 128-dimensional SIFT descriptor. All features are taken from a circular region with a radius $0.75 \cdot s_n$ around the hand location x_n^h. From this region we extract $s_n \times s_n$ pixel rectangular patch features centered at all Canny

edge points sub-sampled with a $0.2 \cdot s_n$ pixel grid. Denote the set of patch features extracted from image I_n by $\left\{f_n^m = \left[SIFT_n^m, \left(x_n^m - x_n^f\right)\right]\right\}$, where $SIFT_n^m$ is the descriptor of the m-th feature, x_n^m is its image location, $\left(x_n^m - x_n^f\right)$ is the offset between the feature and the face, and square brackets denote a row vector. The index m enumerates the features in arbitrary order for each image. Denote by k_n the number of patch features extracted from image I_n.

The probabilistic generative model explaining all the gathered data is defined as follows. The observed variables of the model are: the face-hand offset $O_H^F = o_n^{fh} \equiv x_n^h - x_n^f$, the hand-elbow offset $O_E^H = o_n^{he} \equiv x_n^e - x_n^h$, and the patch features $\{F^m = f_n^m\}$. The unobserved variables of the model are the action label variable A, and the set of binary variables $\{B^m\}$, one for each extracted patch feature. The meaning of $B^m = 1$ is that the m-th patch feature was generated by the action A, while the meaning of $B^m = 0$ is that the m-th patch feature was generated independently of A. Throughout the paper we will use a shorthand form of variable assignments, e.g., $P\left(o_n^{fh}, o_n^{he}\right)$ instead of $P\left(O_H^F = o_n^{fh}, O_E^H = o_n^{he}\right)$. We define the joint distribution of the model that generates the data for image I_n as:

$$P\left(A, \{B^m\}, o_n^{fh}, o_n^{he}, \{f_n^m\}\right) = P\left(A\right) \cdot P\left(o_n^{fh}, o_n^{he}\right) \cdot \prod_{m=1}^{k_n} P\left(B^m\right) \cdot P\left(f_n^m \left| A, o_n^{fh}, o_n^{he}, B^m \right.\right) \tag{1}$$

Here $P\left(A\right)$ is a prior action distribution, which we take to be uniform, and:

$$P\left(f_n^m \left| A, o_n^{fh}, o_n^{he}, B^m \right.\right) = \begin{cases} P\left(f_n^m \left| A, o_n^{fh}, o_n^{he} \right.\right) & \text{if } B^m = 1 \\ P\left(f_n^m \left| o_n^{fh}, o_n^{he} \right.\right) & \text{otherwise} \end{cases} \tag{2}$$

The $P\left(B^m\right) = \alpha$, is the prior probability for the m-th feature to be generated from the action, and we assume it maintains the following relation: $P\left(B^m = 1\right) = \alpha \ll (1 - \alpha) = P\left(B^m = 0\right)$ reflecting the fact that most patch features are not related to the action. Figure 1c shows the graphical representation of the proposed model.

As shown in the Appendix A, in order to find the action label assignment to A that maximizes the posterior of the proposed probabilistic generative model, it is sufficient to compute:

$$\arg\max_A \log P\left(A \left| o_n^{fh}, o_n^{he}, \{f_n^m\} \right.\right) = \arg\max_A \sum_{m=1}^{k_n} \frac{P\left(f_n^m, A, o_n^{fh}, o_n^{he}\right)}{P\left(f_n^m, o_n^{fh}, o_n^{he}\right)} \tag{3}$$

As can be seen from eq. 3, and as shown in the Appendix A, the inference is independent of the exact value of α (as long as $\alpha \ll (1 - \alpha)$). In the following section 2.3 we explain how to empirically estimate the probabilities $P\left(f_n^m, A, o_n^{fh}, o_n^{he}\right)$ and $P\left(f_n^m, o_n^{fh}, o_n^{he}\right)$ that are necessary to compute 3.

2.3 Model probabilities

The model probabilities are estimated from the training data using Kernel Density Estimation (KDE) [19]. Assume we are given a set of samples from some

distribution of interest $\{Y_1, \ldots, Y_R\}$. Given a new sample from the same distribution - Y, a symmetric Gaussian KDE estimate $P(Y)$ for the probability of Y can be approximated as:

$$P(Y) \propto \frac{1}{R} \cdot \sum_{r=1}^{R} \exp\left(-0.5 \cdot \|Y - Y_r\|^2 \Big/ \sigma^2\right) \approx \frac{1}{R} \cdot \sum_{Y_r \in NN} \exp\left(-0.5 \cdot \|Y - Y_r\|^2 \Big/ \sigma^2\right) \tag{4}$$

where NN is the set of nearest neighbors of Y within the given set of samples. When the number of samples R is large, brute-force search for the NN set becomes infeasible. Therefore, we use Approximate Nearest Neighbor (ANN) search (using the implementation of [20]) to compute the KDE. To compute $P\left(f_n^m, A = a, o_n^{fh}, o_n^{he}\right)$ for the m-th patch feature $f_n^m = \left[SIFT_n^m, \left(x_n^m - x_n^h\right)\right]$ in test image I_n, we compute the NN set by searching for the nearest neighbors of the row vector $\left[SIFT_n^m, \frac{1}{s_n}\left(x_n^m - x_n^h\right), \frac{1}{s_n} o_n^{fh}, \frac{1}{s_n} o_n^{he}\right]$ in a set of row vectors:

$$\left\{ \left[SIFT_t^r, \frac{1}{s_t}\left(x_t^r - x_t^h\right), \frac{1}{s_t} o_t^{fh}, \frac{1}{s_t} o_t^{he}\right] \middle| \text{ all } f_t^r \text{ in training images } I_t, \text{ s.t. } a_t = a \right\}$$

using an ANN query. Recall that s_n was defined as the width of the detected face in image I_n, and hence $\frac{1}{s_n}$ is the scale factor that we use for the offsets in the query. The query returns a set of K nearest neighbors, and the Gaussian KDE 4, with $\sigma = 0.2$, is applied to this set to compute the estimated probability $P\left(f_n^m, A = a, o_n^{fh}, o_n^{he}\right)$. In our experiments we found that it is sufficient to use $K = 25$. The $P\left(f_n^m, o_n^{fh}, o_n^{he}\right)$ is computed as:

$$P\left(f_n^m, o_n^{fh}, o_n^{he}\right) = \sum_a P\left(f_n^m, A = a, o_n^{fh}, o_n^{he}\right)$$

3 Results

To test our approach, we have applied it to two static transitive action recognition datasets. The first dataset was created by us and contained 12 similar transitive actions performed by 10 different people, appearing against different natural backgrounds. The second dataset was compiled by [7] for dynamic transitive action recognition. It contained 9 different people performing 6 transitive actions. Although originally designed and used by Gupta *et al* in [7] for dynamic action recognition, we transformed it into a static action recognition dataset by assigning action labels to frames actually containing the actions and treating each such frame as a separate static instance of the action. Since successive frames are not independent, the experiments conducted on both datasets were all performed in a person-leave-one-out manner, meaning that during the training we completely excluded all the frames of the tested person. Sections 3.1 and 3.2 describe the respective experiments and results on our and the Gupta *et al* datasets. To make the task more challenging, all experiments were performed on grayscale versions of the images. Figures 3 and 6a show examples of successfully

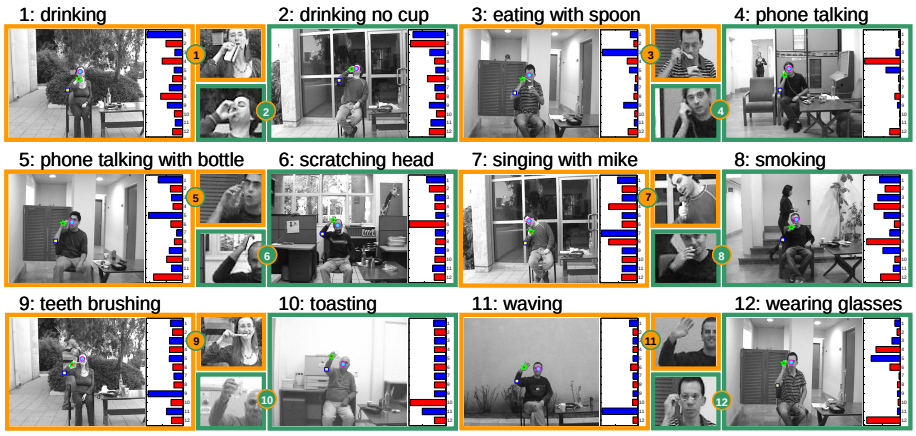


Fig. 3. Examples of similar static transitive action recognition on our 12-actions / 10-people ('12/10') dataset. On all examples, the detected face, hand and elbow are shown by cyan circle, red-green star and yellow square, respectively. At the right hand side of each image, the bar graph shows the estimated log-posterior of the action variable A. Each example shows a zoomed in ROI of the action. Additional examples are provided in supplementary material.

recognized static transitive actions from the two tested datasets respectively, and figure 4b shows some interesting failures.

3.1 Our dataset experiments

Our '12/10' dataset consists of 120 videos of 10 people performing 12 similar transitive actions, namely drinking, talking on the phone, scratching, toasting, waving, brushing teeth, smoking, wearing glasses, eating with a spoon, singing to a microphone, and also drinking without a cup and making a phone call with a bottle. All people except one were filmed against different backgrounds. All backgrounds were natural indoor / outdoor scenes containing both clutter and people passing by. The drinking and toasting actions were performed with 2-4 different tools, and phone talking was performed with mobile and regular phones. Overall, the dataset contains 44,522 frames. Since not all frames contain actions of interest (e.g. in drinking there are frames where the person reaches to / puts down a cup or a bottle), we selected the relevant frames and assigned action labels for them in a semi-automatic manner. Since all the actions of interest are performed in a 'hand-near-face' pose, we assigned the label of the movie sequence to all the frames in which the automatically detected hand was within twice face width from the automatically detected face. This produced 23,277 relevant action frames, and each of them was considered a separate instance of an action inheriting the label of its video sequence. To validate the assigned action labels a significant portion of selected frames was checked. Overall, the obtained ground truth was consistent with the human perception in about 95% of the cases.

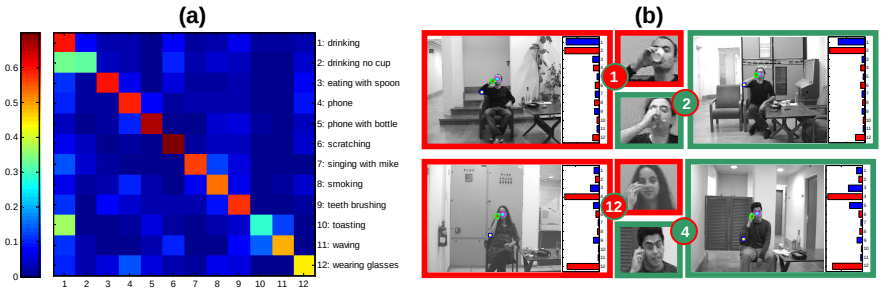


Fig. 4. (a) Average static action confusion matrix obtained by leave-one-out cross validation of the proposed method on the '12/10' dataset; (b) Some interesting failures (red boxes), on the right of each failure there is a successfully recognized instance of an action with which the method has confused.

In our dataset experiments, the part detection models for the face, hand and elbow described in section 2.1, were trained using 10 additional short movies, one for each person, in which the actors were asked to randomly wave their hands. On these 10 movies, face, hand and elbow locations were manually marked. The learned models were then applied to detect their respective parts on the 120 movie sequences of our dataset. In order to estimate the performance of the hand detection, we applied particle-filter tracking to the per-frame probability masks obtained by the hand detection. The results of the tracking were then viewed in video mode and found to be almost always consistent with the correct hand location (examples are provided in the supplementary material). We then compared the maximal scoring location in each probability mask with the location obtained by tracking and obtained 94.1% match.

The action recognition stage of the method, explained in section 2.2, operates using the results of the part detection. Since during training we assume video input, we used the results of the tracked hand location for the learning and used the unfiltered static hand detection results for test. The action recognition was tested in a person-leave-one-out manner. In each of the 10 test iterations (one for each person in the set), one 'test' person was left out, the action models were trained on all the action instances of the 9 remaining 'training' people and tested on the action instances of the test person. Figure 4a shows the resulting mean confusion matrix computed over all 10 test people. The average correct recognition accuracy was $53.2 \pm 9\%$.

Using our dataset, we compared the proposed approach with two baselines. For both baselines the same set of 23,277 relevant labeled static action frames was used. Both baselines were intended to test the contribution of the detection of relevant parts (hand, elbow) to the recognition of similar static transitive actions.

The first baseline was applying our method described in 2.2, but using only the face detection and without hand and elbow detection. Instead of collecting patch features from only the near-hand region, for both training and test, the features were collected from the entire bounding box of the person in each im-

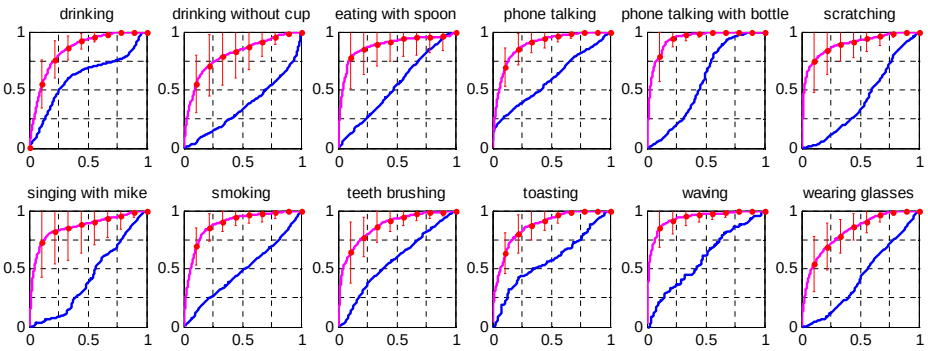


Fig. 5. ROC based comparison with the state-of-the-art method of object detection [21] applied to recognize static actions. For each action, the blue line is the average ROC of [21], and the magenta line is the average ROC of the proposed method.

age. This is a reasonable classifier for using the data without prior body parts detection (obtaining state of the art performance for the object detection task on some standard datasets). The bounding boxes were manually provided (in both training and test). The hand-face and elbow-hand offsets were not given to the baseline. The average performance of this baseline (again measured by a person-leave-one-out test) was $24.6 \pm 12.5\%$, which is significantly lower ($p < 0.0005$) than that for the full method that uses accurate part detections.

The second baseline was applying [21], one of the state-of-the-art object recognition schemes, achieving top scores on recent PASCAL-VOC competitions. The method of [21] was applied to train a separate classifier to each of the 12 actions using manually marked bounding boxes around each training person (which is not required by our method). The test was again performed in a person-leave-one-out manner. Since the resulting classifiers were unbalanced in terms of score, figure 5 provides ROC-based comparison of the results of our full approach with the ones obtained by [21] on a per-action basis and averaged over the different people. The highly significant performance gain obtained by our method relative to this baseline suggests that it is difficult to employ standard state-of-the-art object recognition techniques in order to recognize and distinguish similar static actions without focusing on the relevant parts, such as the hand, the elbow and the face.

3.2 Gupta et al. dataset experiments

This dataset was suggested by [7]. It consists of 46 movie sequences of 9 different people performing 6 distinct transitive actions: drinking, spraying, answering the phone, making a call, pouring from a pitcher and lighting a flashlight. In each movie, we manually assigned action labels to all the frames actually containing the action (essentially excluding frames in which objects are picked up or returned to the table). Since the distinction between 'making a call' and 'answering phone' was in the presence or absence of the 'dialing' action, we re-labeled

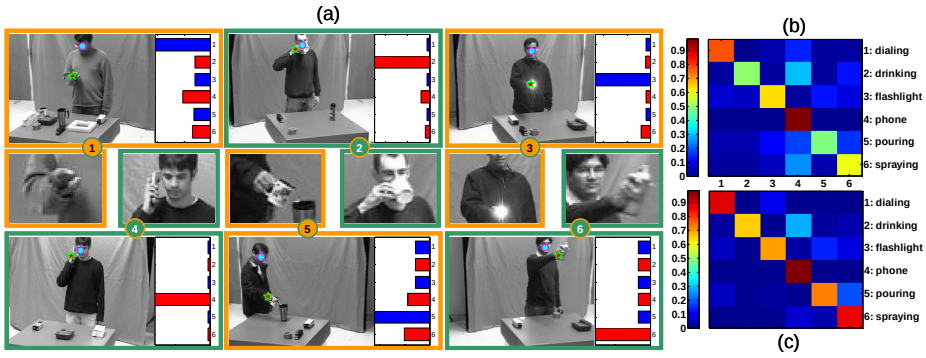


Fig. 6. (a) some successfully identified action examples from the dataset of [7]; (b) mean static action confusion matrix for leave-one-out cross validation experiments on the Gupta et al. dataset; (c) mean static action confusion matrix assuming perfect hand detection (using ground truth hand locations).

the frames of these actions into 'phone talking' and 'dialing'. Finally, we obtained a set with 2,514 frames containing instances of one of the 6 static actions.

The part detection models for the experiments on this dataset were trained on a combined set of 10 part detection training movies from our dataset and 8 movies from 6 people (out of 9) from the Gupta *et al* dataset. The mean hand detection performance on the remaining 38 movies in the dataset was 84.9%. The mean hand detection performance on the movies of the 3 completely unseen test people was 87.3%. Since most people in the dataset wear very dark clothing, in many cases the elbow is invisible and therefore it was not used in this experiment (it is straightforward to remove it from the model by assigning a fixed value to the hand-elbow offset in both training and test). The action recognition performance was measured using the person-leave-one-out cross-validation, in the same manner as for our dataset. During training only, particle filter was used to track the hand over the hand detection probability masks. During testing, both the first and the second maximum of the hand detection probability mask were used as candidate hand locations, and the action scores were maximized among the ones obtained for each candidate location separately. The average accuracy over the 6 static actions was $68 \pm 9.4\%$, and average confusion matrix is shown in figure 6b. The presented results are for the static action recognition, and hence are not directly comparable to the results obtained on this dataset for the dynamic action recognition by [7], who obtained 93.34% recognition (measured over video sequences) using both the temporal information and a priori models for the participating objects (cup, pitcher, flashlight, spray bottle and phone). The action recognition performance depends on the hand detection accuracy. To validate the accuracy of the action recognition alone, we also tested using ground truth hand locations (for both training and test), and obtained $83.5 \pm 12.1\%$ average accuracy. Figure 6c shows the average confusion matrix obtained for the ideal hand detection experiment.

4 Discussion

We have presented a method for recognizing similar transitive actions from single images. The key idea of the approach is to employ features which, are anchored to body parts relevant for the action during both learning and recognition. The use of these features is made possible by the ability of the method to accurately detect these body parts. We have shown that without using these body anchored features there is a highly significant drop in performance even when employing state-of-the-art methods for object recognition. The presented approach is more sequential in nature than most schemes in current use. The face of the person is detected first, and its detection is sequentially extended through feature chains to the detection of hands and elbows (other starting location could be used, but we found the face to be the most reliable). Then, the region around the hand is extracted and analyzed while conditioning on the relative configuration of the detected parts. Following this analysis, relevant objects and hand-object configuration are discovered (through the configuration of their image features) during learning and used to decide on the performed action during testing. This method allows us to approach the learning of the actions in a realistic weakly supervised manner, when only the action labels are given, and learn relevant object and hand-object models on the fly, and without the need for an a priori dictionary of objects we may encounter.

As for future directions, following the accurate detection of relevant body parts, it becomes possible to learn and use additional components associated with specific body parts, which are useful for action recognition but not yet modeled in the current method. These include: 3D face orientation, facial expression, gaze direction, explicit model for the configuration of the fingers, etc. In addition, more detailed models of tools can be obtained by combining hand-detection with local segmentation that separates the hand and the tool, and by extracting images of action-tools in videos before they are grasped or after they are released.

References

1. Kruger, V., Kragic, D., Geib, C.: The meaning of action a review on action recognition and mapping. (2007)
2. Niebles, J.C., Wang, H., Fei-Fei, L.: Unsupervised learning of human action categories using spatial-temporal words. In: BMVC. (2006)
3. Jhuang, H., Serre, T., Wolf, L., Poggio, T.: A biologically inspired system for action recognition. In: ICCV. (2007) 1–8
4. Laptev, I., Perez, P.: Retrieving actions in movies. ICCV (2007) 1–8
5. Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. PAMI **29** (2007) 2247–2253
6. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR. (2008) 1–8
7. Gupta, A., Kembhavi, A., Davis, L.: Observing human-object interactions: Using spatial and functional compatibility for recognition. PAMI (2009)
8. Filipovych, R., Ribeiro, E.: Recognizing primitive interactions by exploring actor-object states. In: CVPR. (2008) 1–7

9. Wang, Y., Jiang, H., Drew, M.S., nian Li, Z., Mori, G.: Unsupervised discovery of action classes. In: CVPR. (2006) 5
10. Li, L., Fei-Fei, L.: What, where and who? classifying events by scene and object recognition. In: ICCV. (2007) 1–8
11. Thurau, C., Hlavac, V.: Pose primitive based human action recognition in videos or still images. In: CVPR. (2008) 1–8
12. Blake, A., Rother, C., Brown, M., Perez, P., Torr, P.: Interactive image segmentation using an adaptive gmmrf model. ECCV (2004)
13. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local svm approach. ICPR (2004)
14. Ferrari, V., Marin, M., Zisserman, A.: Progressive search space reduction for human pose estimation. CVPR (2008)
15. Andriluka, M., Roth, S., Schiele, B.: Pictorial structures revisited: People detection and articulated pose estimation. CVPR (2009)
16. Karlinsky, L., Dinerstein, M., Harari, D., Ullman, S.: The chains model for detecting parts by their context. CVPR (2010)
17. Felzenszwalb, P., Huttenlocher, D.: Pictorial structures for object recognition. IJCV **61** (2005) 55–79
18. Ramanan, D., Forsyth, D.A., Barnard, K.: Building models of animals from video. PAMI (2006)
19. Duda, R., Hart, P.: Pattern classification and scene analysis. Wiley (1973)
20. Mount, D., Arya, S.: Ann: A library for approximate nearest neighbor searching. CGC 2nd Annual Workshop on Comp. Geometry (1997)
21. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. CVPR (2008) 1–8

A Log-posterior derivation

Here we derive the equivalent form of log-posterior (eq. 3) of the proposed probabilistic action recognition model defined in eq. 1. In 5, the symbol $\sim$ means equivalent in terms of maximizing over the values of the action variable A.

$$\begin{array}{l}
\log P\left(A\left|o_n^{fh},o_n^{he},\{f_n^m\}\right.\right)\sim\log\sum_{\{B^m\}}P\left(A,\{B^m\},o_n^{fh},o_n^{he},\{f_n^m\}\right)= \\
\log\left[P\left(A\right)\cdot P\left(o_n^{fh},o_n^{he}\right)\cdot\prod_{m=1}^{k_n}\left[\sum_{B^m=0}^{1}P\left(B^m\right)\cdot P\left(f_n^m\left|A,o_n^{fh},o_n^{he},B^m\right.\right)\right]\right]\sim \\
\sum_{m=1}^{k_n}\log\left[\sum_{B^m=0}^{1}P\left(B^m\right)\cdot P\left(f_n^m\left|A,o_n^{fh},o_n^{he},B^m\right.\right)\right]= \\
\sum_{m=1}^{k_n}\log\left[\alpha\cdot P\left(f_n^m\left|A,o_n^{fh},o_n^{he}\right.\right)+(1-\alpha)\cdot P\left(f_n^m\left|o_n^{fh},o_n^{he}\right.\right)\right]= \\
\sum_{m=1}^{k_n}\log\left[1+\frac{P\left(f_n^m\left|A,o_n^{fh},o_n^{he}\right.\right)}{\gamma\cdot P\left(f_n^m\left|o_n^{fh},o_n^{he}\right.\right)}\right]+\sum_{m=1}^{k_n}\log\left[(1-\alpha)\cdot P\left(f_n^m\left|o_n^{fh},o_n^{he}\right.\right)\right]\sim \\
\sum_{m=1}^{k_n}\log\left[1+\frac{P\left(f_n^m\left|A,o_n^{fh},o_n^{he}\right.\right)}{\gamma\cdot P\left(f_n^m\left|o_n^{fh},o_n^{he}\right.\right)}\right]\underset{(*)}{\approx}\sum_{m=1}^{k_n}\frac{P\left(f_n^m\left|A,o_n^{fh},o_n^{he}\right.\right)}{\gamma\cdot P\left(f_n^m\left|o_n^{fh},o_n^{he}\right.\right)}\sim\sum_{m=1}^{k_n}\frac{P\left(f_n^m\left|A,o_n^{fh},o_n^{he}\right.\right)}{P\left(f_n^m\left|o_n^{fh},o_n^{he}\right.\right)}\sim \\
\sum_{m=1}^{k_n}\frac{P\left(f_n^m,A,o_n^{fh},o_n^{he}\right)}{P\left(f_n^m,o_n^{fh},o_n^{he}\right)}
\end{array}\tag{5}$$

In eq. 5, $\gamma=\alpha/\left(1-\alpha\right)$, the term $\sum_{m=1}^{k_n}\log\left[(1-\alpha)\cdot P\left(f_n^m\left|o_n^{fh},o_n^{he}\right.\right)\right]$ is independent of the action (constant for a given image I_n) and thus can be dropped, and $(*)$ follows from $\log\left(1+\varepsilon\right)\approx\varepsilon$ for $\varepsilon\ll 1$ and from γ being large due to our assumption that $\alpha\ll(1-\alpha)$.

Supplementary Material (paper ID 105)

This manuscript contains the supplementary material for the paper ID 105: 'Identifying similar transitive actions in single images'. Section 1 describes the movies, provided within the current supplementary. The movies show examples of deformable part detection, namely the face, the hand and the elbow, and also examples of static action recognition (performed on single images). The static action recognition method was applied on two datasets: '12/10' dataset and Gupta *et al* dataset [7]. The aforementioned datasets are described in the paper. In addition, figure 1 depicts several examples of interesting action recognition failures.

1 Movies

Three movies are provided together with this supplementary material. The first two movies contain sample results of successful action recognition from static images (frames) from two datasets. The first dataset, titled '12/10', consists of 120 videos of 10 people performing 12 similar transitive actions, namely drinking, talking on the phone, scratching, toasting, waving, brushing teeth, smoking, wearing glasses, eating with a spoon, singing to a microphone, and also drinking without a cup and making a phone call with a bottle. The second dataset, provided by Gupta *et al* [7], consists of 46 videos of 9 people performing 6 transitive actions, namely drinking, spraying, answering the phone, making a call, pouring from a pitcher and lighting a flashlight. The third movie contains examples of detection of the hand, the face, and the elbow on the '12/10' dataset.

All the action recognition experiments conducted are performed in leave-one-out manner, where all the frames that contain a test person are excluded from the training. The details of the experiments and the quantitative evaluation of the results can be found in the paper.

To view the movies it is necessary to have the Xvid codec installed. The codec can be found within K-lite codec pack:

http://www.codecguide.com/download_kl.htm

The movies illustrating the results of static action recognition experiments performed on the two datasets are:

1. **'12_actions_10_people_Xvid.avi'** - shows sample successful static action recognition results on '12/10' dataset.

2. **'6_actions_9_people_Gupta_et_al_Xvid.avi'** - shows sample successful static action recognition results on Gupta *et al* dataset [7].

The movie containing examples of the parts detection is:

1. **'example_part_detections_12_10_Xvid.avi'** - shows example detections of the hand, the face, and the elbow on frames from '12/10' dataset.

In each frame of all the movies the detected location of the face of the person is marked by a magenta circle filled in by light blue. In the current method only the right hand of the person is being detected. A green star with red color inside marks the detected location of the right hand. The detected location of the elbow is marked by a yellow square.

In movies showing action recognition examples, the bar graph on the right shows the log-likelihood estimates obtained by the proposed method for all the considered actions.

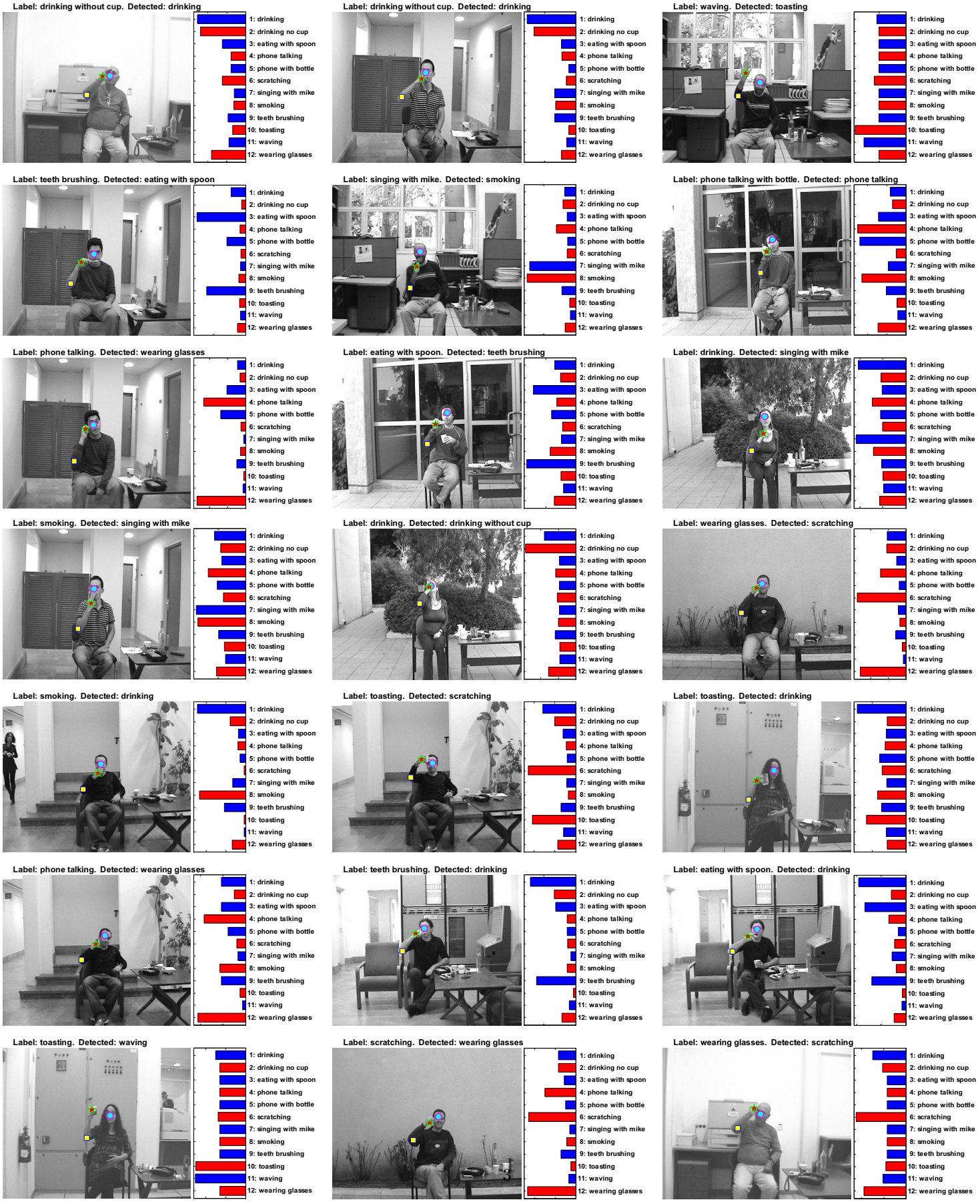


Figure 1: Some examples of interesting static action recognition failures. In order to see the fine details, please zoom in.