ABSTRACT

During the past decade, computer vision research has focused on constructing image based appearance models of objects and action classes using large databases of examples (positive and negative) and machine learning to construct models. Visual inference however involves not only detecting and recognizing objects and actions but also extracting rich relationships between objects and actions to form storylines or plots. These relationships are also improve recognition performance of appearance-based models. Instead of identifying individual objects and actions in isolation, such systems improve recognition rates by augmenting appearance based models with contextual models based on object-object, action-action and object-action relationships. In this thesis, we look at the problem of using contextual information for recognition from three different perspectives: (a) Representation of Contextual Models (b) Role of language in learning semantic/contextual models (c) Learning of contextual models from weakly labeled data.

Our work departs from the traditional view to visual and contextual learning where individual detectors and relationships are learned separately. Our work focuses on simultaneous learning of visual appearance and contextual models from richly annotated, weakly labeled datasets. Specifically, we show how rich annotations can be utilized to constrain the learning of visually grounded models of nouns, prepositions and comparative adjectives from weakly labeled data. I will also show how visually grounded models of prepositions and comparative adjectives can be utilized as contextual models for scene analysis. We also present storyline models

for interpretation of videos. Storyline models go beyond pair-wise contextual models and represent higher order constraints by allowing only a few and finite number of possible action sequences (stories). Visual inference using storyline models involve inferring the “plot” of the video (sequence of actions) and recognizing individual activities in the plot.

Beyond Nouns and Verbs

by

Abhinav Gupta

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2009

Advisory Committee:
Prof. Larry S. Davis, Chair/Advisor
Prof. Jianbo Shi, Co-Advisor
Prof. David Jacobs
Prof. Ramani Duraiswami
Prof. Benjamin Kedem

Dedication

This thesis is dedicated to my wonderful family (my parents, sister and my fiancée) for all the unconditional love, guidance, and support that they have always given me. I love you all!

Acknowledgments

I owe my gratitude to all the people who have made this thesis possible and who have made my graduate experience one that I will cherish forever. It is impossible to remember all, however I have tried and I apologize to those I've inadvertently left out.

First and foremost I'd like to thank my advisor, Professor Larry Davis without whom this thesis would not have been possible. Larry's knowledge of the literature, deep insight into problems, and fresh ideas and surprising intuition has made him a great advisor and fun to interact with. I would like to thank him for giving me all the freedom in the world and using his intuitions throughout to save me from taking wrong paths. I would also like to thank him for his patience in listening to my crazy ideas and then supporting me in all my adventures during the last five years.

I am also deeply grateful to my co-advisor, Prof. Jianbo Shi. His detailed and constructive comments have been very helpful in shaping this thesis as it is today. It was a great experience to spend an exploratory summer at Univ. of Pennsylvania which I will cherish for the rest of my career. I also owe him my sincere thanks for personal guidance and advices which have had remarkable influence on my career.

I am also thankful to Prof. David Jacobs for helping me gain an insight into lot of intereting problems through his courses and discussions about my research problems and directions.

I owe my deepest thanks to my lovely fiancée, Swati Jarial, who has cheered me through the good times and especially the bad. She has been the source of energy during the dull moments and shown remarkable patience with me by staying up all night in my office with me during conference deadlines and weekends. Thank you for bearing with me for the last few years, and helping me in every facet of life (including dataset collection and annotation). I am also thankful to my parents and my sister who have raised me to be the person I am today.

I'd also like to thank my colleagues Shiv Naga Prasad, Vinay Shet, Qihui Zhu, Praveen Srinivasan, Vlad Morariu, Ryan Farrell, Aniruddha Kembhavi, Arpit Jain and Behjat Siddiqui for their help and suggestions on various research problems that I worked on during my Ph.D. I am specifically thankful to my co-authors, Aniruddha Kembhavi and Praveen Srinivasan for the technical discussions which we have had and the nightouts before the deadlines.

I'd like to express my gratitude to my friends Srikanth Vishnubhotla, Anuj Rawat, Ravi Tandon, Pavan Turaga and Himanshu Tyagi for their support, long discussions on my work and helping me finish my thesis smoothly. I am also thankful to Rahul Ratan, Manish Shukla, Archana Anibha, Vishal Khandelwal and Gaurav Agarwal for making my stay at UMD a pleasant and memorable experience.

Credits

The research described in the thesis has been supported by US Government's VACE program. I would like to acknowledge Kobus Barnard and Jia Li for providing the Corel-5K dataset and event recognition dataset. I also thank Mohamed Hussein for object detection code, Antonio Torralba and colleagues for scene-gist code, Navneet Dalal for HOG based human detection code, Shiv Naga Prasad for reachmotion detection code and Vlad Morariu for figtree code. Example in Figure 1.1 has been taken from Blakemore et. al [13].

Table of Contents

Chapter 1

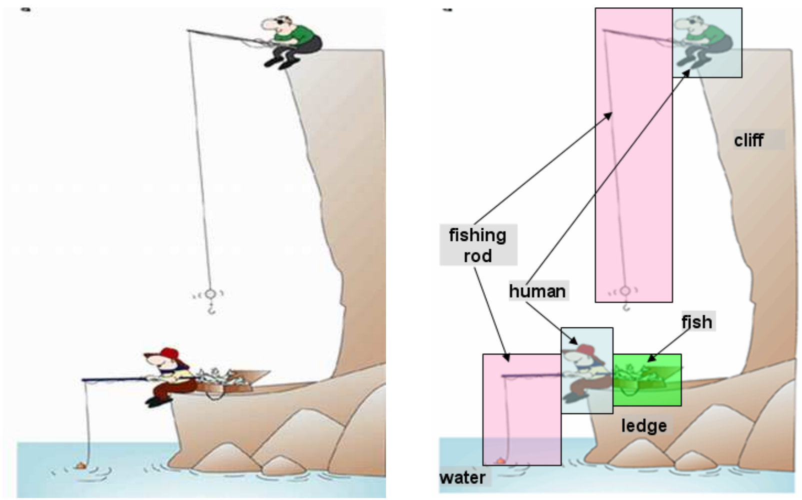
Introduction

The universe is made of stories, not of atoms

Muriel Rukeyser

Traditionally, researchers in computer vision have focused on the problem of modeling and recognizing object and action classes, which are represented in language by nouns and verbs respectively. These problems are central problems in the quest to develop automated systems that "understanding scenes and videos". But only detecting and classifying objects and actions falls short of matching the way humans "see" and perceive the world around them. For example, consider the image shown in figure 1.1(a). An ideal computer vision algorithm would certainly recognize the objects and regions in the image and list the corresponding nouns (See figure 1.1(b)). Humans would additionally recognize and understand object-object, action-action and object-action relationships to produce the explanation - "*A person wearing a green shirt is stealing the fish caught by a person wearing a red shirt*".

Our goal is to create representations of the world depicted in images and videos which are based on physical, functional and causal relationships. These relationships



(a) Original Image (b) Current Approaches

Figure 1.1: Understanding stories hidden in images: Humans tend to extract relationships among objects and actors to infer stories hidden in the images.

can be used to extract rich storylines of images and videos. Similar to humans, we would ideally like to develop a system which understands the intentions of actors to extract the stories in understanding their world. In this thesis, I address the problem of learning and representing "concepts" beyond the modeling of object and action categories which are useful in representing physical, functional and causal relationships in the world. These relationships capture semantic knowledge and serve as contextual information that can be used to improve object and action recognition.

The main contributions of this dissertation [1] are:

- **Storyline Model - Representation and Learning**: We present an ap-

[1] These contribution have been reported in the following publications [41, 42, 45, 43]

proach to modeling and learning visually grounded storyline models of video domains. The storyline of a video describes causal relationships between actions. Beyond recognition of individual actions, discovering causal relationships reveals the semantic meaning of the activities.

- **Language for Learning Contextual Models**: We show how other parts of speech such as prepositions and comparative adjectives can be harnessed to represent the contextual information and how we can use rich annotations to constrain the learning from weakly labeled data.

- **Functional Relationships for Recognition**: We present an approach to represent functional relationships between objects and actions by co-occurrence statistics and we show how using these relationships improves both action recognition and object recognition.

1.1 Understanding Videos, Constructing Plots

People create stories create people; or rather stories create people create stories

Chinua Achebe

Analyzing videos of human activities involves not only recognizing actions (typically based on their appearances), but also determining the story/plot of the video. The storyline of a video includes the actions that occur in that video and the causal relationships between them. Beyond recognition of individual actions, discovering causal relationships helps to better understand the semantic meaning of the activities. However, storylines of videos differ across videos in a domain. There is a substantial difference in terms of the actions that are part of the storyline, agents that perform those actions and the relationships between those actions. Our goal is to learn the space of allowable storylines for a given domain and use a particular "instance" as a contextual model for action recognition. A model that represents the set of storylines that can occur in a video corpus and the general causal relationships amongst actions in the video corpus is referred to as a "storyline model".

A storyline model can be regarded as a (stochastic) grammar, whose language (individual storylines) represents potential plausible "explanations" of new videos in a domain. For example, in analysing a collection of surveillance videos of a traffic intersection scene, a plausible (incomplete) storyline-model is: *When a traffic light turns green traffic starts moving. If, while traffic is moving, a pedestrian walks into an intersection, then the traffic suddenly stops. Otherwise, traffic stops when the signal turns red.*. Not only are the actions "turns green", "moving" and "walks" observable, but there are causal relationships among the actions: traffic starts moving because a light turns green, but it stops because a pedestrian entered an intersection or the signal turned red. Beyond recognition of individual actions, understanding the causal relationships among them provides information about the semantic meaning of the activity in video - the entire set of actions is greater than the sum of the individual actions. The causal relationships are often represented in terms of spatio-temporal relationships between actions. These relationships provide semantic/spatio-temporal context useful for inference of the storyline and recognition of individual actions in subsequent, unannotated videos.

The inference using a storyline model involves simultaneously estimating the storyline of the video and recognizing all the actions that are part of the storyline. We formulate an Integer Programming framework for action recognition and storyline extraction using the storyline model and visual groundings learned from training data.

1.2 Language for Learning Semantic Models

Most of the fundamental ideas of science are essentially simple, and may, as a rule, be expressed in a language comprehensible to everyone.

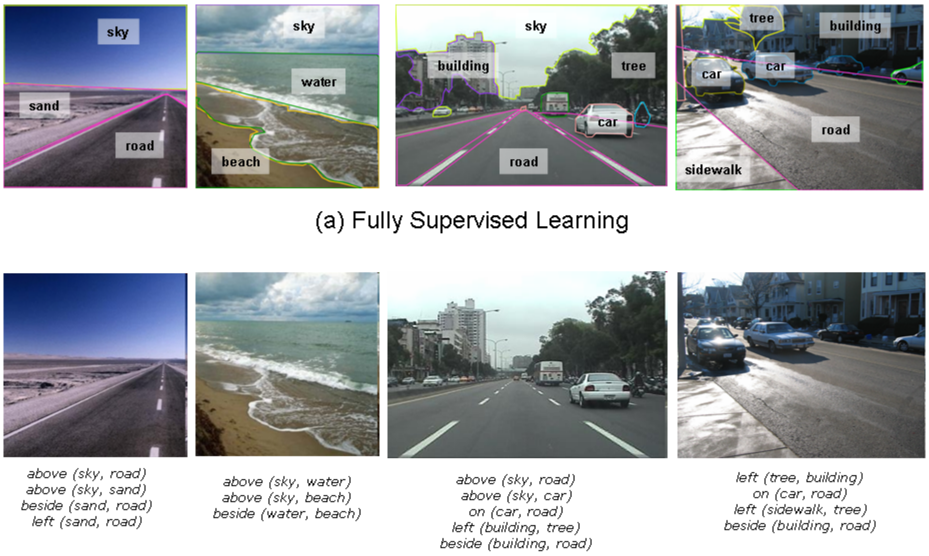
Albert Einstein

Our goal is to learn the semantic model and visual groundings of each action. Traditionally, computer vision approaches have used large visual datasets to learn contextual models which represents relationships between different actions. Such approaches, however, require large labeled datasets which are expensive to obtain. On the other hand, humans are quick to learn relationships with even few and rare instances. One way to learn relationships is direct interaction with the physical world. For example, as a child, when we touch a hot object we realize that it hurts. Current computer vision systems lack the ability to use direct interaction with the world to learn the rules/physical constraints which are the basis of the semantic models. However, most of the rules/constraints can be encoded in language and be used to learn semantic models.

In this thesis, we propose the use of rich weakly annotated data (See Figure 1.2) to simultaneously learn semantic models and visual grounding of these models. Weakly annotated data refers to a large visual dataset where each image/video is described by text. Current vision systems have evaluated the use of "list of nouns" as description to learn object classifiers. We present an approach to harness the richness in language, such as using parts of speech like "prepositions" and "comparative-adjectives" to simultaneously learn object appearances models and contextual models for scene understanding. Another advantage of using the rich linguistic descriptions is that it constrains the learning problem as compared to the original learning problem using only a list of nouns. This leads to better appearance models of nouns as well.

1.3 Functional Recognition - Linking Nouns and Verbs

We also investigated the use of contextual models involving functional relationships. We studied the problem of understanding images and videos of human object interactions. Interpretation of such images/videos involves understanding



(a) Fully Supervised Learning

(b) Weakly Supervised Learning

Figure 1.2: In fully supervised learning the segmentation of the image and region labels are provided. In conventional weakly supervised datasets, only the original image and the list of nouns are provided. No segmentation and correspondence between segments and labels is provided. We propose the use of rich, weakly annotated data in which the original images, list of nouns and some relationships between nouns are provided.

scene/event, analyzing human movements, recognizing manipulable objects and observing the effect of the human movement on those objects. While each of these perceptual tasks can be conducted independently, recognition rates improve when interactions between them are considered. Motivated by psychological studies of human perception, we present a Bayesian approach which integrates various perceptual tasks involved in understanding human object interactions. Our approach goes beyond these traditional approaches and applies spatial and functional constraints on each of the perceptual elements for coherent semantic interpretation. Such constraints allow us to recognize objects and actions when the appearances are not discriminative enough. We also demonstrate the use of such constraints in recognition of actions from static images without using any motion information.

1.4 Organization of Thesis

We first discuss the role of language in learning contextual models. In the following chapter, we will investigate how richness in the language can be harnessed to (a) Improve learning from weakly labeled datasets (b) Learn contextual relationships between objects. More specifically we present an approach to simultaneously learn visual groundings of nouns, prepositions and comparative adjectives from richly annotated, weakly labeled datasets (shown in figure 1.3). We also show how visually grounded models of prepositions and comparative adjectives can be used as contextual models for improving recognition (See figure 1.4).

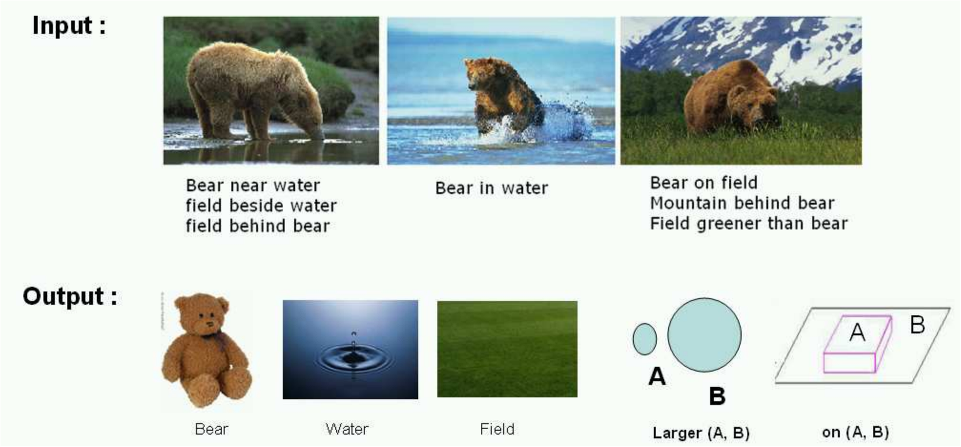


Figure 1.3: In Chapter 2 we focus on how rich linguistic annotations can constrain the learning problem and how we can learn grounded models of nouns, prepositions and comparative adjectives simultaneously.

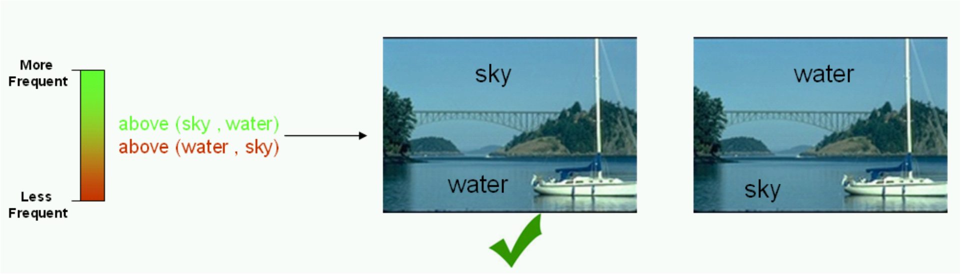
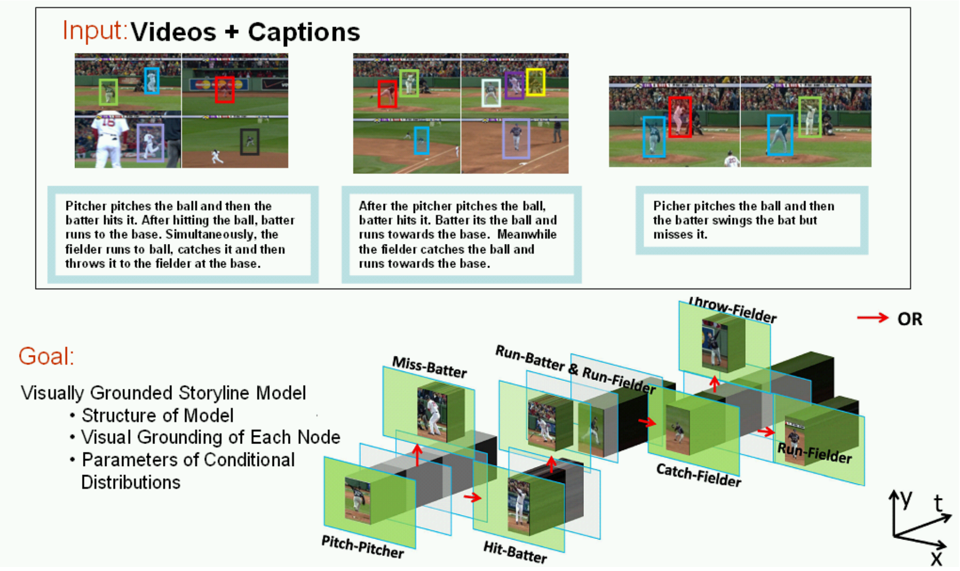


Figure 1.4: Grounded models of prepositions and comparative adjectives provide us contextual model for improving recognition. For example, the two hypothesis of labeling shown in the image are equally likely based on appearances. However, since we know that the relationship **sky above water** occurs more frequently; the left hypothesis becomes more likely.

While Chapter 2 focuses on learning contextual models, based on pair-wise constraints, represented by prepositions and comparative adjectives, such an approach is unable to represent high-order causal relationships between actions in videos. In Chapter 3, we propose the use of storylines which can represent higher order constraints between actions. However, there is substantial variation in storylines across different videos in a domain (in terms of actions and agents performing those actions and relationships between those actions). We present a storyline model which not only encodes the contextual relationships between actions but also learns the space of allowed storylines for videos in a given domain. We present an approach to learn the storyline model from richly annotated videos (See Figure 1.5). Our approach presented in chapter 2 provides the initialization for an iterative structure search of a storyline model and rich linguistic annotations provide additional constraints on the structure learning. We also present an approach to simultaneously estimate storylines and recognize actions given new unseen videos.

Our approaches in chapter 2 and 3 consider contextual models based on noun-noun and verb-verb relationships respectively. While the problem of object recognition and action recognition can be solved separately, with each having its own contextual model, recognition rates improve considerably when functional relationships between objects and actions are also considered. In Chapter 4, we present an approach to evaluate the use of functional constraints for improving action and object recognition. We represent functional constraints by simple co-occurrence relationships and show that such constraints improve both action and object recog-



Input: Videos + Captions

Pitcher pitches the ball and then the batter hits it. After hitting the ball, batter runs to the base. Simultaneously, the fielder runs to ball, catches it and then throws it to the fielder at the base.

After the pitcher pitches the ball, batter hits it. Batter its the ball and runs towards the base. Meanwhile the fielder catches the ball and runs towards the base.

Picher pitches the ball and then the batter swings the bat but misses it.

Goal:

Visually Grounded Storyline Model

- Structure of Model
- Visual Grounding of Each Node
- Parameters of Conditional Distributions

Figure 1.5: In Chapter 3 we present our storyline model for action recognition. The storyline model is learned from richly annotated videos as shown

nition considerably. In this chapter, we also present an approach to recognize actions from static images using functional relationships between objects and human motion/pose.

Chapter 2

Exploiting Richness of Language for Learning Contextual Concepts

Language is not only the vehicle of thought, it is a great and efficient instrument in thinking.

Sir H. Davy

2.1 Weakly Labeled Datasets

One of the long term goals of computer vision has been to represent and recognize objects in natural images. Even young children can name and recognize thousands of objects, and the problem of object recognition is central to computer vision. It is hard to imagine a truly useful robot companion that could not recognize (and interact with) a large repertoire of natural and man made objects.

Early research on computer object recognition emphasized matching of three dimensional object models against images. But progress was limited both because it is hard to acquire very accurate 3D information from images, and, more importantly, because many of the objects in our world have a large diversity of 3D structure due to non-rigidity, articulation, and within class variability. So, during the past decade, research has instead focused on constructing image based appearance models of objects using large image databases and machine learning to construct models.

Obtaining large datasets with full labeling requires huge manual effort (See figure 2.1). For this reason, there has been recent interest in learning visual classifiers of objects from weakly labeled datasets. Weakly labeled datasets are image datasets with associated text and captions, however, there is no segmentation and correspondence given between the text and the regions which generate the text. Learning a visual classifier involves establishing correspondence between image regions and semantic object classes named by the nouns in the text.

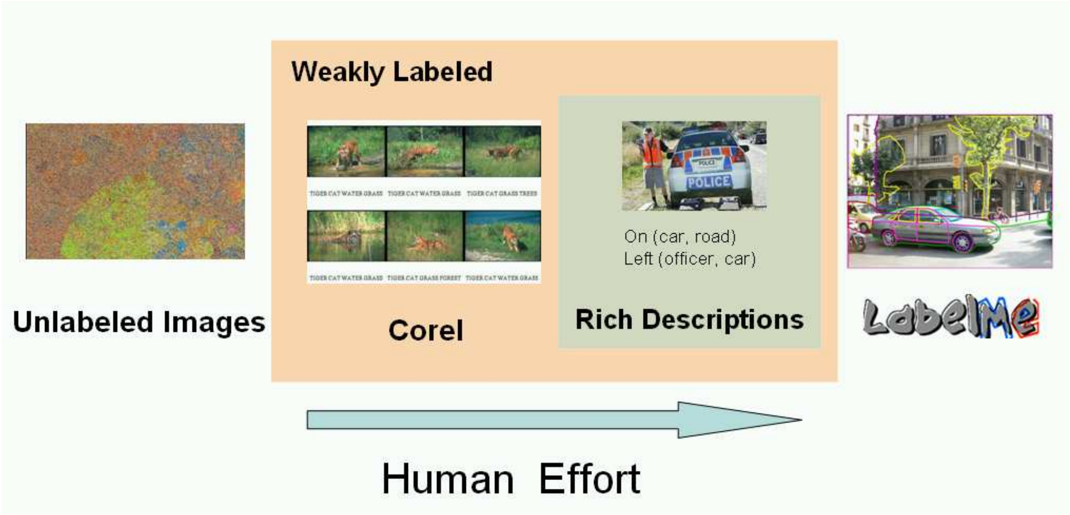


Figure 2.1: Human effort for different type of annotations

Traditionally, computer vision researchers have used a particular kind of weakly labeled datasets - images with lists of nouns (referred to as *tags*) associated with them. These tags are either extracted from the text caption or the dataset is collected in manner where users provide tags. There exist significant ambiguities in correspondence of visual features and object classes. For example, figure 2.2 contains an image which has been annotated with the nouns “car” and “street”. It is difficult to determine which regions of the image correspond to which word unless additional images are available containing “street” but not “car” (and vice-versa). A wide range of automatic image annotation approaches use such co-occurrence relationships to address the correspondence problem.

Some words, however, almost always occur together, which limits the utility of co-occurrence relationships, alone, to reduce ambiguities in correspondence. For example, since cars are typically found on streets, it is difficult to resolve the correspondence using co-occurrence relationships alone. While such confusion is not a serious impediment for image annotation, it is a problem if localization is a goal [1].

We describe how to reduce ambiguities in correspondence by exploiting natural relationships that exists between objects in an image. These relationships correspond to language constructs such as “prepositions” (e.g. above, below) and “comparative adjectives” (e.g. brighter, smaller). If models for such relationships were known and images were annotated with them, then they would constrain the correspondence problem and help resolve ambiguities. For example, in figure 2.2, consider the binary relationship $on(car, street)$. Using this relationship, we can trivially infer that the green region corresponds to “car” and the magenta region

[1] It has also been argued [3] that for accurate retrieval, understanding image semantics (spatial localization) is critical

corresponds to “street”.

The size of the vocabulary of binary relationships is very small compared to the vocabulary of nouns/objects. Therefore, human knowledge could be tapped to specify rules which can act as classifiers for such relationships (for example, a binary relationship $above(s_1, p_1) \Rightarrow s_1.y < p_1.y$). Alternatively, models can be learned from annotated images. Learning such binary relationships from a weakly-labeled dataset would be “straight forward” if we had a solution to the correspondence problem at hand. This leads to a chicken-egg problem, where models for the binary relationships are needed for solving the correspondence problem, and the solution of the correspondence problem is required for acquiring models of the binary relationships. We utilize an EM-based approach to simultaneously learn visual classifiers of objects and “differential” models of common prepositions and comparative binary relationships.

Grounded models of prepositions and comparative adjectives can also be used to represent semantic relationships between objects. These grounded models can therefore be used as a contextual model for scene analysis and improve recognition of objects. Figure 2.3 shows an example. In the image shown in the figure, based on appearances it is hard to classify whether the the region associated with the sun or its reflection is the sun. However, if we can learn a grounding for *above(B,A)* - the y-coordinate of B is more than y-coordinate of A. Then, based on our linguistic knowledge of the world that - *Sun occurs above Sea* and the position of the sea, we can classify the region associated with the sun.

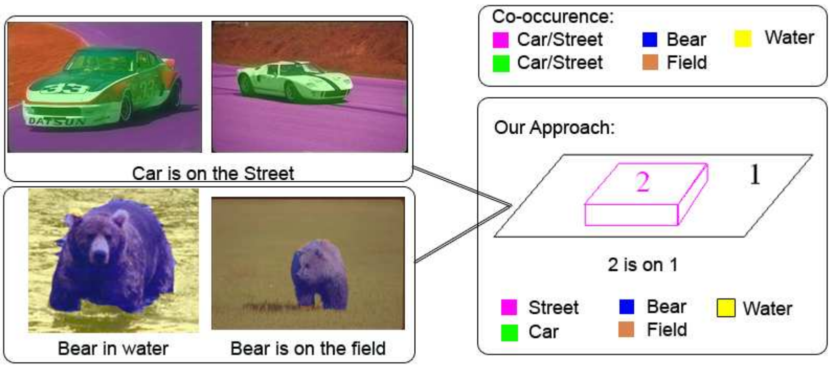


Figure 2.2: An example of how our approach can resolve ambiguities. In the case of co-occurrence based approaches, it is hard to correspond the magenta/green regions to 'car'/'street'. 'Bear', 'water' and 'field' are easy to correspond. However, the correct correspondences of 'bear' and 'field' can be used to acquire a model for the relation 'on'. We can then use that model to classify the green region as belonging to 'car' and the magenta one to 'street', since only this assignment satisfies the binary relationship.

The significance of the work described in this chapter is threefold: (1) It allows us to learn classifiers (i.e models) for a vocabulary of prepositions and comparative adjectives. These classifiers are based on differential features extracted from pairs of regions in an image. (2) Simultaneous learning of nouns and relationships reduces correspondence ambiguity and leads to better learning performance. (3) Learning priors on relationships that exist between nouns constrains the annotation problem and leads to better labeling and localization performance on the test dataset.

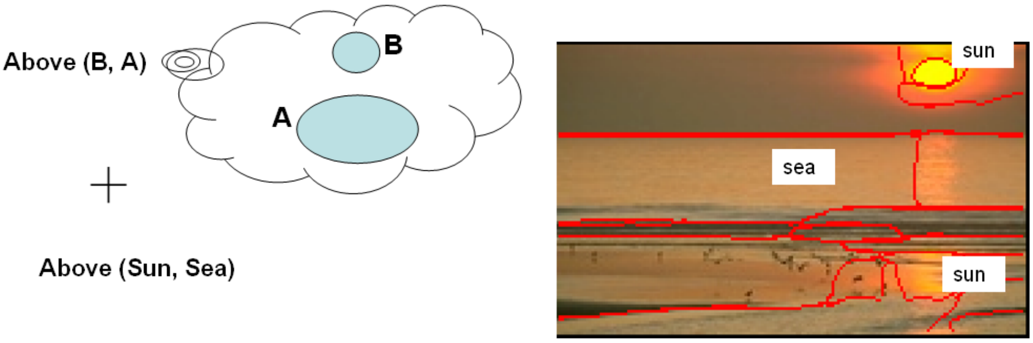


Figure 2.3: Language based Contextual Model

2.2 Related Work

Our work is clearly related to prior work on relating text captions and image features for automatic image annotation [5, 22, 21]. Many learning approaches have been used for annotating images which include translation models [28], statistical models [5, 8], classification approaches [2, 59, 63] and relevance language models [58, 50, 31].

Classification based approaches build classifiers without solving the correspondence problem. These classifiers are learned on positive and negative examples generated from captions. Relevance language models annotate a test image by finding similar images in the training dataset and using the annotation words shared by them.

Statistical approaches model the joint distribution of nouns and image features. These approaches use co-occurrence counts between nouns and image features to predict the annotation of a test image [69, 19]. Barnard et. al [8] presented a generative model for image annotation that induces hierarchical structure from the co-occurrence data. Srikanth et. al [88] proposed an approach to use the hierarchy induced by WordNet for image annotation. Duygulu et. al [28] modeled the problem as a standard machine translation problem. The image is assumed to be a collection of blobs (vocabulary of image features) and the problem becomes analogous to learning a lexicon from aligned bi-text. Other approaches such as [51] also model word to word correlations where prediction of one word induces a prior on prediction of other words.

All these approaches use co-occurrence relationships between nouns and image features; but they cannot, generally, resolve all correspondence ambiguities. They do not utilize other constructs from natural language and speech tagging approaches [17, 18]. As a trivial example, given the annotation “pink flower” and a model of the adjective “pink”, one would expect a dramatic reduction in the set of regions that would be classified as a flower in such an image. Other language constructs, such as “prepositions” or “comparative adjectives”, which express relationships between two or more objects in the image, can also resolve ambiguities.

Our goal is to learn models, in the form of classifiers, for such language constructs. Ferrari et. al [33] presented an approach to learn visual attributes from a training dataset of positive and negative images using a generative model. However, collecting a dataset for all such visual attributes is cumbersome. Ideally we would like to use the original training dataset with captions to learn the appearance of nouns/adjectives and also understand the meanings of common prepositions and comparative adjectives. Barnard et. al [10] presented an approach for learning adjectives and nouns from the same dataset. They treat adjectives similarly to nouns and use a two step process to learn the models. In the first step, they consider only adjectives as annotated text and learn models for them using a latent model. In the second step, they use the same latent model to learn nouns where learned models of adjectives are used to provide prior probabilities for labeling nouns. While such an approach might be applicable to learning models for adjectives, it cannot be applied to learning models for higher order(binary) relationships unless the models for the nouns are given.

Barnard et. al [6] also presented an approach to reduce correspondence ambiguity in weakly labeled data. They separate the problems of learning models of nouns from resolving correspondence ambiguities. They use a loose model for defining affinities between different regions and use the principal of exclusion reasoning to resolve ambiguities. On the other hand, we propose an approach to simultaneously resolve correspondence ambiguities and learn models of nouns using other language constructs which represent higher order relationships [2].

We also present a systematic approach to employing contextual information (second-order) for labeling images. The use of second order contextual information is very important during labeling because it can help resolve the ambiguities due

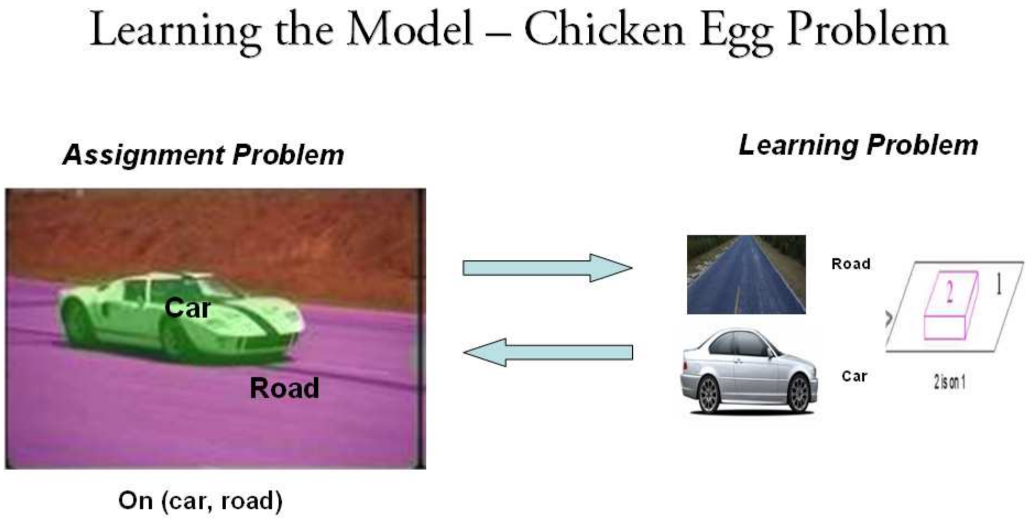
[2] The principles of exclusion reasoning are also applicable to our problem. We, however, ignore them here

to appearance confusion in many cases. For example, a blue homogeneous region, B, can be labeled as "water" as well as "sky" due to the similarity in appearance. However, the relation of the region to other nouns such as the "sun" can resolve the ambiguity. If the relation $below(B, sun)$ is more likely than $in(sun, B)$, then the region B can be labeled as "water" (and vice-versa). As compared to [6], which uses adjacency relations for resolution, our approach provides a broader range of relations(prepositions and comparative adjectives) that can be learned simultaneously with the nouns.

2.3 Overview

Each image in a training set is annotated with nouns and relationships between some subset of pairs of those nouns. We refer to each relationship instance, such as $above(A, B)$, as a predicate. Our goal is to learn classifiers for nouns and relationships (prepositions and comparative adjectives). Similar to [28], we represent each image with a set of image regions. Each image region is represented by a set of visual features based on appearance and shape (e.g area, RGB). The classifiers for nouns are based on these features. The classifiers for relationships are based on differential features extracted from pairs of regions such as the difference in area of two regions.

Learning models of both nouns and relationships requires assigning image regions to annotated nouns. As the data is weakly labeled, there is no explicit assignment of words to image regions. One could, however, assign regions to nouns if the models of nouns and relationships were known. This leads to a chicken-egg problem (See Figure 2.4). We treat assignment as the missing data and use an EM-approach to learn assignment and models simultaneously. In the E-step we evaluate possible assignments using the parameters obtained at previous iterations. Using the probabilistic distribution of assignment computed in the E-step, we estimate the maximum likelihood parameters of the classifiers in the M-step.



Learning the Model – Chicken Egg Problem

Assignment Problem

Learning Problem

Figure 2.4: Correspondence/assignment of regions to words is dependent on learned appearance models; however learned appearance models themselves depend on the correspondence

In the next section, we first discuss our model of generating predicates for a pair of image regions. This is followed by a discussion on learning the parameters of the model, which are the parameters of classifiers for nouns, prepositions and comparative adjectives.

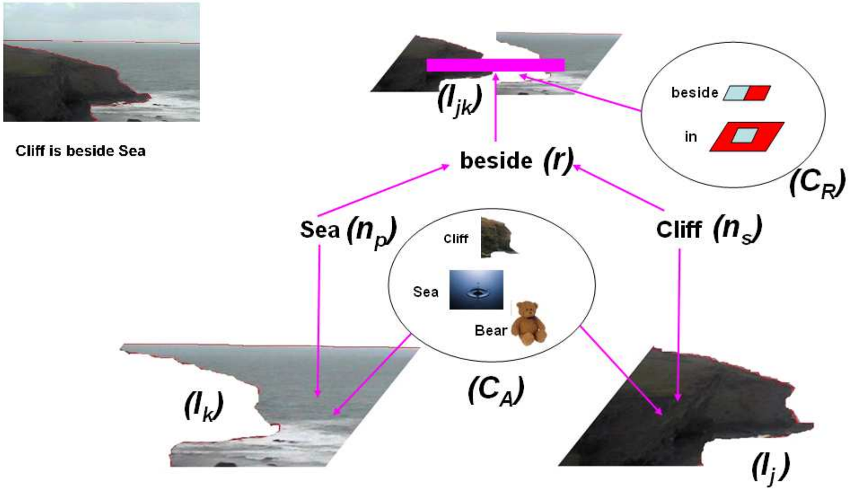


Figure 2.5: The Graphical Model for Image Annotation

2.4 Our Approach

2.4.1 Generative Model

We next describe the model for language and image generation for a pair of objects. Figure 2.5 shows our generative model.

Each image is represented with a set of image regions and each region is associated with an object which can be classified as belonging to a certain semantic object class. These semantic object classes are represented by nouns in the vocabulary[3].

[3] Generally, there will not be a one-one relationship between semantic object classes and nouns. For example, the word "bar" refers to two different semantic concepts in the sentences: "He went to the bar for a drink" and "There were bars in the window to prevent escape". Similarly, one semantic object class can be described by two or more words(synonyms). While dealing with synonyms and word sense disambiguation [9] is an important problem, we simplify the exposition by assuming a

Assume two regions j and k are associated with objects belonging to semantic object classes, n_s and n_p respectively. Each region is described by a set of visual features I_j and I_k. The likelihood of image features I_j and I_k would depend on the nouns n_s and n_p and the parameters of the appearance models(C_A) of these nouns. These parameters encode visual appearance of the object classes.

For every pair of image regions, there exist some relationships between them based on their locations and appearances. Relationship types are represented by a vocabulary of prepositions and comparative adjectives. Let r be a type of relationship (such as “above”, “below”) that holds between the objects associated with regions j and k. The nouns associated with the regions, n_s and n_p, provide priors on the types of relationships in which they might participate (For example, there is a high prior for the relationship “above” if the nouns are “sky” and “water”, since in most images “sky” will occur above “water”). Every relationship is described by differential image features I_{jk}. The likelihood of the differential features depends on the type of relationship r and the parameters of the relationship model C_R.

2.4.2 Learning the Model

The training data consists of images annotated with nouns $(n_1^l, n_2^l..)$ and a set of relationships between these nouns represented by predicates $\mathcal{P}^l$, where l is the image number. Learning the model involves maximizing the likelihood of training

one-one relationship between semantic object classes and the nouns in the annotation.

images being associated with predicates given in the training data. The maximum likelihood parameters are the parameters of object and relationship classifiers, which are represented by $\theta = (C_A, C_R)$. However, evaluating the likelihood is expensive since it requires summation over all possible assignments of image regions to nouns. We instead treat the assignment as missing data and use an EM formulation to estimate θ^{ML}.

$$\begin{aligned} \theta^{ML} &= \arg\max_{\theta} P(\mathcal{P}^1, \mathcal{P}^2..|I^1, I^2.., \theta) = \arg\max_{\theta} \sum_{A} P(\mathcal{P}^1, \mathcal{P}^2.., A|I^1, I^2.., \theta) \\ &= \arg\max_{\theta} \prod_{l=1}^{N} \sum_{A^l} P(\mathcal{P}^l|I^l, \theta, A^l) P(A^l|I^l, \theta) \end{aligned} \qquad (2.1)$$

where A^l defines the assignment of image regions to annotated nouns in image l. Therefore, $A^l_i = j$ indicates that noun n^l_i is associated to region j in image l.

The first term in equation 2.1 represents the joint predicate likelihood given the assignments, classifier parameters and image regions. A predicate is represented as $r^l_i(n^l_{s_i}, n^l_{p_i})$, where r^l_i is a relationship that exists between the nouns associated with region $A^l_{s_i}$ and $A^l_{p_i}$. We assume that each predicate is generated independently of others, given an image and assignment. Therefore, we rewrite the likelihood as:

$$\begin{aligned} P(\mathcal{P}^l|I^l, \theta, A^l) &= \prod_{i=1}^{|\mathcal{P}^l|} P(\mathcal{P}^l_i|I^l, A^l, \theta) \\ &\propto \prod_{i=1}^{|\mathcal{P}^l|} P(r^l_i|I^l_{A^l_{s_i} A^l_{p_i}}, C_R) P(r^l_i|n_{s_i}, n_{p_i}) \end{aligned}$$

Table 2.1: Notation

N: Number of images

l: Image under consideration (superscript)

$\mathcal{P}^l$: Set of Predicates for image l

$(n^l_1, n^l_2...)$: Set of Nouns for image l

$\mathcal{P}^l_i = r^l_i(n^l_{s_i}, n^l_{p_i})$: i^{th} predicate

$A^l_i = j$: Noun n^l_i is associated with region j

C_A: Parameters of models of nouns

C_R:Parameters of models of relationships

r^l_i: Relationship represented by i^{th} predicate

s_i: Index of noun which appears as argument1 in i^{th} predicate

p_i: Index of noun which appears as argument2 in i^{th} predicate

$I^l_{A^l_i}$: Image features for region assigned to noun n^l_i

$$\propto \prod_{i=1}^{|\mathcal{P}^l|} P(I^l_{A^l_{s_i} A^l_{p_i}} | r^l_i, C_R) P(r^l_i | C_R) P(r^l_i | n_{s_i}, n_{p_i})$$

Given the assignments, the probability of associating a predicate $\mathcal{P}^l_i$ to the image is the probability of associating the relationship r^l_i to the differential features associated with the pair of regions assigned to n_{s_i} and n_{p_i}. Using Bayes rule, we transform this into the differential feature likelihood given the relationship word and the parameters of the classifier for that relationship word. $P(r^l_i|C_R)$ represents the prior on relationship words and is assumed uniform.

The second term in equation 2.1 evaluates the probability of an assignment of image regions to nouns given the image and the classifier parameters. Using Bayes rule, we rewrite this as:

$$\begin{aligned} P(A^l|I^l, \theta) &= \prod_{i=1}^{|A^l|} P(n^l_i | I^l_{A^l_i}, C_A) \\ &\propto \prod_{i=1}^{|A^l|} P(I^l_{A^l_i} | n^l_i, C_A) P(n^l_i | C_A) \end{aligned}$$

where $|A^l|$ is the number of annotated nouns in the image, $P(I^l_{A^l_i}|n^l_i, C_A)$ is the image likelihood of the region assigned to the noun, given the noun and the parameters of the object model, $P(n^l_i|C_A)$ is the prior over nouns given the parameters of object models.

2.4.2.1 EM-approach

We use an EM approach to simultaneously solve for the correspondence and for learning the parameters of classifiers represented by θ.

1. **E-step:** Compute the noun assignment for a given set of parameters from the previous iteration represented by θ^{old}. The probability of assignment in which noun i correspond to region j is given by:

$$P(A_i^l = j|\mathcal{P}^l, I^l, \theta^{old}) = \frac{\sum_{A' \in \mathcal{A}_{ij}^l} P(A'|\mathcal{P}^l, I^l, \theta^{old})}{\sum_k \sum_{A' \in \mathcal{A}_{ik}^l} P(A'|\mathcal{P}^l, I^l, \theta^{old})} \tag{2.2}$$

where $\mathcal{A}_{ij}$ refers to the subset of the set of all possible assignments for an image in which noun i is assigned to region j. The probability of any assignment A' for the image can be computed using Bayes rule:

$$P(A'|\mathcal{P}^l, I^l, \theta^{old}) \propto P(\mathcal{P}^l|A', I^l, \theta^{old})P(A'|I^l, \theta^{old}) \tag{2.3}$$

2. **M-step:** For the noun assignment computed in the E-step, we find the new ML parameters by learning both relationship and object classifiers. The ML parameters depend on the type of classifier used. For example, for a gaussian classifier we estimate the mean and variance for each object class and relationship class.

For initialization of the EM approach, we can use any image annotation approach with localization such as the translation based model described in [28].

Based on initial assignments, we initialize the parameters of both relationship and object classifiers.

We also want to learn the priors on relationship types given the nouns represented by $P(r|n_s, n_p)$. After learning the maximum likelihood parameters, we use the relationship classifier and the assignment to find possible relationships between all pairs of words. Using these generated relationship annotations we form a co-occurrence table which is used to compute $P(r|n_s, n_p)$.

2.4.3 Inference

Similar to training, we first divide the test image into regions. Each region j is associated with some features I_j and noun n_j. In this case, I_j acts as an observed variable and we have to estimate n_j. Previous approaches estimate nouns for regions independently of each other. We want to use priors on relationships between pair of nouns to constrain the labeling problem. Therefore, the assignment of labels cannot be done independently of each other. Searching the space of all possible assignments is infeasible.

We use a Bayesian network to represent our labeling problem and use belief propagation for inference. For each region, we have two nodes corresponding to the noun and image features from that region. For all possible pairs of regions, we have another two nodes representing a relationship word and differential features from that pair of regions. Figure 2.6 shows an example of an image with three regions

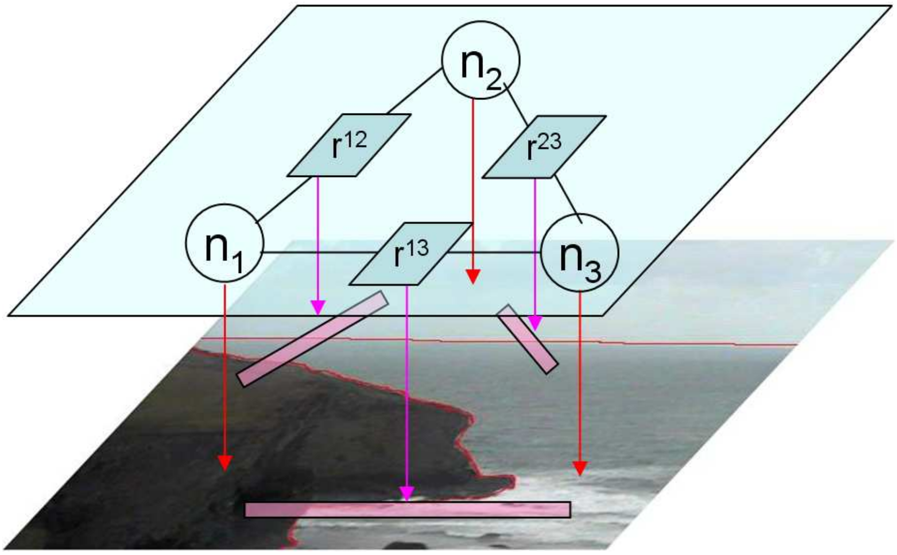


Figure 2.6: An example of a Bayesian network with 3 regions. The r^{jk} represent the possible words for the relationship between regions (j, k). Due to the non-symmetric nature of relationships we consider both (j, k) and (k, j) pairs (in the figure only one is shown). The magenta blocks in the image represent differential features (I_{jk}).

and its associated Bayesian network. The word likelihood is given by:

$$P(n_1, n_2..|I_1, I_2..I_{12}, .., C_A, C_R) \propto \prod_i P(I_i|n_i, C_A) \prod_{(j,k)} \sum_{r_{jk}} P(I_{jk}|r_{jk}, C_R) P(r_{jk}|n_j, n_k) \tag{2.4}$$

2.5 Experimental Results - Corel5K Dataset

In all the experiments, we use a nearest neighbor based likelihood model for nouns and decision stump based likelihood model for relationships. We assume each relationship model is based on one differential feature(for example, the relationship "above" is based on difference in y locations of 2 regions). The parameter learning M-step therefore also involves feature selection for relationship classifiers. For evaluation we use a subset of the Corel5k training and test dataset used in [28]. For training we use 850 images with nouns and hand-labeled the relationships between subsets of pairs of those nouns. We use a vocabulary of 173 nouns and 19 relationships [4].

2.5.1 Learning of Relationship Words

Sixteen relationship words were learned correctly - assigned to correct differential feature. For example, words like *above* and *left* were assigned to differences in y coordinates and x coordinates respectively. One of the interesting case was the word *behind*. While our differential feature set did not have any features which correspond to depth values, the word was associated with difference in texturedness [5]. Three relationship words were associated with the wrong features; words like *in* and *on* are hard to be captured by color, shape and location features. In case of word

[4] above, behind, below, beside, more textured, brighter, in, greener, larger, left, near, far from, ontopof, more blue, right, similar, smaller, taller, shorter.

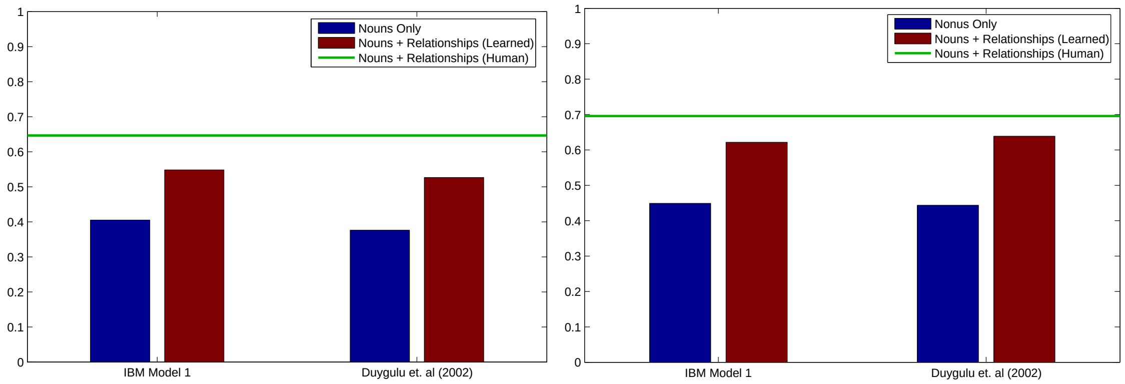
[5] closer regions having higher texturedness than farther regions

taller, most of the tall objects are thin and our segmentation algorithm tends to fragment them.

2.5.2 Resolution of Correspondence Ambiguities

We first evaluate the performance of our approach for the resolution of correspondence ambiguities in the training dataset. To evaluate the localization performance, we randomly sampled 150 images from the training dataset and compare it to human labeling. Similar to [7], we evaluate the performance in terms of two measures: “range of semantics identified” and “frequency correct”. The first measure counts the number of words that are labeled properly by the algorithm. In this case, each word has similar importance regardless of the frequency with which it occurs. In the second case, a word which occurs more frequently is given higher importance. For example, suppose there are two algorithms one of which only labels ’car’ properly and other which only labels ’sky’ properly. Using the first measure, both algorithms have similar performance because they can correctly label one word each. However, using the second measure the latter algorithm is better as sky is more common and hence the number of correctly identified regions would be higher for the latter algorithm.

We compare our approach to image annotation algorithms which can be used for localization of nouns as well. These approaches are used to bootstrap our EM-algorithm. For our experiments, a co-occurrence based translation model [19] and



(a) Semantic Range

(b) Frequency Correct

Figure 2.7: Comparison of normalized “semantic range” and “frequency correct” scores for the training dataset. The performance increases substantially by using prepositions and comparative adjectives in addition to nouns. The green line shows the performance when relationships are not learned but are defined by a human. The two red blocks show the performance of our approach where relationships and nouns are learned using the EM algorithm and bootstrapped by IBM Model1 or Duygulu et. al respectively.

translation based model with mixing probabilities [28] form the baseline algorithms. To show the importance of using “prepositions” and “comparative adjectives” for resolution of correspondence ambiguities, we use both algorithms to bootstrap EM and present our results. We also compare our performance with the algorithm where relationships are defined by a human instead of learning them from the dataset itself. Figure 2.7 compares the performance of all the algorithms with respect to the two measures described above. Figure 2.8 shows some examples of how ambiguity is removed using prepositions and comparative adjectives.

2.5.3 Labeling New Images

We also tested our model on labeling new test images. We used a subset of 500 test images provided in the Corel5k dataset. The subset was chosen based on the vocabulary of nouns learned from the training. The images were selected randomly from those images which had been annotated with the words present in our learned vocabulary. To find the missed labels we compute $\mathcal{S}_t \backslash \mathcal{S}_g$, where $\mathcal{S}_t$ is the set of annotations provided from the Corel dataset and $\mathcal{S}_g$ is the set of annotations generated by the algorithm. However, to test the correctness of labels generated by the algorithm we ask human observers to verify the annotations. We do not use the annotations in the Corel dataset since they contain only a subset of all possible nouns that describe an image. Using Corel annotations for evaluation can be misleading, for example, if there is "sky" in an image and an algorithm generates an annotation "sky" it may be labeled as incorrect because of the absence of sky from the Corel annotations. Figure 2.9 shows the performance of the algorithm on the test dataset. Using the proposed Bayesian model, the number of missed labels decreases by 24% for IBM Model 1 and by 17% for Duygulu et. al [28]. Also, using our approach 63% and 59% of false labels are removed respectively.

Figure 2.11 shows some examples of the labeling on the test set. The examples show how Bayesian reasoning leads to better labeling by applying priors on relationships between nouns. The recall and precision ratios for some common words in the vocabulary are shown in Figure 2.10. The recall ratio of a word represents the ratio of the number of images correctly annotated with that word using the algorithm to the number of images that should have been annotated with that word. The precision ratio of a word is the ratio of number of images that have been correctly annotated with that word to the number of images which were annotated with the word by the algorithm. While recall rates are reported with respect to corel annotations, precision rates are reported with respect to correctness defined by human observers. The results show that using a constrained bayesian model leads to improvement in labeling performance of common words in terms of both recall and precision rates.

2.6 Experimental Results - Perfect Segmentations

The results in the previous section show that our approach outperforms co-occurrence based approach. However, the results in the previous experiments are confounded by bad segmentations. We performed another small experiment to study the effect of using prepositions and comparative adjectives when segmentations are perfect. In this experiment, we used 75 training images from Berkeley and MSRC datasets and we used a vocabulary of 8 nouns and 19 relationships. Figure 2.12 shows the comparative evaluation of our approach on perfect segmentation dataset. Even in case of perfect segmentations, our approach outperforms the approach of Duygulu et. al and the improvement margins are similar as in case of Corel-5k dataset.

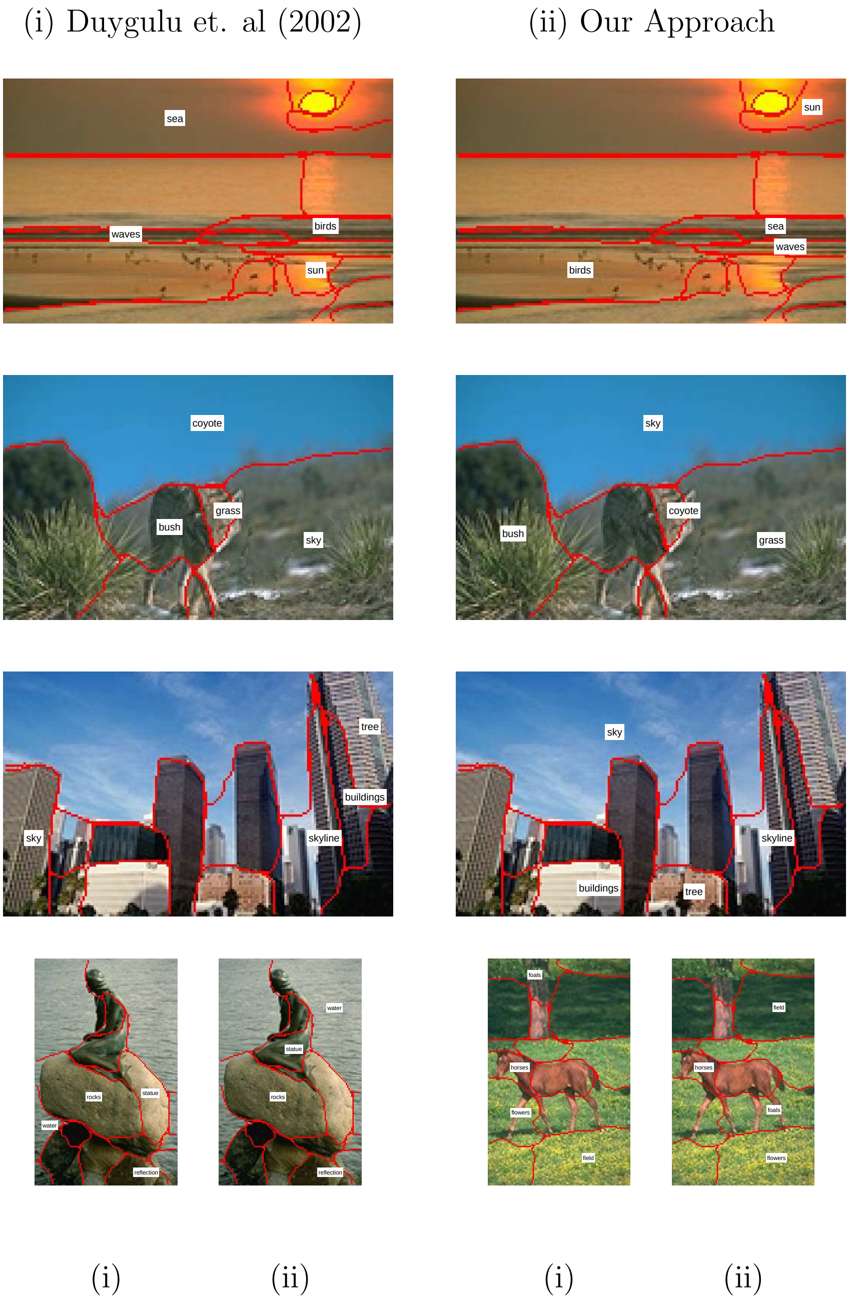
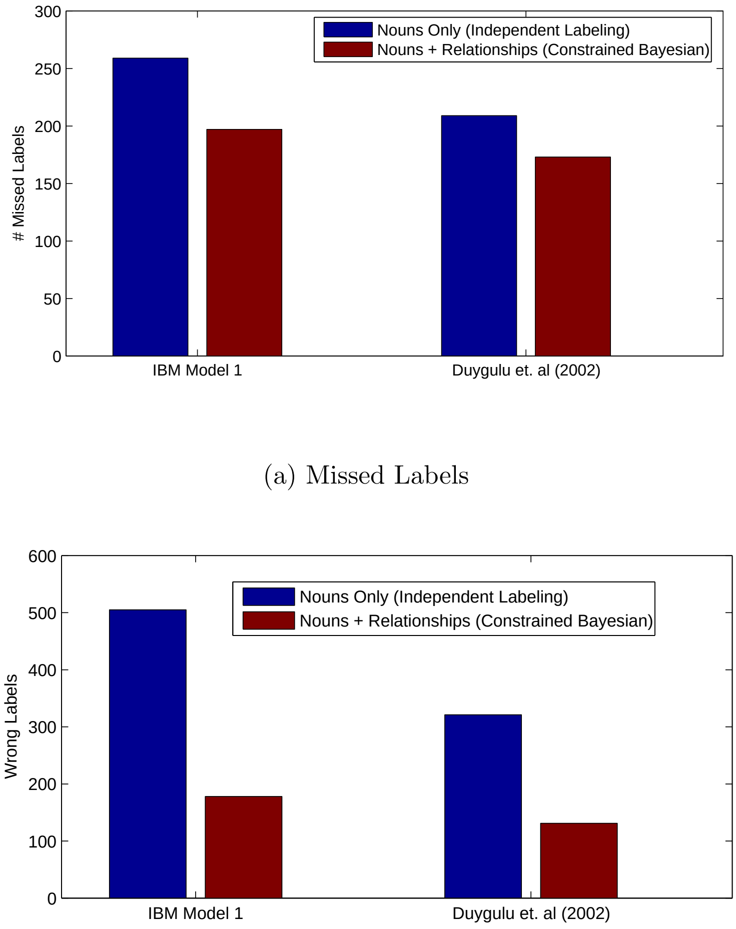


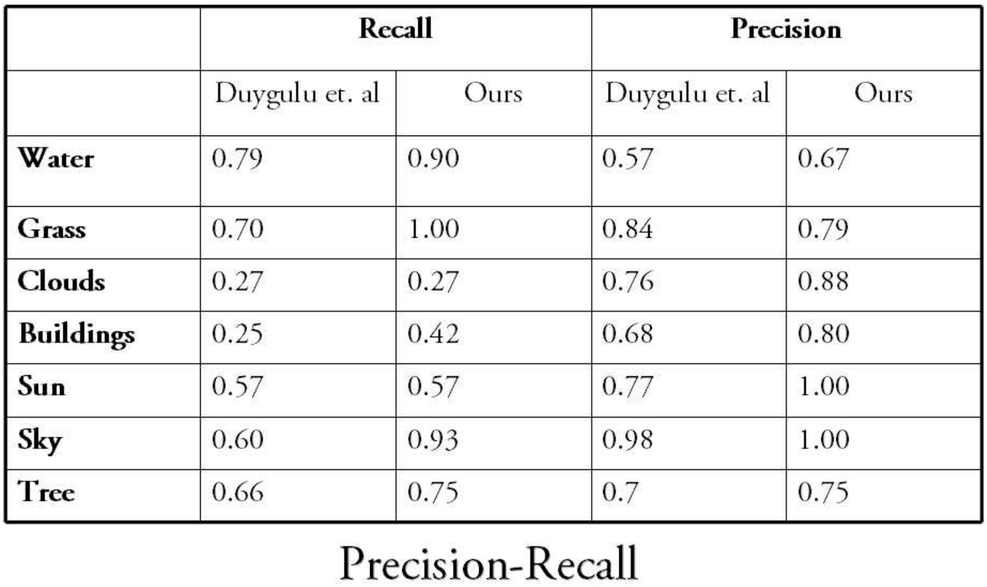
Figure 2.8: Some examples of how correspondence ambiguity can be reduced using prepositions and comparative adjectives. Some of the annotations for the images are: **(a)** *near(birds,sea); below(birds,sun); above(sun, sea); larger(sea,sun); brighter(sun, sea); below(waves,sun)* **(b)** *below(coyote, sky); below(bush, sky); left(bush, coyote); greener(grass, coyote); below(grass,sky)* **(c)** *below(building, sky); below(tree,building); below(tree, skyline); behind(buildings,tree) blueish(sky, tree)* **(d)** *above(statue,rocks); ontopof(rocks, water); larger(water,statue)* **(e)** *below(flowers,horses); ontopof(horses, field); below(flowers,foals)*



(a) Missed Labels

(b) False Labels

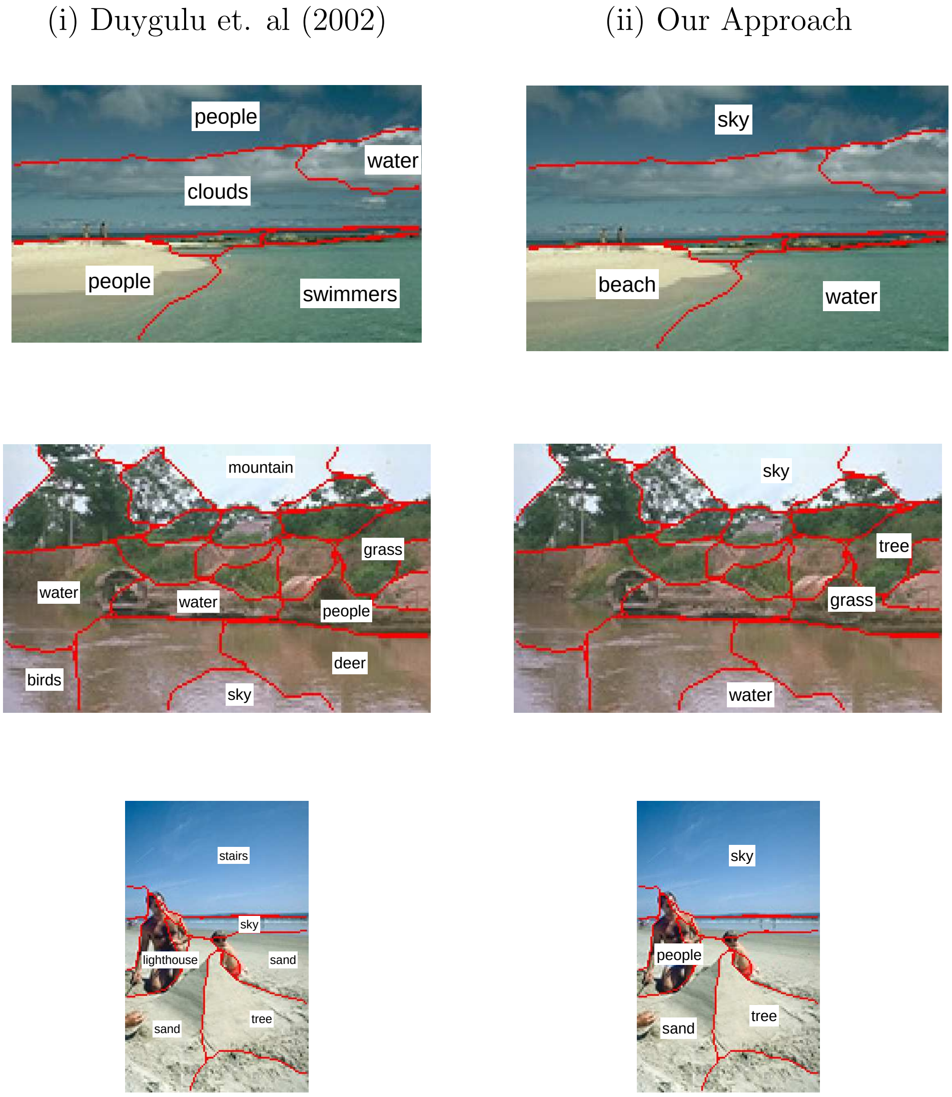
Figure 2.9: Labeling performance on set of 100 test images. We do not consider localization errors in this evaluation. Each image has on average 4 labels in the Corel dataset.



	Recall		Precision	
	Duygulu et. al	Ours	Duygulu et. al	Ours
Water	0.79	0.90	0.57	0.67
Grass	0.70	1.00	0.84	0.79
Clouds	0.27	0.27	0.76	0.88
Buildings	0.25	0.42	0.68	0.80
Sun	0.57	0.57	0.77	1.00
Sky	0.60	0.93	0.98	1.00
Tree	0.66	0.75	0.7	0.75

Precision-Recall

Figure 2.10: Precision-Recall ratios on common words in Corel-5K dataset.



(i) Duygulu et. al (2002)

(ii) Our Approach

Figure 2.11: Some examples of labeling on test dataset. By applying priors on relationships between different nouns, we can improve the labeling performance. For example, when labels are predicted independently, there can be labeling where region labeled “water” is above region labeled “clouds” as shown in the first image. This is however incongruent with the priors learned from training data where “clouds” are mostly above “water”. Bayesian reasoning over such priors and likelihoods lead to better labeling performance.

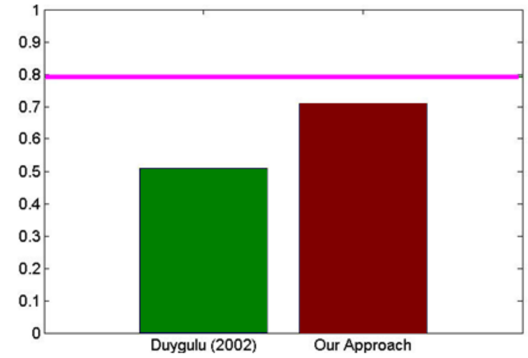


Figure 2.12: Performance on dataset with perfect segmentations. The measure used is the frequency correct measure.

Chapter 3

Visually Grounded Storyline Model for Video Understanding

In the end all we haveare stories and methods of finding and using those stories

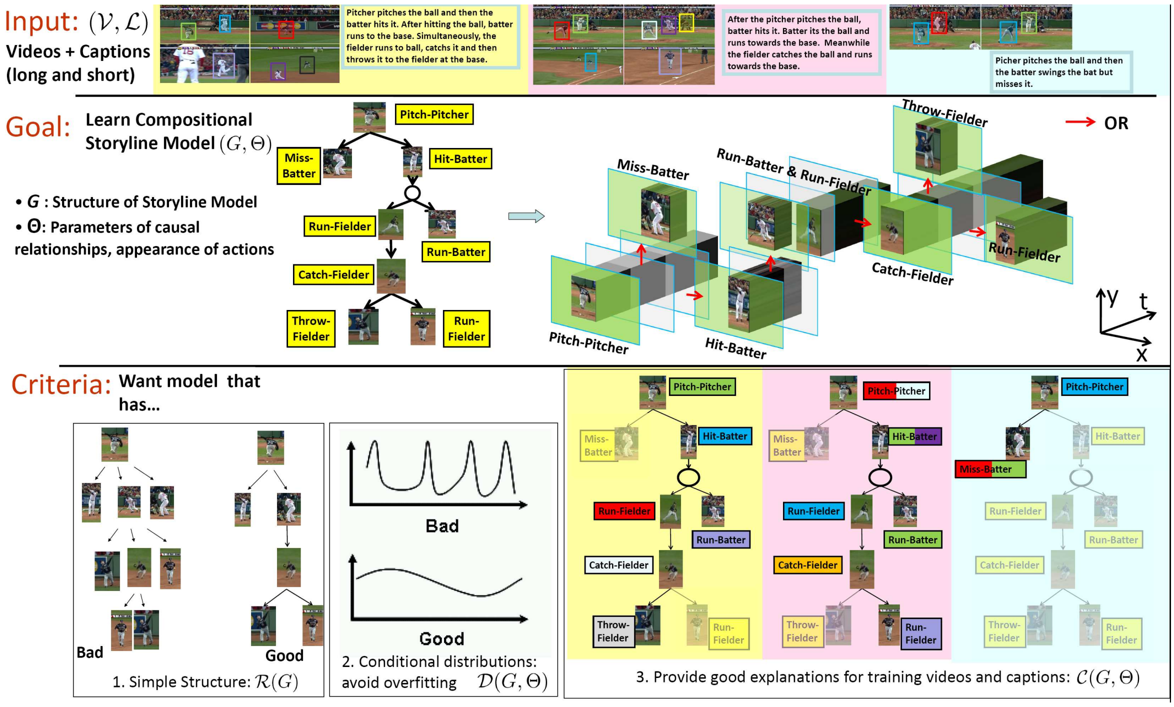
Roger C. Shank

Human actions are (typically) defined by their appearances/motion characteristics and the complex and structured causal dependencies that relate them. These causal dependencies define the goals and intentions of the agents. The *storyline* of a video includes the actions that occur in that video and causal relationships [79] between them. A model that represents the set of storylines that can occur in a video corpus and the general causal relationships amongst actions in the video corpus is referred to as a “storyline model”. Storyline models also indicate the agents likely to perform various actions and the visual appearance of actions. A storyline model can be regarded as a (stochastic) grammar, whose language (individual storylines) represents potential plausible “explanations” of new videos in a domain. Beyond recognition of individual actions, understanding the causal relationships among them provides information about the semantic meaning of the activity in video - the entire set of actions is greater than the sum of the individual actions. The causal relationships are often represented in terms of spatio-temporal relationships between actions. These relationships provide semantic/spatio-temporal context useful for inference of the storyline and recognition of individual actions in subsequent, unannotated videos.

The representational mechanism of the storyline model is very important; traditional action recognition has heavily utilized graphical models, most commonly Dynamic Bayesian networks (DBNs). However, the fixed structure of such models (often encoded by a domain expert) severely limits the storylines that can be represented by the model. At each time step, only a fixed set of actions and agents are available to model the video, which is not sufficient for situations in which the numbers of agents and actions varies. For example, in sports, sequences of actions are governed by the rules of the game and the goals of players/teams. These rules and goals represent a structure that extends beyond a simple fixed structure of recurring events. The set of possible or probable actions and/or agents at any given time may vary substantially. An important contribution of this work is the introduction of AND-OR graphs [80, 105] as a representation mechanism for storyline models. In addition, unlike approaches where human experts design graphical models, we learn the structure and parameters of the graph from weakly labeled videos using linguistic annotations and visual data. Simultaneous learning of storyline models and appearance models of actions constrains the learning process and leads to improved visual appearance models. Finally, we show that the storyline model can be used as a contextual model for inference of the storyline and recognition of actions in new videos.

Our approach to modeling and learning storyline models of actions from weakly labeled data is summarized in Figure 3.1. The storyline models are represented by AND-OR graphs, where selections are made at OR-nodes to generate storyline variations. For example, in the AND-OR graph shown in the figure, the 'pitching' OR-node has two children 'hit' and 'miss' which represent two possibilities in the storyline, i.e after pitching either a 'hit' or a 'miss' can occur. The edges in the AND-OR graph represent causal relationships and are defined in terms of spatio-temporal constraints. For example, an edge from 'catch' to 'throw' indicates that 'throw' is causally dependent on 'catch'(a ball can be thrown only after it has been caught). This causal relationship can be defined in terms of time as $t_{catch} < t_{throw}$. The causal relationship has a spatial constraint also - someone typically throws to another agent at a different location.

Our goal is to learn the storyline model and the visual groundings of each action from the weakly labeled data - videos with captions. We exploit the fact that actions have temporal orderings and spatial relationships, and that many actions either "causally" influence or are "causally" dependent on other actions. Humans learn these "causal" relationships between different actions by utilizing sources of information including language, vision and direct experience (interaction with the world). In our approach, we utilize human generated linguistic annotations of videos to support learning of storyline models.



Input: $(\mathcal{V}, \mathcal{L})$
Videos + Captions (long and short)

Pitcher pitches the ball and then the batter hits it. After hitting the ball, batter runs to the base. Simultaneously, the fielder runs to ball, catchs it and then throws it to the fielder at the base.

After the pitcher pitches the ball, batter hits it. Batter its the ball and runs towards the base. Meanwhile the fielder catches the ball and runs towards the base.

Picher pitches the ball and then the batter swings the bat but misses it.

Goal: Learn Compositional Storyline Model (G, Θ)

- G : Structure of Storyline Model
- Θ: Parameters of causal relationships, appearance of actions

Criteria: Want model that has…

1. Simple Structure: $\mathcal{R}(G)$

2. Conditional distributions: avoid overfitting $\mathcal{D}(G, \Theta)$

3. Provide good explanations for training videos and captions: $\mathcal{C}(G, \Theta)$

Figure 3.1: Visually Grounded Storyline-Model: Given annotated videos, we learn the storyline model and the visual grounding of each action. The optimization function for searching the storyline model has three terms: (1) Simple structure. (2) Connections based on simple conditional distributions. (3) Provides explanations for visual and text data in the training set. The figure also shows how our AND-OR graph can encode the variations in storylines (three videos at the top with different storylines (bottom-right)), not possible with graphical models like DBNs.

3.1 Related Work

Existing datasets for learning action appearance models provide samples for a few classes and in controlled and simplified settings. Such datasets fail to generalize to actions with large intra-class variations and are unsuitable for learning contextual models due to unnatural settings. On the other hand, using realistic videos would require significant human labeling effort, making it infeasible to create such datasets for learning contextual models. There has been recent interest in utilizing large amounts of weakly labeled datasets, such as movies/TV shows in conjunction with scripts/subtitles. Approaches such as [29, 57] provide assignment of frames/faces to actions/names. Such approaches regard assignment and appearance learning as separate process. Therefore, these approaches do not utilize the co-occurrence statistics of visual features and the internal structure of videos for assignment. Nitta et al. [77] present an approach to annotate sports videos by associating text to images based on previously specified knowledge of the game. In contrast we simultaneously learn a storyline model of the video corpus and match tracked humans in the videos to action verbs (i.e, solving the segmentation and correspondence problems).

Our approach is motivated by work in image annotation which typically model the joint distribution of images and keywords to learn keyword appearance models [5]. Similar models have been applied to video retrieval, where annotation words are actions instead of object names [34]. While such models exploit the co-occurrence of image features and keywords, they fail to exploit the overall structure in the video.

In previous chapter, we presented an approach to simultaneously learn models of both nouns and prepositions from weakly labeled data. Visually grounded models of prepositions are used to learn a contextual model for improving labeling performance. However, spatial reasoning is performed independently for each image. Some spatial reasoning annotations in the images are not incidental and can be shared across most images in the dataset (For example, for all the images in the dataset sun is above water). In addition, the contextual model based on priors over possible relationship words restricts the clique size of the Bayesian network used for inference. Also, it requires a fully connected network, which can lead to intractable inference. In contrast, our approach learns a computationally tractable storyline model based on causal relationships that generally hold in the given video domain.

There has been significant research in using contextual models for action recognition [41, 94, 39, 14, 74]. Much of this work has focused on the use of graphical models such as Hidden Markov Models (HMMs) [99], Dynamic Bayesian Networks (DBNs) [39] to model contextual relationships among actions. A drawback of these approaches is their fixed structure, defined by human experts. The set of probable actions and/or agents at any given time may vary greatly, so a fixed set of successor actions is insufficient. Our AND-OR graph storyline model can model both contextual relationships (like graphical models) while simultaneously modeling variation in structure (like grammars [14]). For example, in sports, sequences of actions are governed in part by the rules of the game. These rules represent a structure that extends beyond a simple fixed structure of recurring events. The set of possible or probable actions and/or agents at any given time may depend on events that occurred well in the past, so a fixed set of possible successor actions is insufficient. Our AND-OR graph storyline model can model both contextual relationships (like graphical models) while simultaneously modeling variation in structure (like grammars).

In computer vision, AND-OR graphs have been used to represent compositional patterns [105, 24]. Zhu and Mumford [105] used AND-OR graph to represent a stochastic grammar of images. Zhu et. al [103] present an approach to learn AND-OR graphs for representing an object shape directly from weakly supervised data. Lin et. al [61] also used an AND-OR graph representation for modeling activity in an outdoor surveillance setting. While their approach assumes hand-labeled annotations of spatio-temporal relationships and AND-OR structure is provided, our approach learns the AND-OR graph structure and its parameters using text based captions. Furthermore, [61] assumes one-one correspondence between nodes and tracks as compared to one-many correspondence used in our approach.

Our work is similar in spirit to structure learning of Bayesian networks in [36], which proposed a structural-EM algorithm for combining the standard EM-algorithm for optimizing parameters with search over the Bayesian network structure. We also employ an iterative approach to search for parameters and structure. The structure search in [36] was over the space of possible edges given a fixed set of nodes. However, in our case, both the nodes and edges are unknown. This is because a node can occur more than once in a network, depending upon the context in which it occurs (See figure 3.2). Therefore, the search space is much larger than the one considered in [36].

3.2 Storyline Model

We model the storyline of a collection of videos as an AND-OR graph $G = (V_{and}, V_{or}, E)$. The graph has two types of nodes - OR-nodes, V_{or} and AND-nodes V_{and}. Each OR-node $v \in V_{or}$ represents an action which is described by its type and agent. Each action-type has a visual appearance model which provides visual grounding for OR-nodes. Each OR-node is connected to other OR-nodes either directly or via an AND-node. For example, in Fig 3.1, middle, the OR-node "Pitch" has two OR-children which represents two possibilities after a pitch (i.e either the batter hits the ball ("Hit-Batter") or misses it ("Miss-Batter"). A path from an OR-node v_i to an OR-node v_j (directly or via an AND-node) represents the causal dependence of action v_j upon action v_i. Here, AND-nodes are dummy nodes and only used when an activity can causally influence two or more simultaneous activities. The causal relationships between two OR-nodes are defined by spatio-temporal constraints. For example, the causal relationship that 'hitting' depends on 'pitching' the ball can be defined temporally as $t_{pitch} < t_{hit}$ (hitting occurs after pitching) and spatially as the pitcher must be some distance d' from the batter $d(pitch, hit) \approx d'$. Figure 3.1 shows several examples (top) of videos whose actions are represented by AND-OR graphs. Note that the AND-OR graph can simultaneously capture both

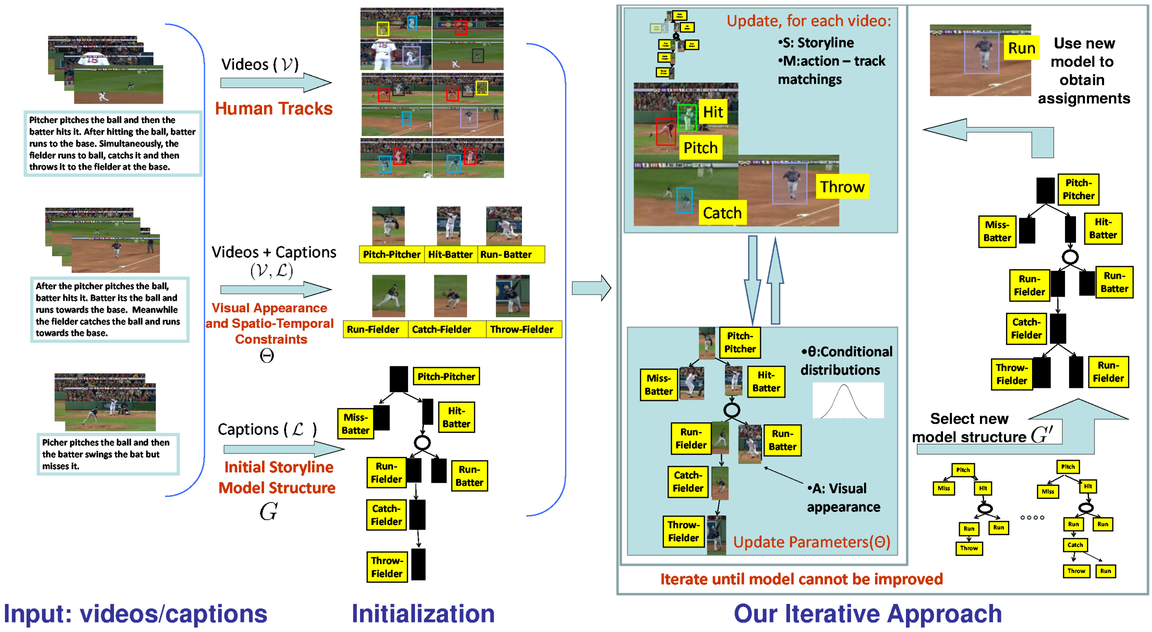

Pitcher pitches the ball and then the batter hits it. After hitting the ball, batter runs to the base. Simultaneously, the fielder runs to ball, catchs it and then throws it to the fielder at the base.
After the pitcher pitches the ball, batter hits it. Batter its the ball and runs towards the base. Meanwhile the fielder catches the ball and runs towards the base.
Picher pitches the ball and then the batter swings the bat but misses it.
Videos ($\mathcal{V}$)
Human Tracks
Videos + Captions ($\mathcal{V}, \mathcal{L}$)
Visual Appearance and Spatio-Temporal Constraints Θ
Captions ($\mathcal{L}$)
Initial Storyline Model Structure G
Pitch-Pitcher
Hit-Batter
Run- Batter
Run-Fielder
Catch-Fielder
Throw-Fielder
Miss-Batter
Update, for each video:
•S: Storyline
•M:action – track matchings
Hit
Pitch
Throw
Catch
•θ:Conditional distributions
•A: Visual appearance
Update Parameters(Θ)
Iterate until model cannot be improved
Run
Use new model to obtain assignments
Select new model structure G'
Pitch
Miss
Hit
Run
Throw
Catch
Input: videos/captions
Initialization
Our Iterative Approach

Figure 3.2: An Overview of our approach; our storyline model (G, Θ) is initialized using videos and captions, and we propose an iterative procedure to improve its parameters Θ and the structure G.

long and short duration storylines in a single structure (bottom-right).

3.3 Learning the Storyline Model

Our goal is to learn a visually grounded storyline model from weakly labeled data. Video annotations include names of actions in the videos and some subset of the temporal and spatial relationships between those actions. These relationships are provided by both spatial and temporal prepositions such as “before”, “left” and “above”. Each temporal preposition is further modeled in terms of the relationships described in Allen’s Interval Logic. As part of bottom-up processing, we assume that each video has a set of human-tracks, some of which correspond to actions of interest. The feature vector that describes each track is based on appearance histograms of Spatio-Temporal Interest Points (STIPs) [56, 76] extracted from the videos.

Establishing causal relationships between actions and learning groundings of actions involves solving a matching problem. We need to know which human-tracks in the training videos match to different action-verbs of the storyline to learn their appearance models and the storyline-model of videos. However, matching of tracks to action-verbs and storyline extraction of a particular video depends on the structure of the storyline-model, the appearances of actions and causal relationships between them. This leads to yet another chicken-and-egg problem, and we employ a structural EM-like iterative approach to simultaneously learn the storyline-model and appearance models of actions from collections of annotated videos. Formally, we want to learn the structure G and parameters of the storyline model $\Theta = (\theta, A)$ (θ-Conditional Distributions, A-Appearance models), given the set of videos $(\mathcal{V}_1..\mathcal{V}_n)$ and their associated annotations $(\mathcal{L}_1..\mathcal{L}_n)$:

$$\begin{aligned}(G,\Theta) &= \arg\max_{G',\Theta'} P(G',\Theta'|\mathcal{V}_1..\mathcal{V}_n,\mathcal{L}_1..\mathcal{L}_n)\\ &\propto \arg\max_{G',\Theta'} \prod_i \sum_{M^i,S^i} P(\mathcal{V}_i,\mathcal{L}_i|G',\Theta',M^i,S^i)P(G',\Theta')\end{aligned}$$

- S^i : Storyline for video i.
- M^i : Matchings of tracks to actions for video i.

We treat both S and M as missing data and formulate an EM-approach. The prior, $P(G, \Theta)$, is based on simple structure ($\mathcal{R}(G)$) and simple conditional distributions terms ($\mathcal{D}(G, \Theta)$) and the likelihood terms are based on how well the storyline model generates storylines which can explain both the videos and their linguistic annotations($\mathcal{C}(G, \Theta)$).

Figure 3.2 summarizes our approach for learning visually grounded storyline-models of training videos. Given an AND-OR graph structure at the beginning of an iteration, we fix the structure and iterate between learning parameters/visual grounding of the AND-OR graph and the matching of tracks to action nodes(Sec. 3.3.1). In the hard-E step, we estimate storyline and matchings for all training videos using the current G, Θ. In the M step, we update Θ using the estimated storylines and matchings for all videos. After convergence or a few iterations, we generate new graph proposals by local modifications to the original graph (Sec. 3.3.2) and select the modification that best represents the set of storylines for the videos(Sec. 3.3.3). This new storyline model is then used for re-initializing the iterative approach, which iterates between appearances and matchings. The new storyline model, which is a better model of the variations in storylines across the training videos, allows better interpretation. For example, in the figure, the "run-fielder" action after the "catch" was labeled as "throw" since the initial storyline-model did not allow "run" after "catch". Subsequently, an updated storyline model allows for "run" after "catching", and the assignments improve because of the new expanded storyline.

We begin by explaining our model, and parsing procedure we use to analyze a video using our model, inferring the storyline of the video and matching of video segments to the actions in the storyline. Afterwards, we explain our procedure for learning a model from weakly-labeled video.

3.3.1 Parsing Videos

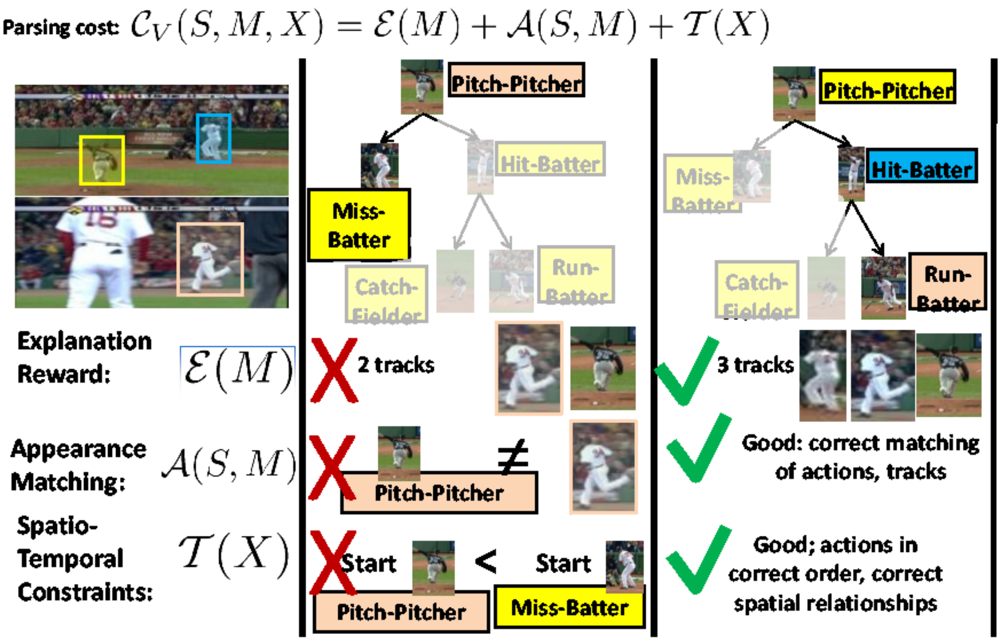
We now describe how, an AND-OR storyline model is used to analyze, or parse, videos and obtain their storylines and matchings of human tracks to storyline actions. We provide a one-many matching formulation, where several human tracks can be matched to a single action. Matching of tracks to actions also requires making a selection at each OR-nodes to select one storyline out of the set of possible storylines. While there have been several heuristic inference algorithms for AND-OR graphs, we formulate an integer programming approach to obtain the storyline and matchings, and solve a relaxed version of the problem in the form of a linear program.

Given an AND-OR graph G, a valid instantiation, S(representing a storyline), of the AND-OR graph is a function $S : i \in V_{and} \cup V_{or} \rightarrow \{0, 1\}$ that obeys the following constraints: (1) At each OR-node v_i there is an associated variable S_i which represents whether the or-node has been selected for a particular storyline of not. For example, in fig 3.3, 'hit' is a part of the storyline, therefore $S_3 = 1$, and miss is not a part of the storyline so $S_2 = 0$. (2) Since OR-children represent alternate possible storyline extensions, exactly one child can be selected at each OR-node. (3) An OR-node, i, can be instantiated (i.e $S_i = 1$) only when all the OR-nodes in the unique path from the root to node i have been instantiated. For example, since the path from 'pitching' to 'catching' includes 'hitting', 'catching' can be part of a storyline if and only if 'hitting' is part of the storyline.

Given T human tracks in a video, a matching of tracks and nodes is a mapping $M : i \in V_{or}, j \in \{1, ..., T+1\} \rightarrow \{0, 1\}$. $M_{ij} = 1$ indicates that the action at the OR-node i is matched with track j. Since some of the actions might not be visible due to occlusion and camera-view, we add a dummy track which can be associated with any action with some penalty. Depending on the constraints imposed on M, different matchings between actions and tracks can be allowed: many-to-many, many-to-one, one-to-many, or one-to-one. We consider those mappings that associate one action to many tracks, which is represented by the constraint $1 * M^T = 1$. Furthermore, no tracks should be matched to an OR node that is not instantiated: $\forall i \in V_{or}$, $M_{ij} \leq S_i$.

Finally, to incorporate pairwise constraints (such as temporal ordering and spatial relationships) between matches of two nodes i and k, M_{ij} and M_{kl}, we introduce variables $\boldsymbol{X} : \boldsymbol{x_{ijkl}} \in \boldsymbol{\{0, 1\}}$; $\boldsymbol{x_{ijkl} = 1}$ **indicates that the action at node i and track j are matched, and the action at node k and track l are matched.** Instead of enforcing a computationally difficult hard constraint $x_{ijkl} = M_{ij} * M_{kl}$, we marginalize both sides over l and represent the constraint as: $\forall k, \sum_l x_{ijkl} = M_{ij}$.

In parsing, we search for a "best" valid instantiation S (representing a storyline



Parsing cost: $\mathcal{C}_V(S,M,X) = \mathcal{E}(M) + \mathcal{A}(S,M) + \mathcal{T}(X)$
Pitch-Pitcher
Hit-Batter
Miss-Batter
Catch-Fielder
Run-Batter
Explanation Reward:
$\mathcal{E}(M)$
2 tracks
3 tracks
Appearance Matching:
$\mathcal{A}(S,M)$
≠
Good: correct matching of actions, tracks
Spatio-Temporal Constraints:
$\mathcal{T}(X)$
Start
<
Start
Good; actions in correct order, correct spatial relationships

Figure 3.3: Given the upper left video, we show two possible parses and costs for each parsing cost function component. The left parse is worse since it explains fewer tracks (Explanation $\mathcal{E}(M)$), matches actions to tracks with different appearance (Appearance $\mathcal{A}(S,M)$), and violates spatial and temporal constraints (the pitch-pitcher action should be matched to a track which occurs before the miss-batter track, and the two tracks should appear in the usual pitcher-batter configuration) (Spatio -Temporal $\mathcal{T}(X)$). The correct parse on the right scores well according to all three cost function components.

from the storyline model) and a matching M of tracks and actions (representing visual grounding of nodes). The optimization function for selection is based on three terms: (1) Select a storyline consisting of nodes for which good matching tracks can be found. (2) Select a storyline which can explain as many human tracks as possible. (3) The matching of nodes to tracks should not violate spatio-temporal constraints defined by the storyline model (See Figure 3.3). The three terms that form the basis of the objective to be minimized, subject to the above constraints on S, M and X, are:

Appearance Matching: The cost of a matching is based on the similarity of the appearances of instantiated nodes and corresponding tracks. This cost can be written as:

$$\mathcal{A}(S,M) = \sum_i ||(\sum_j M_{ij})t_j - S_i A_i|| \tag{3.1}$$

where t_j represents the appearance histogram of track j and A_i represents the appearance histogram model of the action at node i. In many to one matching, multiple tracks combine to match to a single action node. Therefore, the first term,$(\sum_j M_{ij} t_j)$, sums the appearance histograms of human tracks that match to node i. This is then compared to the appearance model at node i by measuring the L1-norm. Figure 3.3 shows an example of parsing with high(left parse) and low matching costs(right parse). The left parse has high matching cost since the track of a batter running is assigned to the pitching node which are not similar in appearance.

Explanation Reward: Using only appearance matching would cause the optimization algorithm to prefer small storylines, since they require less matching. To remove this bias, we introduce a reward, $\mathcal{E}$, for explaining as many of the STIPs as possible. We compute $\mathcal{E}$ as:

$$\mathcal{E}(M) = -\sum_j min(\sum_i M_{ij}, 1)||t_j|| \tag{3.2}$$

This term computes the number of tracks/STIPs that have been assigned to a node in the AND-OR graph and therefore explained by the storyline model.

Spatio-Temporal Constraints: We also penalize matchings which violate spatio-temporal constraints imposed by causal relationships. If p_{ijkl} encodes the violation cost of having an incompatible pair of matches (node i to track j and node k to track l), the term for spatio-temporal violation cost is represented as: $\mathcal{T}(X) = \sum_{ijkl} p_{ijkl}x_{ijkl}$. This term prefers matchings that do not violate the spatio-temporal constraints imposed by the learned AND-OR graph. For example, the left parse in Figure 3.3 matches the 'pitching' and 'miss' actions to incorrect tracks, resulting in 'pitching' starting after 'batting' in the video, which is physically impossible. The tracks are also not in the typical pitcher-batter spatial configuration. Therefore, this matching has a high cost as compared to the matching shown in the right parse.

The above objective and the constraints result in an Integer Program which is a NP-Hard problem. We approximate the solution by relaxing the variables S, M and X to lie in $[0, 1]$. The result is a linear program, which can be solved very quickly. For the learning procedure, we have the annotated list of actions that occur in the video. We utilize these annotations to obtain a valid instantiation/storyline S and then optimize the function over M, X only. For inference, given a new video with no annotations, we simultaneously optimize the objective over S, M, X.

3.3.2 Generating new Storyline Model Proposals

After every few inner iterations of the algorithm, we search for a better storyline model to explain the matchings and causal-relationships between actions. To do this, we generate new graph proposals based on local modifications to the AND-OR graph structure from the previous iteration.

The local modifications are: (1) Deletion of an edge and adding a new edge (2) Adding a new edge (3) Adding a new node. The selection of edges to delete and add is random and based on the importance sampling procedure, where deletion of important edges are avoided and addition of an important edge is preferred. The importance is defined on the basis of the likelihood that the head and tail of the edge are related by a causal relationship.

3.3.3 Selecting the New Storyline Model

Each iteration selects the AND-OR graph from the set of modifications which best represents the storylines of the training videos. The criteria for selection is based on four different terms:

Track Matching Likelihood: The first criteria measures how well a proposal explains the matchings obtained in the previous parsing step. The matching of tracks to actions from the previous step is used to obtain a likelihood of an AND-OR graph generating such a matching. The likelihood of the p^{th} graph proposal, G_r^p generating the pairwise matchings X^{r-1} (at iteration $r-1$) is given by $\frac{1}{Z}exp(-\mathcal{T}_{G_r^p}(X^{r-1}))$. This likelihood is based on the third term from the parsing cost, but here the penalty terms are computed with respect to the individual graph proposals.

Annotation Likelihood: The AND-OR graph representing the storyline model should not only explain the matching of tracks to actions, but also the linguistic annotations associated with each video. The underlying idea is that the same storyline model is used to generate the visual data and linguistic annotations. The cost function measures how likely an instantiation of the AND-OR graph storyline model accounts for the video's actions annotations and how well the constraints specified by linguistic prepositions in annotations are satisfied by the AND-OR graph constraints. For example, if the annotation for a training video includes 'pitching before hitting', a good AND-OR graph would not only generate a storyline including 'pitching' and 'hitting' but also have the conditional distribution for the edge pitching → hitting, such that $P(t_{hit} - t_{pitch} > 0|\theta)$ is high.

Structure Complexity If we only consider likelihoods based on linguistic and visual data, more complex graphs which represent large numbers of possibilities will always be preferred over simple graphs. Therefore, an important criteria for selection of an AND-OR graph is that it should be simple. This provides a prior over the space of possible structures. We use a simplicity prior similar to [37], which prefers linear chains over non-linear structures.

Distribution Complexity The complexity of an AND-OR graph depends not only on its graph structure, but also the conditional distributions of children actions given parent actions. For an action i (OR-node) in an AND-OR graph, we form a distribution over all possible successors, or sets of actions that could appear immediately after action i in a storyline. The individual spatio-temporal conditional distributions between i and its successors are combined into a single distribution over successors, and we compute the entropy of this combined distribution. The entropies of the successor distributions for all OR-nodes in the graph are averaged, providing a measure of the complexity of the conditional distributions contained in the AND-OR graph. Our cost prefers higher entropy distributions; empirically, we have found that this results in better ranking of structures. We can also draw intuition from work on maximum entropy Markov models [65], where higher entropy distributions are preferred in learning conditional distributions to prevent overfitting.

3.3.4 Initializing the Search

For initialization, we need some plausible AND-OR causal graph to represent the storyline model and appearance models of actions. Establishing a causal se-

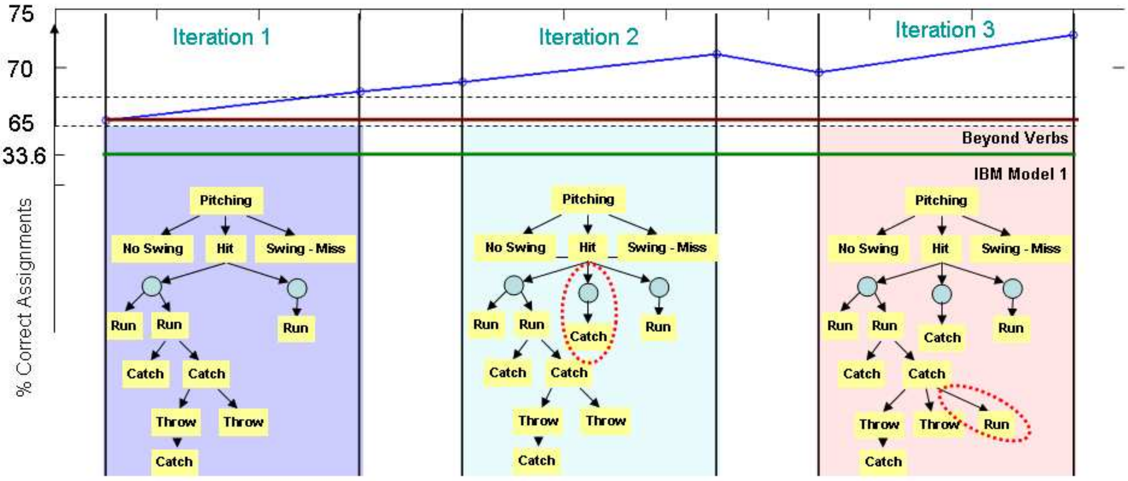


Figure 3.4: Quantitative evaluation of training performance and how the storyline model changes with iterations. Within each colored block, the storyline model remains fixed and the algorithm iterates between parsing and parameter estimation. At the end of each colored block, the structure of the AND-OR graph is modified and a new structure is learned. The structural changes for the three iterations are shown.

quence of actions from passive visual data is a difficult problem. While one can establish a statistical association between two variables X and Y, inferring causality - whether $X \rightarrow Y$ or $Y \rightarrow X$- is difficult. For initialization, we use the linguistic annotations of the videos. Based on psychological studies of causal learning, we use 'time' as a cue to generate the initial storyline model [55]. If an action A immediately precedes action B, then A is more likely to be the cause and B is more likely to be the effect.

We initialize the AND-OR graph with the minimum number of nodes required to represent all the actions in the annotations of the training videos. Some actions

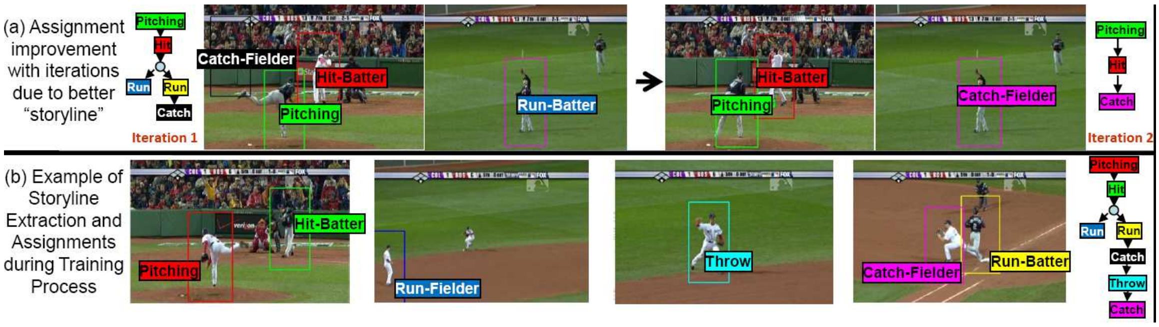


Figure 3.5: (a) Improvement in assignments with iterations: In the first iteration, the true storyline for the video pitching → hit → catching, is not a valid instantiation of the AND-OR graph. The closest plausible storyline involves activities like run which have to be hallucinated in order to obtain assignments. However, as the storyline model improves in iteration 2, the true storyline now becomes a valid storyline and is used for obtaining the assignments. (b) Another example of the assignments obtained in training. The assignments are shown by color-coding, each track is associated to the node which has similar color in the instantiated AND-OR graph.

might have more than one node due to their multiple occurrences in the same video and due to different contexts under which the action occur. For example, ‘catch-fielder’ can occur in a video under two different contexts. The action ’catching’ in the outfield and ’catching’ at a base are different and require different nodes in the AND-OR graph. Using Allen’s interval temporal logic, we obtain the weight of all possible edges in the graph, which are then selected in a greedy manner such that there is no cycle in the graph. Dummy AND-nodes are then inserted by predicting the likelihood of two activities occurring simultaneously in a video.

For initialization of appearance models, we use the approach proposed in the previous chapter. Using the spatio-temporal reasoning based on the prepositions and the co-occurrence of visual features, we obtain a one-one matching of tracks to actions which is used to learn the initial appearance models.

3.4 Experimental Evaluation

For our dataset, we manually chose video clips of a wide variety of individual plays from a set of baseball DVDs for the 2007 World Series and processed them as follows: We first detect humans using the human detector[25]. Applied to each frame with a low detection threshold, the output of the detector is a set of detection windows which potentially contain humans. To create tracks, we perform agglomerative clustering of these detection windows over time, comparing windows in nearby frames according to the distance between their centroids, and similarity of color histograms as measured by the Chi-square distance. The resulting tracks can be improved by extending each track forwards and backwards in time using color histogram matching. STIPs that fall within the detection window of a track in a frame contribute to the track's appearance histogram.

Training: We trained the storyline model on 39 videos (individual baseball plays), consisting of approximately 8000 frames. The training videos contained both very short and very long plays. We evaluate the performance of our training algorithm in terms of number of actions correctly matched to tracks. Figure 3.4 shows how this accuracy changed over the training process. The figure is divided into three colored blocks. Within each colored block, the structure of the storyline model remains the same and the approach iterates between parsing and parameter update. At the end of each colored block, we update our storyline model and select a new storyline model which is then used to parse videos and estimate parameters. We can see that the accuracy rises significantly over the course of training, well above the initial baselines, validating our iterative approach to training. The percentage improvement over the "Beyond Nouns" approach explained in previous chapter is as much as 10%.

Figure 3.5 a) shows an example of how a parse for a video improves with iterations. Figure 3.5 b) shows an additional example of a video with its inferred storyline and matchings of actions to tracks; we can see that all but the run-fielder action are correctly matched.

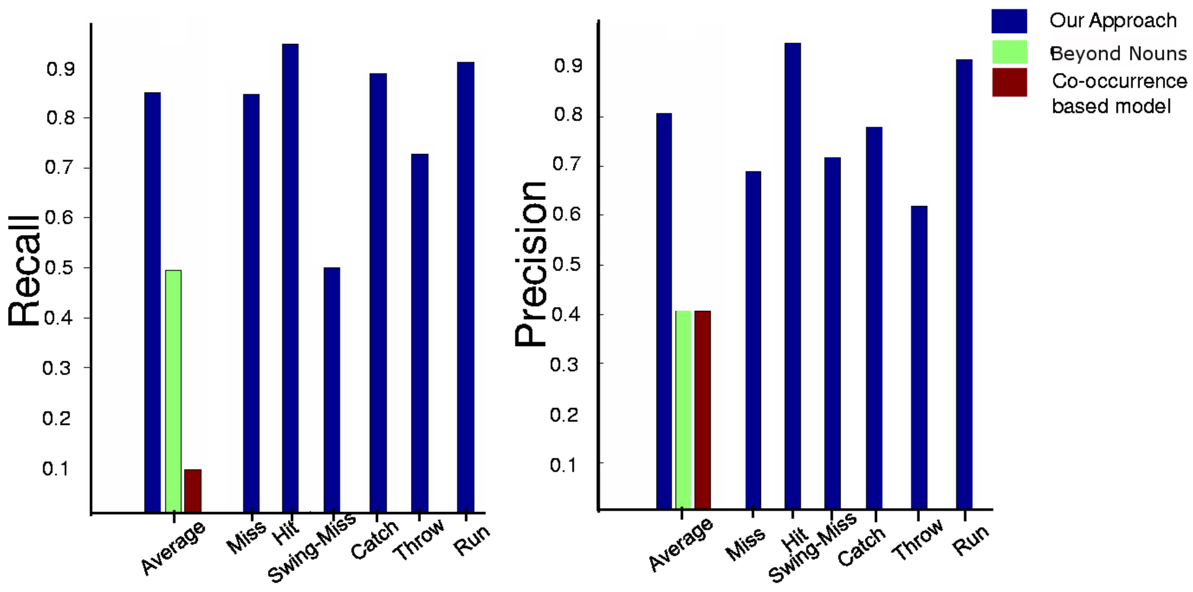
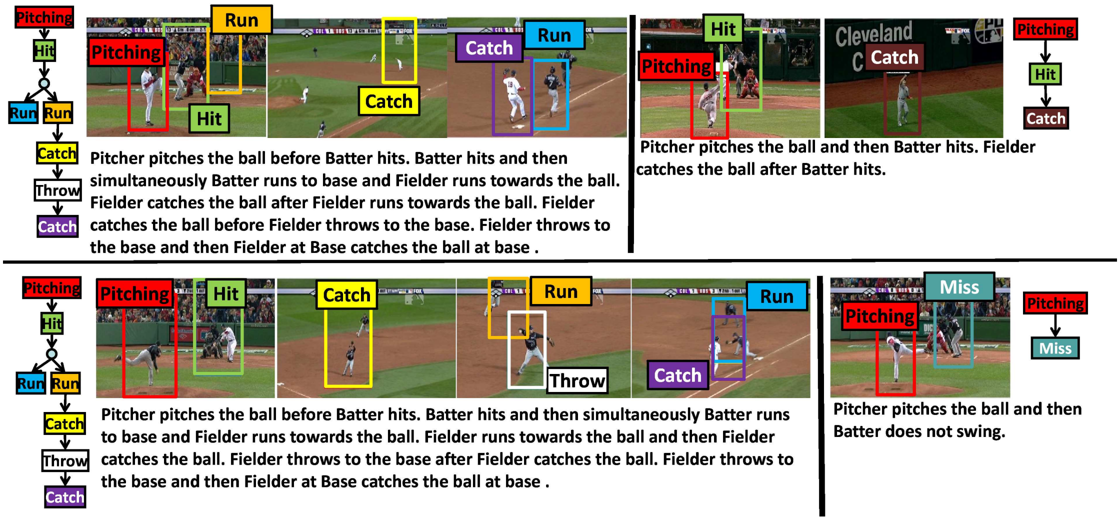


Figure 3.6: Quantitative Evaluation of Labeling Accuracy



Pitcher pitches the ball before Batter hits. Batter hits and then simultaneously Batter runs to base and Fielder runs towards the ball. Fielder catches the ball after Fielder runs towards the ball. Fielder catches the ball before Fielder throws to the base. Fielder throws to the base and then Fielder at Base catches the ball at base .

Pitcher pitches the ball and then Batter hits. Fielder catches the ball after Batter hits.

Pitcher pitches the ball before Batter hits. Batter hits and then simultaneously Batter runs to base and Fielder runs towards the ball. Fielder runs towards the ball and then Fielder catches the ball. Fielder throws to the base after Fielder catches the ball. Fielder throws to the base and then Fielder at Base catches the ball at base .

Pitcher pitches the ball and then Batter does not swing.

Figure 3.7: Storyline Extraction for New Videos: We show the instantiation of AND-OR graph obtained for each training video and the story generated in text by our algorithm. The assignments of tracks to action nodes are shown by color coding.

Storyline Extraction for New Videos: Our test set includes 42 videos from the same baseball corpus. Again, they ranged from very short and simple to longer and more complicated. We first evaluated the performance in terms of storyline extraction. Fig.3.7 shows some qualitative examples of the storyline extraction in terms of the instantiation of AND-OR graphs, assignment of tracks to actions and the text that is generated from the storyline model. We use recall and precision values of action labeling to measure the performance of storyline extraction. We compare the performance of our approach to the baseline methods of Gupta et.al [42] and IBM Model 1[19]. Figure 3.6 shows two bar plots, one for recall (left) and the other for precision (right). For the baseline methods, we show the average precision and recall values and compare against our method's performance (block of blue, red and green bars). Our method nearly doubles the precision of the baseline methods (.8 vs. .4), and has a much higher recall (.85 vs. 0.5 for [42] and 0.1 for [19]). It performs well over most of the actions, with the exception of the action Swing-Miss (low recall). We also evaluated the number of correct matchings obtained for the actions in the predicted storylines. Quantitatively, we obtained **70%** correct assignments of tracks to actions.

We attribute the success of our approach to three reasons: (1) An important reason for improvement in training compared to the previous chapter is that we did not feedback the contextual models learned at the end of their single iterative loop of training to relearning models of object appearances. (2) During inference, the coupling of actions via the AND-OR graph model provides a more structured model than simple context from co-occurrence statistics and binary relationship words can provide. (3) The one-many (action to track matching) framework used here is more powerful than the one-one framework in the previous chapter and handles the problem of fragmented segmentation.

Chapter 4

Function Recognition - Linking Nouns and Verbs

In chapters 2 and 3, we presented contextual models which are based on noun-noun spatial relationships and verb-verb causal relationships (characterized by spatio-temporal distributions). In this chapter, we investigate how noun-verb relationships can be used in contextual models for constraining the recognition problem. Specifically, we present a Bayesian approach for interpretation of human-object interactions, that integrates information from perceptual tasks such as scene analysis, human motion/pose estimation [1], manipulable object detection and "object reaction" determination [2]. While each of these tasks can be conducted independently, recognition rates improve when we integrate information from different perceptual analysis and also consider spatial and functional constraints.

Integrating information from different perceptual analyses enables us to form a coherent semantic interpretation of human object interactions. Such an interpre-

[1] Recognition of action in static images is based on "implied" motion. "Implied" motion refers to the dynamic information implicit in the static image [52]. The inference of action from static images depends on implied motion, which itself depends on the phase of the action [53, 95]. This indicates that human pose provides important cues for action recognition in static images

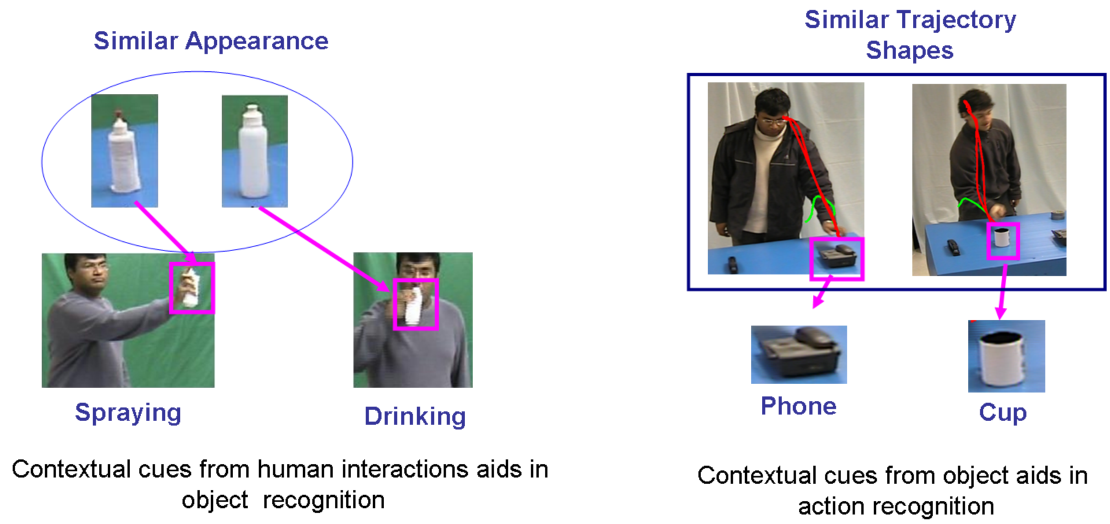
[2] Object reaction is the effect of manipulation of an object by human actor

tation not only supports recognizing the interactions, but also the objects involved in those interactions and the effect of those interactions on those objects.

Interactions between different perceptual analyses allows us to recognize actions and objects when appearances are not discriminative enough. Consider two objects, such as the spray bottle and a drinking bottle shown in Figure 4.1. These objects are similar in appearance and shape, but have different functionality. Due to their functional dissimilarity, people's interaction with these objects provides context for their recognition. Similarly, two similar human movements/poses can serve different purposes depending on the context in which they occur. For example, the poses of the humans shown in Figure 4.2 are similar, but due to the difference in context, the first action is inferred to be running and the second action to be kicking.

Another important element in the interpretation of human object interactions is the effect of manipulation on objects. When interaction movements are too subtle to observe using computer vision, the effects of these movements can provide information on functional properties of the object. For example, when lighting a flashlight, recognizing the pressing of a button might be very difficult. However, the resulting illumination change can be used to infer the manipulation.

We present two computational models for interpretation of human object interactions in videos and static images, respectively. Our approach combines action recognition and object recognition in an integrated framework, and allows us to apply spatial and functional constraints for recognition. The significance of our

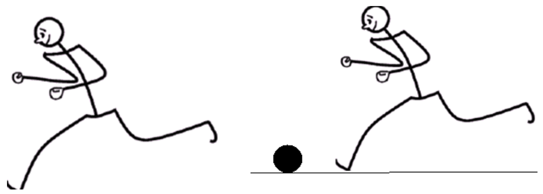


Contextual cues from human interactions aids in object recognition

Contextual cues from object aids in action recognition

Figure 4.1: Importance of interaction context in recognition of object and vice-versa. While the objects might be difficult to recognize using shape features alone, when interaction context is applied the object is easy to recognize. Similarly, two actions might have similar dynamics and trajectories. It is difficult to differentiate between two actions based on shape of trajectories. However, when cues from object are used in conjunction with cues from human dynamics it is easy to differentiate between two actions.

approach is threefold: (a)Human actions and object reactions are used to locate and recognize objects which might be difficult to locate or recognize otherwise. (b) Object context and object reactions are used to recognize actions which might otherwise be too similar to distinguish or too difficult to observe. In some cases, such as in recognition of actions from static images, there is no dynamic information; however contextual information can be used in such cases for recognition. (c) We provide an approach for recognition of actions from "static" images. The extraction



(a)Running (b)Kicking

Figure 4.2: Action recognition from static images requires contextual information. Same poses can have different meanings based on the context.

of "dynamic information" from static images has been well studied in the fields of psychology and neuroscience, but has not been investigated by the computer vision community.

4.1 Related Work

4.1.1 Psychological Studies

Milner and Goodale [66] proposed psychological theories of human information processing where action execution and object perception are considered two separate processes with their own pathways in the human brain. However, with the discovery of mirror neurons in monkeys, there has been a renewed interest in studying the relationships between object recognition, action understanding and action execution [75, 38, 40]. With the same neurons involved in execution and perception, a link between object recognition and action understanding has been established [75] in humans. Gallese et. al [38] showed that movement analysis in humans depends on the presence of objects. The cortical responses for goal directed actions are different from the responses evoked when the same action is executed but without the presence of the object. In another study, Frey et. al [48] showed that human inferior frontal cortex responds to static pictures of human object interactions. The response was only observed in the presence of congruent poses and objects, suggesting that human poses are evaluated in the context of objects. On the other hand, the importance of action in perceiving and recognizing objects (especially manipulable objects like tools) has been shown [23].

Recent studies in experimental psychology have also confirmed the role of object recognition in action understanding and vice-versa. Helbig et. al [46] show the role of action priming in object recognition and how recognition rates improve with action-priming. Recognition rates of target objects were higher when the priming object was used in a similar action as the target object. In another study, Bub et. al [20] investigated the role of object priming in static gesture recognition. While passive viewing of an object did not lead to priming effects, priming was observed when humans were first asked to recognize the object and then recognize the image of a related hand gesture. In a recent study, Bach et. al [4] showed that when actions involving objects are perceived, spatial and functional relations provide context in which these actions are judged. These studies suggest that humans perceive implied motion from static poses under object and scene context.

While most of this work suggests interactions between object and action perception in humans, they have not examined the nature of the interaction between action and object recognition. Vaina et. al [96] address this through the study of pantomimes. They ranked the properties of objects that can be estimated robustly by perception of pantomimes of human-object interaction. They discovered that the weight of an object is most robustly estimated, while size and shape are harder to estimate.

Our computational model for recognition from static image is motivated from psychological studies of extraction of dynamic information from static images. The human visual system is highly tuned to perceive motion and produce dynamic information. Psychophysical studies have shown that humans not only tend to infer motion from static images, but they also store pose representations as if the object/agent was indeed moving [35]. Neuro-psychological studies of monkeys have shown that cortical cell responses to static posture were related to the implied action rather than the static posture per se [49]. The responses of the cells are different for implied human motions as compared to observation of non-biological entities (e.g., flowing water) with implied motion. In the case of implied human motion, TMS studies have shown the specific motor activation of muscles involved in the execution of the very same action [95].

4.1.2 Computational Approaches

There has been a very large body of work carried out in both, object recognition and action recognition. Most approaches, however, address one or both of these problems, independent of the other.

Computational approaches for object recognition typically use local static features, based on shape and textural appearance [25, 70]. Berg et. al [11] proposed the 'geometric blur' feature that is robust under affine distortions. Bosch et. al [16] proposed the Pyramidal Histogram of Oriented Gradients (PHOG) feature and the Pyramidal Histogram of Visual Words (PHOW) feature to represent local image shape and its spatial layout. Wu et.al [100] proposed a set of silhouette oriented features, called edgelet features, which were learned in a boosting framework to detect humans. Such approaches work well for detecting articulated/rigid objects, but encounter difficulties in recognizing manipulable objects due to the lack of discriminative power in these features. Todorovic et. al [92] model object categories as characteristic configurations of parts that are themselves simpler subcategories, allowing them to cope better with non-rigid objects. However, like all appearance based approaches, they still cannot deal with the many real-world objects that are similar in appearance but dissimilar in functionality. Functional properties of objects have also been used for object recognition. Functional capabilities of objects are derived from shape [85, 89], physics and motion [27]. These approaches are limited by the lack of generic models that can map static shape to function. There has been recent interest in using contextual information for object recognition. The performance of local recognition based approaches can be improved by modeling object-object [71, 42] and object-scene relationships [90, 72]. Torralba et. al used low level image cues [93] for providing context based on depth and viewpoint cues.

Hoiem et. al [47] presented a unified approach for simultaneous estimation of object locations and scene geometry. Rabinovich et. al [83] proposed incorporating semantic object context as a post-processing step to any object category recognition system using a conditional random field (CRF) framework.

There are a wide range of approaches to human action recognition [86, 67]. Analysing human dynamics from image sequences of actions is a common theme to many of these approaches [15, 102, 84, 91]. While human dynamics provides important clues for action recognition, they are not sufficient for recognition of activities which involve action on objects. Many human actions involve similar movements/dynamics, but due to their context sensitive nature have different meanings. Vaina et. al [96] suggested that action comprehension requires understanding the goal of an action. The properties necessary for achieving the goal were called Action Requirements and are related to the compatibility of an object with human movements such as grasps.

Compared to the large body of work carried out in human action recognition from video sequences, there has been little work on recognition from single images. Wang et. al [98] presented an approach for discovery of action classes from static images using the shape of humans described by shape context histograms. Jia-Li et. al [60] tackled a different, but related, problem of event recognition from static images. They presented an approach to combine scene categorization and object recognition for performing event classification such as badminton and tennis. The problem of action recognition from static images is one level lower in the action hierarchy and corresponds to "verb" recognition in the hierarchy suggested by Nagel et. al [73].

Attempts have been made before to model the contextual relationship between object and action recognition. Wilson et. al [99] introduced parametric Hidden Markov Model (PHMM) for human action recognition. They indirectly model the effect of object properties on human actions. Davis et. al [26] presented an approach to estimate the weight of a bag carried by a person using cues from the dynamics of a walking person. Moore et. al [68] conduct action recognition based on scene context derived from other objects in the scene. The scene context is also used to facilitate object recognition of new objects introduced in the scene. Kuniyoshi et. al [54] describe a neural network for recognition of "true" actions. The requirements for a "true" action included spatial and temporal relationships between object and movement patterns. Peursum et. al [81] studied the problem of object recognition based on interactions. Regions in an image were classified as belonging to a particular object based on the relative position of the region to the human skeleton and the class of action being performed. All of the above work models only one of the possible interactions between two perceptual elements. Either they try to model the dependence of object recognition on human actions or vice-versa. This assumes that one of the problems can be solved independent of the other and the information from one can be used to aid in recognition of the other.

4.2 Video Interpretation Framework

We first describe a computational model for interpretation of human-object interaction videos. We identify three classes of human movements involved in interactions with manipulable objects. These movements are (a) Reaching for an object of interest. (b) Grasping the object and (c) Manipulating the object. These movements are ordered in time. The reach movement is followed by grasping which precedes manipulation. In our model, we ignore the grasping motion since the hand movements are too subtle to be perceived at the resolution of typical video cameras when the whole body and context are imaged.

4.2.1 Overview

We present a graphical model for modeling human object interactions. The nodes in the model correspond to the perceptual analyses corresponding to the recognition of objects, reach motions, manipulation motions, object reactions. The edges in the graphical model represent the interactions/dependencies between different nodes.

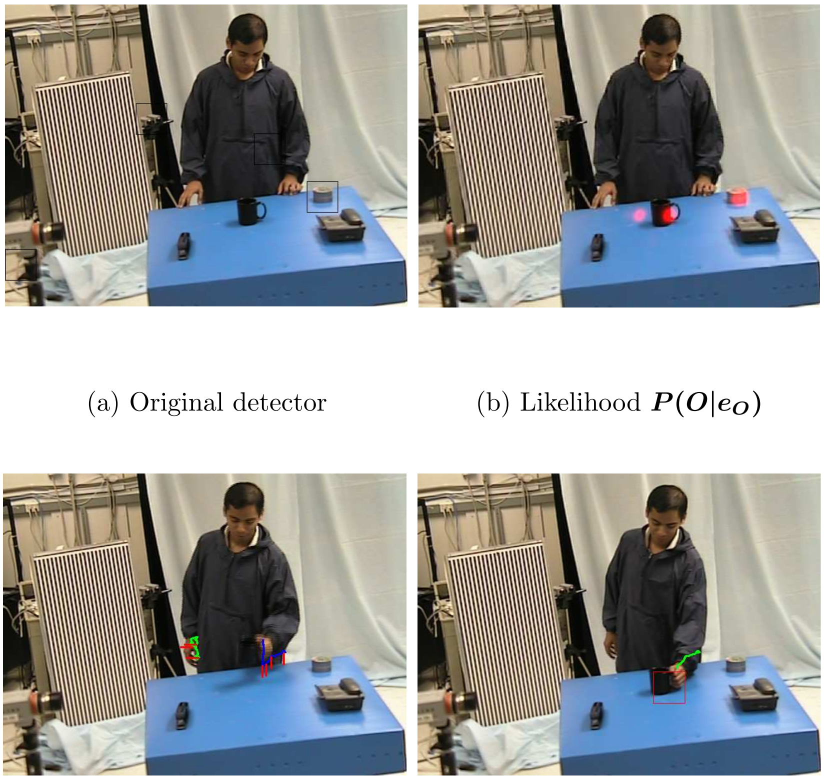
Reach movements enable object localization since there is a high probability of an object being present at the endpoint of a reach motion. Similarly, object recognition disables false positives in reach motion detection, since there should be an object present at the endpoint of a reach motion (See Figure 4.3). Reach motions also help to identify the possible segments of video corresponding to manipulation of the object, since manipulation motion is preceded by reach motion. Manipulation movements provide contextual information about the type of object being acted on and object class provides contextual information on possible interactions with them, depending on affordances and function. Therefore, a joint estimation of the two perceptual elements provides better estimates as compared to the case when the two are estimated independently(See Figure 4.4).

The object reaction to a human action, such as pouring liquid from a carafe into a cup or pressing a button that activates a device, provides contextual information about the object class and the manipulation motion. Our approach combines all these types of evidence into a single video interpretation framework. In the next section, we present a probabilistic model for describing the relationship between different elements in human object interactions.

4.2.2 Our Bayesian Model

Our goal is to simultaneously estimate object type, location, movement segments corresponding to reach movements, manipulation movements, type of manipulation movement and their effects on objects by taking advantage of the contextual information provided by each element to the others. We do this using the graphical model shown in Figure 4.5.

In the graphical model, objects are denoted by $\boldsymbol{O}$, reach motions by $\boldsymbol{M_r}$,



(a) Original detector

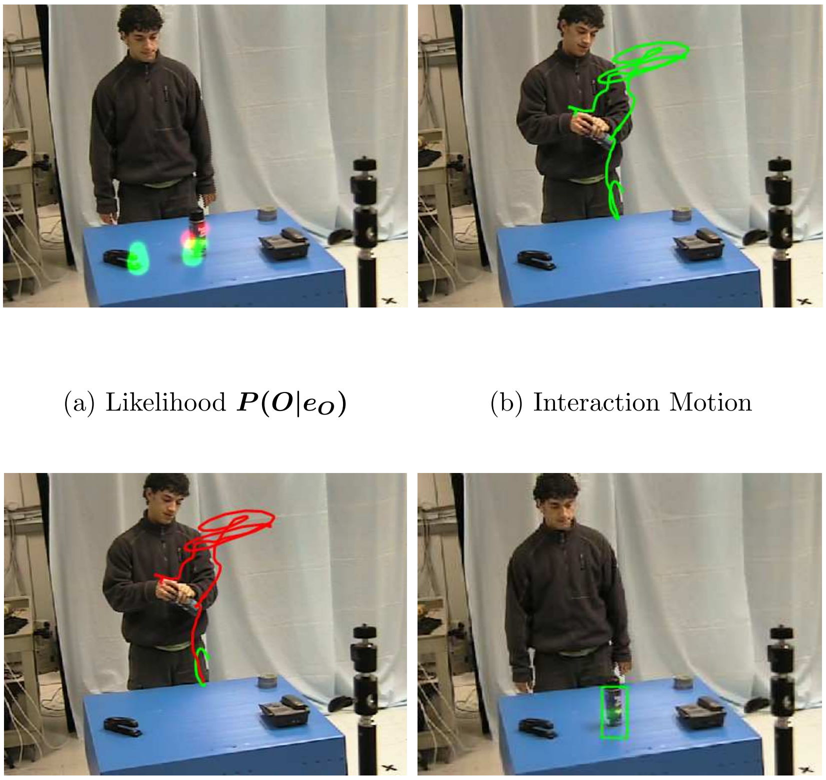
(b) Likelihood $\boldsymbol{P(O|e_O)}$

(c) Reach $\boldsymbol{P(M_r|e_r)}$

(d) $\boldsymbol{P(O, M_r|e_O, e_r)}$

Figure 4.3: Importance of contextual information involved in reach motions and object perception. (a) Object Detectors tend to miss some objects completely (b) Lowering the detection threshold can lead to false positives in detection. The likelihood of a pixel being the center of the cup is shown by intensity of red. (c) Reach Motion Segmentation also suffers from false positives. The trajectories are shown in green and blue with possible end points of reach motion shown in red. (d) Joint probability distribution reduces the false positives in reach motion and false negatives in object detection.

manipulation motions by $\boldsymbol{M_m}$ and object reactions by $\boldsymbol{O_r}$. The video evidence is represented by $\boldsymbol{e = \{e_O, e_r, e_m, e_{or}\}}$ where $\boldsymbol{e_O}$ represents object evidence, $\boldsymbol{e_r}$ and $\boldsymbol{e_m}$ represent reach and manipulation motion evidence and $\boldsymbol{e_{or}}$ represents object



(a) Likelihood $\boldsymbol{P(O|e_O)}$

(b) Interaction Motion

(c) Segmented Motion

(d) Belief: $\boldsymbol{Bel(O)}$

Figure 4.4: Importance of contextual information from interaction motion in object class resolution. In this experiment, object detectors for cups and spray were used. (a) The likelihood value of a pixel being the center of cup and spray bottle is shown by intensity of red and green respectively. (b) Hand trajectory for interaction motion (includes reach and manipulation). (c) The segmentation obtained. The green track shows the reach while the red track shows the manipulation.(d) Likelihood values after belief propagation. By using context from interaction with the object, it was inferred that since the object was subjected to a wave like motion, it is more likely a spray bottle.

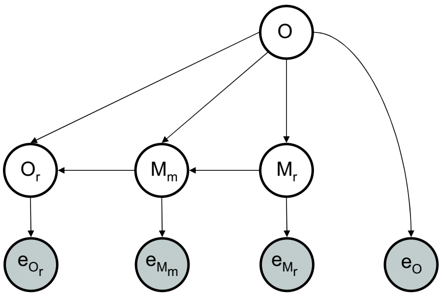


Figure 4.5: Underlying Graphical Model for Human Object Interaction. The observed and hidden nodes are shown in gray and white respectively.

reaction evidence. Using Bayes rule and conditional independence relations, the joint probability distribution can be decomposed as[3]:

$$\boldsymbol{P(O, M_r, M_m, O_r|e) \propto P(O|e_O)P(M_r|O)P(M_r|e_r)\ldots}$$
$$\boldsymbol{\ldots P(M_m|M_r, O)P(M_m|e_m)P(O_r|O, M_m)P(O_r|e_{or})}$$

We use loopy belief propagation algorithm for inference over the graphical model. In next few subsections we discuss how to compute each of these terms. Section 4.2.3 discusses how to compute the object likelihoods $\boldsymbol{P(O|e_O)}$. In section 4.2.4.1 we explain the computation of reach motion likelihood, $\boldsymbol{P(M_r|e_r)}$, and the contextual term $\boldsymbol{P(M_r|O)}$. This is followed by a discussion on computation of manipulation motion likelihood, $\boldsymbol{P(M_m|e_m)}$,and the term $\boldsymbol{P(M_m|M_r, O)}$ in sec-

[3] All the variables are assumed to be uniformly distributed and hence $\boldsymbol{P(O)}$, $\boldsymbol{P(M_r)}$, $\boldsymbol{P(M_m)}$, $\boldsymbol{P(O_r)}$, $\boldsymbol{P(e_O)}$, $\boldsymbol{P(e_r)}$, $\boldsymbol{P(e_m)}$ and $\boldsymbol{P(e_{or})}$ are constant

tion 4.2.4.2. In section 4.2.5, we discuss the object reaction likelihood $\boldsymbol{P(O_r|e_{or})}$ and the prior term, $\boldsymbol{P(O_r|O, M_m)}$.

4.2.3 Object Perception

The object node in the graphical model represents the random variable $\boldsymbol{O}$. We want to estimate the likelihood of the type of object and the location of the object. While our approach is independent of the likelihood model, we employ a variant of the histogram of oriented gradient(HOG) approach from [25, 104] [4]. Our implementation uses a cascade of adaboost classifiers in which the weak classifiers are Fischer Linear Discriminants. This is a window based detector; windows are rejected at each cascade level and a window which passes all levels is classified as a possible object location.

Based on the sum of votes from the weak classifiers, for each cascade level, $\boldsymbol{i}$, we compute the probability $\boldsymbol{P_i(w)}$ of a window, $\boldsymbol{w}$, containing the object. If a window were evaluated at all cascade levels, the probability of it containing an object would be $\boldsymbol{\prod_{i=1}^{L} P_i(w)}$. However, for computational efficiency many windows are rejected at each stage of the cascade [5]. The probability of such a window containing an object is computed based on the assumption that such windows would just exceed

[4] We use linear gradient voting with 9 orientation bins in 0-180; 12x12 pixel blocks of four 6x6 pixel cells.

[5] Our experiments indicate that in many cases locations rejected by a classifier in the cascade are true object locations and selected by our framework

the detection threshold of the remaining stages of the cascade. Therefore, we also compute a threshold probability($\boldsymbol{Pt_i}$) for each cascade level $\boldsymbol{i}$. This is the probability of that window containing an object whose adaboost score was at the rejection threshold. If a detector consists of $\boldsymbol{L}$ levels, but only the first $\boldsymbol{l_w}$ levels classify a window $\boldsymbol{w}$ as containing an object, then the overall likelihood is approximated by:

$$\boldsymbol{P(O = \{obj, w\}|e_O) \approx \prod_{i=1}^{l_w} P_i(w) \prod_{j=l_w+1}^{L} (Pt_j)} \tag{4.1}$$

4.2.4 Motion Analysis

We need to estimate the likelihoods of reach motion and manipulation motion. Our likelihood model is based on hand trajectories and therefore requires estimation of endpoints(hands in case of upperbody pose estimation) in each frame. While one can use independent models for tracking the two hands, this could lead to identity exchange and lost tracks during occlusions. Instead we pose the problem as upperbody pose estimation. We implemented a variant of [30] for estimating the 2D pose of the upper body. In our implementation, we use an edge [44] and silhouette based likelihood representation for body parts. We also use detection results of hands based on shape and appearance features and a temporal tracking framework where smoothness constraints are employed to provide priors. Figure 4.6 shows the results of the algorithm on few poses.

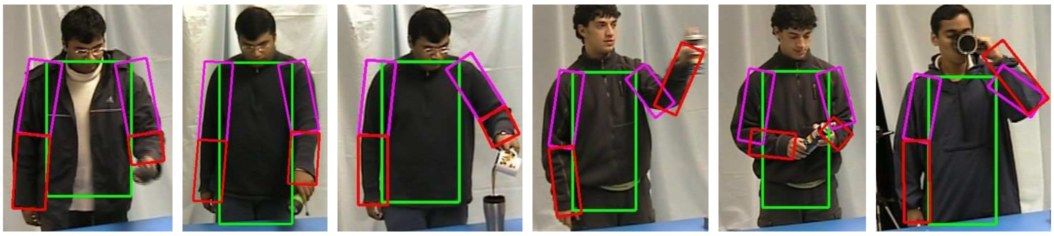


Figure 4.6: Results of Upper Body Pose Estimation Algorithm.

4.2.4.1 Reach Motion

The reach motion is described by three parameters: the start time $(\boldsymbol{t^r_s})$, the end time $(\boldsymbol{t^r_e})$ and the 2D image location being reached for $(\boldsymbol{l_r})$. We want to estimate the likelihood of reach motion $(\boldsymbol{M_r = (t^r_s, t^r_e, l_r)})$ given the hand trajectories. An approach for detecting reach motion was presented in [82]. It is based on psychological studies which indicate that the hand movements corresponding to ballistic motion such as reach/strike have distinct 'bell' shaped velocity profiles [64, 87](See Figure 4.7). There is an initial impulse accelerating the hand/foot towards the target, followed by a decelerating impulse to stop the movement. There is no mid-course correction. Using features such as time to accelerate, peak velocity and magnitude of acceleration and deceleration, the likelihoods of reach movements can be computed from hand trajectories.

However, there are many false positives because of errors in measuring hand trajectories. These false positives are removed using contextual information from object location. In the case of point mass objects, the distance between object location and the location being reached for should be zero. For a rigid body, the

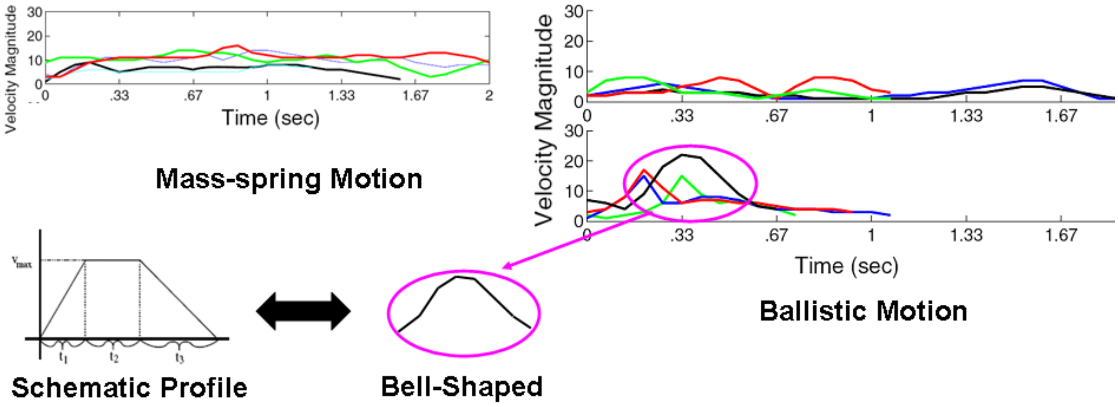


Figure 4.7: Plot on the left shows velocity profiles of some mass-spring motions and the figure on the right shows some ballistic hand movements. The velocity remains low and constant during mass-spring movements. It reduces to zero only at the end of the movement. On the other hand, hand movements corresponding to ballistic motion such as reach/strike have distinct 'bell' shapes.

distance from the center of the object depends on the grasp location. We represent $\boldsymbol{P(M_r|O)}$ using a normal function, $\boldsymbol{\mathcal{N}(|l_r l_o|, \mu, \sigma)}$, where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are the average distance and variance of the distances in a training database between grasp locations and object centers.

4.2.4.2 Manipulation Motion

Manipulation motions also involve three parameters: start time ($\boldsymbol{t_s^m}$), end time ($\boldsymbol{t_e^m}$) and the type of manipulation motion/action ($\boldsymbol{T_m}$) (such as answering a phone, drinking etc). We need to compute $\boldsymbol{P(M_m|e_m)}$, the likelihood of a manipulation given the evidence from hand trajectories. While one can use any gesture recognition approaches based on hand trajectories to estimate the likelihood, we use a simple discrete HMM based approach to estimate it.

We need to first compute a discrete representation of the manipulation motion. Towards this end, we obtain a temporal segmentation of the trajectory based on a limb propulsion model. An approach for such a segmentation was presented in [82]. There are two models for limb propulsion in human movements: ballistic and mass-spring models [87]. Ballistic movements, discussed previously, involve impulsive propulsion of the limbs (acceleration towards the target followed by deceleration to stop the movement). In the mass-spring model, the limb is modeled as a mass connected to a springs. Therefore, the force is applied over a period of time.

To obtain the temporal segmentation of a velocity profile, it is observed that the endpoints of each ballistic segment corresponding to a local minima in the velocity profile. However, due to noise all local minimas are not the endpoints of atomic segments. Therefore, the segmentation problem is treated as that of classifying the points of local minima as being segmentation boundaries or not. The classification is based on features such as accelerating impulse and its duration. Given confidence values for each time instant to be a starting, ending or negligible movement, we compute the most likely segmentation of the velocity profile using Maximum Likelihood(See Figure 4.8).

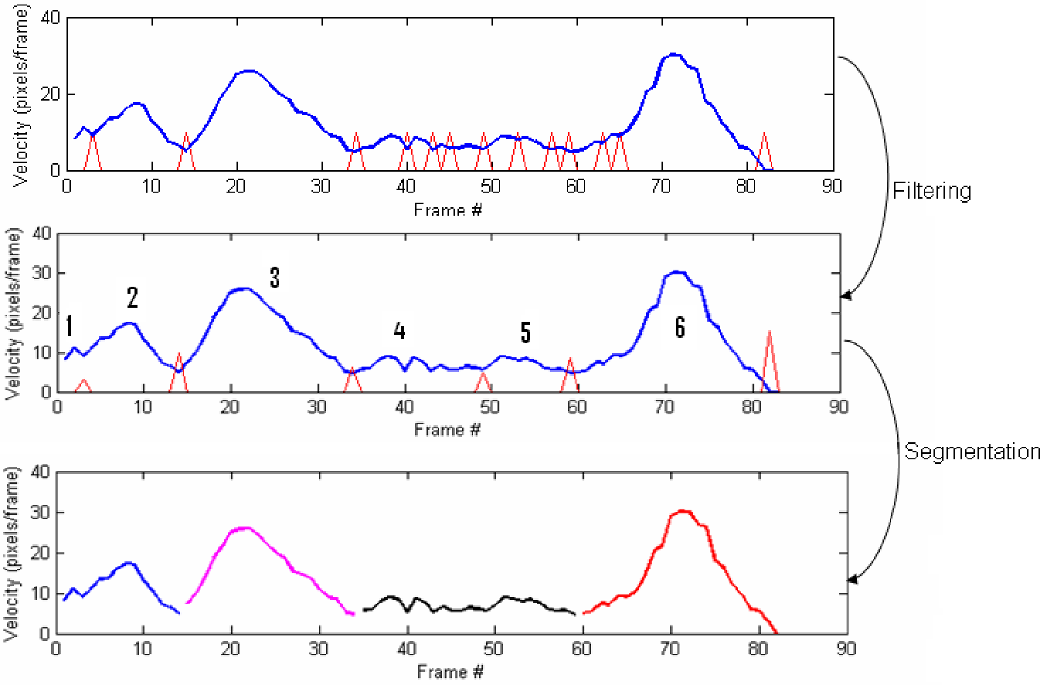


Figure 4.8: Segmentation Procedure: The first graph shows the local minima of velocity profile. These local minima are classified into possible endpoints of each segment. This is followed by a maximum likelihood approach to obtain the segmentation of the velocity profile.

Each segment is then replaced by a discrete alphabet defined as the cross-product of type of propulsion(ballistic/mass-spring) and the hand locations at the end of the motion segments, represented with respect to the face. By using alphabets for atomic segments we transform a continuous observation into a discrete symbol sequence. This is used as input to obtain the likelihoods of different types of manipulation motion from their corresponding HMM's.

In addition to computing the likelihood, we need to compute the term $\boldsymbol{P(M_m|M_r, O)}$. Manipulation motion is defined as a 3-tuple, $\boldsymbol{M_m = (t_s^m, t_e^m, T_m)}$. The starting and ending times, $\boldsymbol{t_s^m}$ and $\boldsymbol{t_e^m}$, depend on $\boldsymbol{M_r}$ but are independent of $\boldsymbol{O}$. Similarly, the type of manipulation motion, $\boldsymbol{T_m}$, depends on $\boldsymbol{O}$ but is independent of $\boldsymbol{M_r}$[6]. Hence, we decompose the prior term as:

$$\boldsymbol{P(M_m|M_r, O) = P(t_s^m, t_e^m|M_r)P(T_m|O)} \tag{4.2}$$

Assuming grasping takes negligible time, the time difference between the ending time of a reach motion and the starting time of a manipulation motion should be zero. We model $\boldsymbol{P(t_s^m, t_e^m|M_r)}$ as a normal distribution $\boldsymbol{\mathcal{N}(t_s^m - t_e^r, 0, \sigma^t)}$ where $\boldsymbol{\sigma^t}$ is the observed variance in the training dataset. $\boldsymbol{P(T_m = mtype|O = obj)}$ is computed based on the number of occurrences of manipulation $\boldsymbol{mtype}$ on object $\boldsymbol{obj}$ in our training dataset.

[6] Type of manipulation also depends upon the direction of reach motion. This factor is, however, ignored in this paper

4.2.5 Object Reactions

Object reaction is defined as the effect of manipulation on the object. In many cases, manipulation movements might be too subtle to observe using computer vision approaches. For example, in the case of a flashlight, the manipulation involved is pressing a button. While the manipulation motion is hard to detect, the effect of such manipulation (the lighting of the flashlight) is easy to detect. Similarly, the observation of object reaction can provide context on object properties. For example, the observation of the effect of pouring can help making the decision of whether a cup was empty or not.

The parameters involved in object reaction are the time of reaction ($\boldsymbol{t_{react}}$) and the type of reaction ($\boldsymbol{T_{or}}$). However, measuring object reaction type is difficult. Mann et. al [62] presented an approach for understanding observations of interacting objects using Newtonian mechanics. This approach can only be used to explain rigid body motions. Apart from rigid body interactions, the interactions which lead to changes in appearances using other forces such as electrical are also of interest to us.

We use the differences of appearance histograms (8 bins each in RGB space) around the hand location as a simple representation for reaction type classification. Such a representation is useful in recognizing reactions in which the appearance of the object at the time of reaction, $\boldsymbol{t_{react}}$, would be different than appearance at the start or the end of the interaction. Therefore, the two appearance histograms are

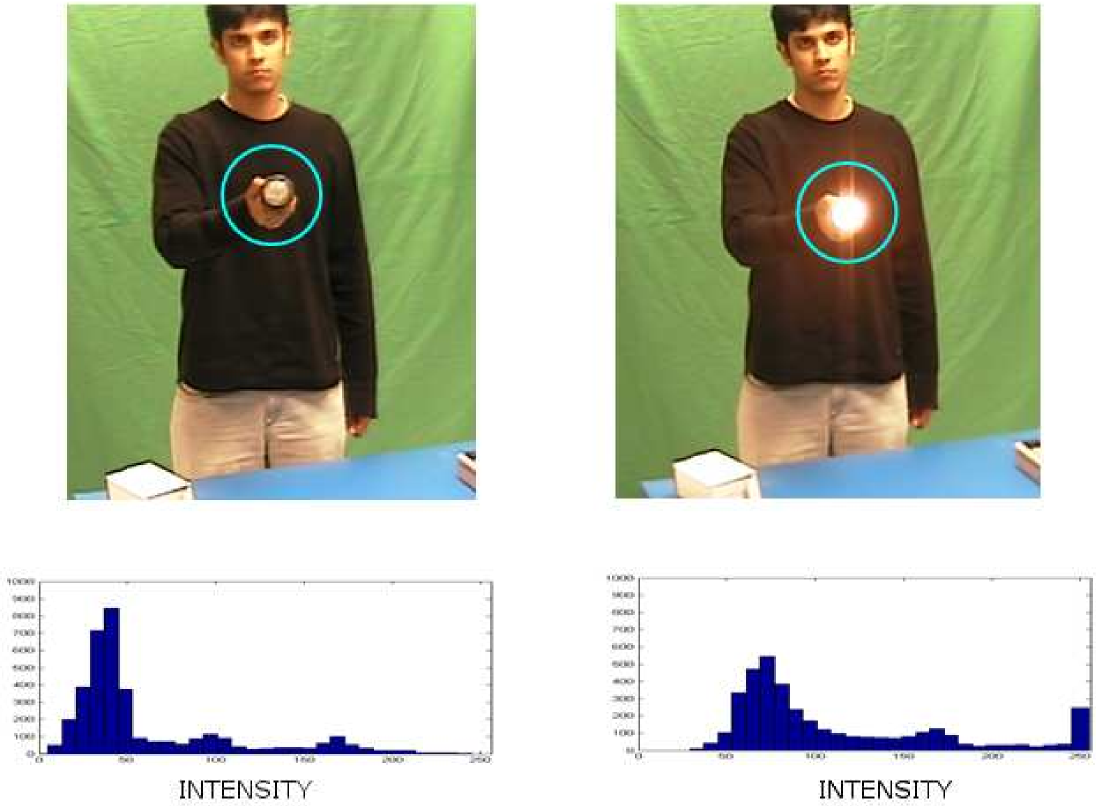


Figure 4.9: Using appearance histograms around hand to estimate $\boldsymbol{P(O_r|e_{or})}$. In the case above, illumination change due to flashlight causes the change in intensity histogram.

subtracted and compared with the difference histograms in the training database to infer the likelihood of the type of reaction($\boldsymbol{T_{or}}$).

In addition, we need to compute the priors $\boldsymbol{P(O_r|M_m, O)}$. Object reaction is defined by a 2-tuple, $\boldsymbol{O_r = (T_{or}, t_{react})}$. Using the independence of the two variables:

$$P(O_r|M_m,O) = P(T_{or}|M_m,O)P(t_{react}|M_m,O) \tag{4.3}$$

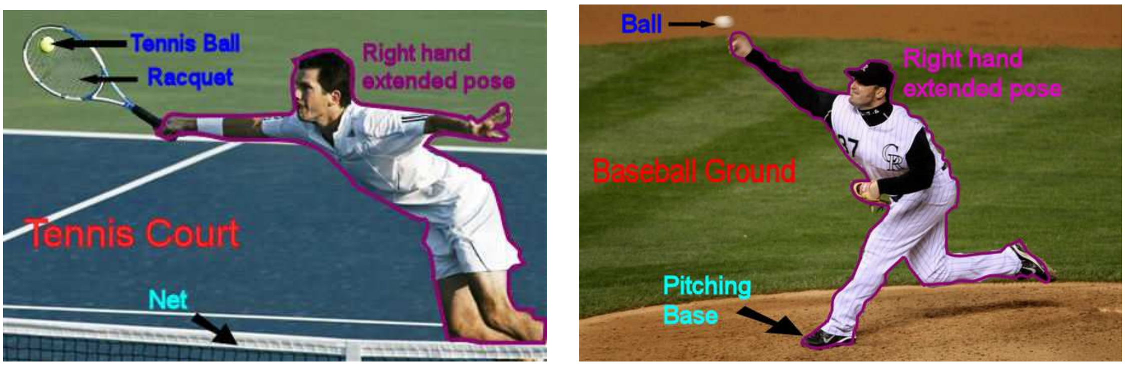
The first term can be computed by counting the occurrences of T_{or} when the manipulation motion is of type $mtype$ and the object is of type obj. For modeling the second term, we observed that the reaction time ratio, $r_r = \frac{t_{react}-t_s^m}{(t_e^m-t_s^m)}$, is generally constant for a combination of object and manipulation. Hence, we model the prior by a normal function $\mathcal{N}(r_r, \mu_r, \sigma_r)$ over the reaction-time ratio, where μ_r and σ_r are the mean and variance of reaction-time ratios in the training dataset.

4.3 Recognizing Interactions from Static Images

While action recognition requires motion information, in the case of static images, contextual information can be used in conjunction with human pose to infer action. Figures 4.10 (a) and (b) show examples of reasoning involved in inference of actions from a static image. In both cases, pose alone does not provide sufficient information for identifying the action. However, when considered in the context of the scene and the objects being manipulated, the pose become informative of the goals and the action.

Relevant objects in the scene generally bear both a semantic[7] and spatial

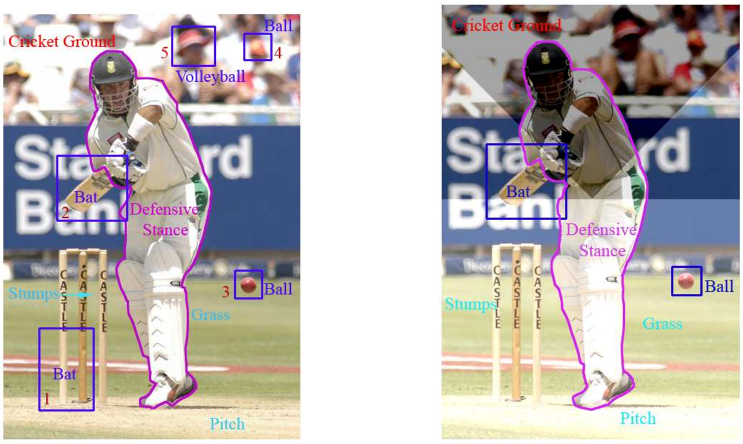
[7] By semantic relationships we refer to those relationships that are captured by co-occurrence statistics



(a)Tennis Forehand (b)Baseball Pitching

Figure 4.10: Examples depicting the reasoning process in action inference from static images. The labels in red are the result of a scene categorization process, cyan labels and blue labels represent scene and manipulable objects, respectively, and the magenta label is the result of a pose estimation algorithm. For understanding actions from static images, information is combined from all components. While the pose is similar in both scenes, the presence of the racket and tennis ball, along with the tennis court environment suggests that the first picture is a 'tennis-forehand' while the second is baseball pitching due to the presence of the pitching area and the baseball field.

relationship with humans and their poses. For example, in a defensive stance of a cricket batsman, the bat is facing down and is generally below or level with the person's centroid. Similarly, the location of the cricket ball is also constrained by the person's location and pose(See Figure 4.11). We describe how to apply spatial constraints on locations of objects in the action recognition framework. By combining action recognition from poses with object detection and scene analysis we also improve the performance of standard object detection algorithms.



(a)Without Spatial Constraints (b)With Spatial Constraints

Figure 4.11: Detection of manipulable objects can be improved using spatial constraints from human action. The ball detector detects two possible cricket balls. In the case of defensive batting, the probability of possible locations of the ball is shown by the shaded regions. Hence the region below the centroid, where the ball is more likely to be present, is brighter. The ball denoted in box 4 lies in a darker region, indicating it is less likely to be a cricket ball due to its location with respect to the human. For objects such as bats, another important spatial constraint is connectedness. A segment of the bat should be connected to a segment of the human; therefore false positives, such as object 1, can be rejected.

We first present an overview of the approach in section 4.3.1. Section 4.3.2 describes our Bayesian model for recognition of actions and objects in static images. This is followed by a description of individual likelihood models and interactions between different perceptual elements in subsequent sections.

4.3.1 Overview

Studies on human object perception suggest that people divide objects into two broad categories: scene and manipulable objects. These objects differ in the way inferences are made about them. Chao et. al [23] showed that when humans see manipulable objects, there is cortical activity in the region that corresponds to action execution. Such responses are absent when scene objects, such as grass and house, are observed. Motivated by such studies, we treat the two classes differently in terms of the role they play in inferring human location and pose and represent them by two different types of nodes in the Bayesian model.

Our Bayesian model consists of four types of nodes, corresponding to scene/event, scene objects, manipulable objects and human. The scene node corresponds to the place where the action is being performed, such as a cricket ground or a tennis court. The scene object nodes correspond to objects which do not have causal dependency on the human actor and are mostly fixed in the scene, such as the net in the tennis court. Manipulable objects correspond to the instruments of the game such as a ball or a racket.

The interactions between these nodes are based on semantic and spatial constraints. The type of objects that occur in an image depends on the scene in which the action takes place. For example, it is more likely for a pitch to occur in a cricket ground than a tennis court. Therefore, there exist semantic relationships between scene and scene objects.

The type of action corresponding to a pose depends on the type of scene and the scene objects present. The type of action also depends on the location of the human with respect to the scene objects. For example, a pose with one hand up in a tennis court can either be a serve or a smash. However, if the human is located at the baseline it will more likely be a serve; otherwise, if he is near the net it will more likely be a smash. While considering such spatial relationships is important, in this paper we consider only the semantic relationships between actions and the scene and scene objects. Since we are not modeling spatial relationships between scene objects and human actions, we only consider the presence/absence of scene objects. Therefore, each scene object node (representing a class such as cricket-stumps) is characterized by a binary variable indicating the presence/absence of that scene object class.

For manipulable objects, there exists both spatial and semantic constraints between people and the objects. The type of manipulable objects in the image depends on the type of action being performed. Also, the location of the manipulable objects is constrained by the location of the human, the type of action and the types of manipulable objects. For example, the location of a tennis ball is constrained by the type of action (in the case of a forehand the ball is located to the side of a person while in the case of a serve it appears above). Spatial constraints also depend on the type of object; objects such as a tennis racket should be connected to the person while objects such as a ball generally have no such connectivity relationships. We describe an approach to represent such relationships in our Bayesian network.

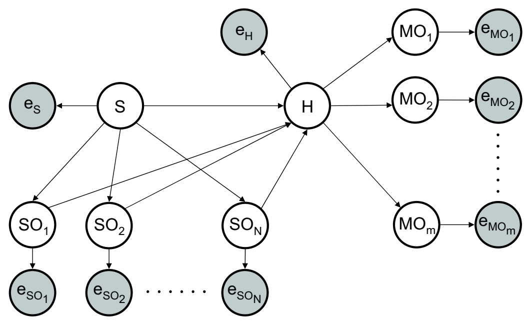


Figure 4.12: Graphical Model. The observed and hidden nodes are shown in gray and white respectively.

4.3.2 Our Bayesian Model

The graphical model used for the scene interpretation framework is shown in figure 4.12. We simultaneously estimate the scene type, scene objects, human action and manipulable object probabilities. Let $\boldsymbol{S}$ represent the scene variable, $\boldsymbol{SO_1...SO_N}$ represent the $\boldsymbol{N}$ type of scene objects, $\boldsymbol{H}$ represent the human and $\boldsymbol{MO_1..MO_M}$ represent the $\boldsymbol{M}$ possible manipulable objects. If $\boldsymbol{e} = \{\boldsymbol{e_S, e_{SO_1}..e_{SO_N}, e_H, e_{MO_1}..e_{MO_N}}\}$ represents the evidential variables, our goal is to estimate $\boldsymbol{P(S, H, SO_1..SO_N, MO_1..MO_M|e)}$. This can be decomposed as:

$$\begin{aligned}\textstyle\prod_j \boldsymbol{P(MO_j|H)P(MO_j|e_{MO_j})P(H|S, SO_1..SO_N)P(H|e_H)\dots}\\ \boldsymbol{\dots}\textstyle\prod_i \boldsymbol{P(SO_i|S)P(SO_i|e_{SO_i})P(S|e_S)}\end{aligned} \tag{4.4}$$

We use loopy belief propagation algorithm for inference over the graphical model.

4.3.3 Scene Perception

A scene is mainly characterized as a place in which we can move [78]. In this paper, the scene corresponds to the place where an action is being performed such as tennis court and croquet field. Each image is associated with a probability of belonging to one of the scene classes. Several experimental studies have shown that when humans view a scene, they extract functional and categorical information from the scene; whereas they tend to ignore information regarding specific objects and their locations. In accordance, Oliva et. al [78] bypass the segmentation and processing of individual objects in their scene classification framework. Rather than looking at a scene as a configuration of objects, they propose to consider a scene like an individual object, with a unitary shape. They show that scenes belonging to the same category share a similar and stable spatial structure that can be extracted at once, without segmenting the image. A set of holistic spatial properties of the scene, together referred to as a Spatial Envelope, are used, which include naturalness, openness, roughness, ruggedness and expansion. We use their approach to compute the concatenated feature vector for every image in the dataset. Using the training feature vectors we train a Support Vector Machine (SVM) for the classification task. For a test image, the SVM returns a score $\boldsymbol{d_S}$ which represents the distance of the test point from the separating hyperplane. Based on this distance, we estimate the probability $\boldsymbol{P(S|e_S)}$ as:

$$\boldsymbol{P(S|e_S) = \frac{1}{Z_{Scene}} exp(-\alpha_{Scene} d_S)} \tag{4.5}$$

where $\boldsymbol{\alpha_{Scene}}$ is the scaling parameter and $\boldsymbol{Z_{scene}}$ is the normalization factor.

4.3.4 Scene Objects

Each scene object node corresponds to a class of scene objects and is represented by the probability of presence of that object class across the image. We uniformly sample points across the image and extract a patch around each point (For experiments, grid points are sampled at $\boldsymbol{25}$ pixels each in x,y direction and the patch size of $\boldsymbol{50 \times 50}$ is used). We classify each patch as belonging to one of the $\boldsymbol{N}$ scene object classes, using an adaboost based classifier [97] based on features such as histogram of oriented gradients (HOG), histograms of each color channel(8 bins each in color channel), and histograms of edge distance map values within the neighborhood. We compute $\boldsymbol{P(SO_i|S)}$ based on the conditional probability tables learned using the co-occurrence relationships in the training dataset.

4.3.5 Human in Action

Every detected person in the image is characterized by the action ($\boldsymbol{A}$) he is performing, and location given by a bounding box ($\boldsymbol{l^H}$). For action classification, we detect humans and employ the pose information. A similar approach has been proposed in a recent paper [32]. In our experiments, we detect humans using an approach similar to [101]. Since the observed image shape of a human changes significantly with articulation, viewpoint and illumination, it is infeasible to train a single human detector for all shapes. Instead, we first cluster the observed shapes from our training data, and train multiple human detectors, one for each shape cluster. Our human detectors closely match those proposed by [25]. Given a bounding box around a detected human, we segment the human using *GrabCut* [12], an efficient tool for foreground segmentation. Once we have a possible human segmentation, we extract shape context features (5 radial bins and 12 orientation bins) from the silhouette of the human. We then cluster shape context features [1] from the training database to build a dictionary of "shape context words". A detected human in an image is then characterized by the histogram of shape context words. The number of words/clusters determines the dimensionality of our pose feature vector. We then use the K-Nearest Neighbor approach for classification, providing $\boldsymbol{P(H|e_H)}$. Given a test sample, we determine the K nearest neighbors in the training data. Each of the K neighbors votes for the class it belongs to with a weight based on its distance from the test sample. The final scores obtained for each class determine the likelihoods for each pose category, $\boldsymbol{P(H|e_H)}$. For the experiments used in the paper we use $\boldsymbol{K = 5}$.

We also need to compute $\boldsymbol{P(H|S, SO_1..SO_N)}$. Assuming conditional independence between scene object categories given human action, we rewrite as:

$$\boldsymbol{P(H|S, SO_1..SO_N) = \prod_{i}^{N} P(H|S, SO_i)} \tag{4.6}$$

Each of these can be computed using co-occurrence statistics of human action-scene-scene object combinations, independently for every scene object class.

4.3.6 Manipulable Objects

Each detected manipulable object in the image has the following attributes: an associated class id ($\boldsymbol{c_i^m}$) and location parameters given by a bounding box ($\boldsymbol{l_i^m}$) around the object. We use the object detector described in section 4.2.3. Using this approach, however, we are unable to distinguish between objects that have the same shape but a different dominant color; for example a cricket ball (often red or white in color) as opposed to a tennis ball (often yellow in color). Thus, we build appearance models of manipulable objects using non-parametric Kernel Density Estimation (KDE) to also perform an appearance based classification. We sample pixels from training images of the manipulable objects and build a 3D model in the RGB space.

$$\boldsymbol{p_{Model}(r,g,b) = \frac{1}{N}\sum_{i=1}^{N} K_{\sigma_r}(r-r_i)K_{\sigma_g}(g-g_i)K_{\sigma_b}(b-b_i)} \tag{4.7}$$

Given a test image, we first use the shape based classifier to detect potential object candidates. Within each candidate window, we sample pixels and build a density estimate using (KDE). This test density is compared to the color model of every object category using the Kullback-Leibler distance. This provides the final manipulable object detection probabilities based on appearance given by $\boldsymbol{P(MO_i|e^{ap}_{MO_i})}$. Therefore the probability $\boldsymbol{P(MO_i|e_{MO_i})}$ is given by:

$$\boldsymbol{P(MO_i|e_{MO_i}) = P(MO_i|e^{sh}_{MO_i})P(MO_i|e^{ap}_{MO_i})} \tag{4.8}$$

where $\boldsymbol{e^{sh}}$ refers to shape and $\boldsymbol{e^{ap}}$ refers to appearance evidence. We also need to compute $\boldsymbol{P(MO_i|H)}$. Human actions and locations provide both semantic and spatial constraints on manipulable objects. The spatial constraints given human locations are with respect to the type of manipulable object and type of action being performed. We model two kinds of spatial constraints: (a) Connectivity - Certain manipulable objects like a tennis racket or a cricket bat should be connected to the human in action. (b) Positional and Directional Constraints: These location constraints are evaluated with respect to the centroid of the human that is acting on them. The conditional probability densities are based on the type of action being performed. For example, given a tennis serve action it is more likely that the ball is above the player, while if the action is forehand it is more likely to the side of the player. We model positional relations in terms of the displacement vector of the object centroid from the centroid of the human body. Thus we obtain:

$$\boldsymbol{P(MO_i = (c^m_i, l^m_i)|H = (A, l^H)) = P(l^m_i|c^m_i, A, l^H)P(c^m_i|A)} \tag{4.9}$$

The first term refers to the spatial constraints and can be learned by discretizing the space around the human as shown in figure 4.13. From the training images, we learn the condition probability tables of the region in which the manipulable object lies given the type of manipulable object and the type of action. The second term is the semantic constraint and is modeled from co-occurrence statistics of human action-manipulable objects combinations from training data.

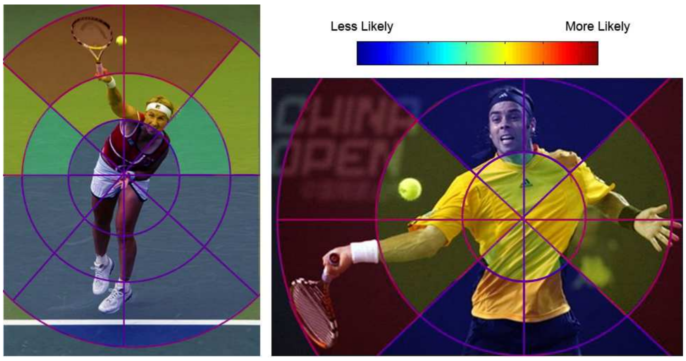


Figure 4.13: Spatial Constraints between locations of manipulable objects and humans for different poses. In an idealized scenario, for a forehand pose, the ball is more likely to be seen on the side; for a tennis serve, it is more likely to be seen above the human. We use **2** radial bins and **8** orientation bins to specify position of manipulable object with respect to the human body.

4.4 Experimental Evaluation

4.4.1 Video Interpretation

We evaluated our video interpretation framework on test dataset [8] of 10 subjects performing 6 possible interactions with 4 different objects. The objects in the test-dataset included cup, spray bottle, phone and flashlight. The interactions with these objects were: drinking from a cup, spraying from a spray bottle, answering a phone call, making a phone call, pouring from a cup and lighting the flashlight.

Training: We used a fully-supervised approach for training the Bayesian model for video interpretation. Training of the model requires training of a HOG based detector for all object classes and HMM models for all classes of interactions. Training for HOG based object detector was done using images from training datasets obtained using Google image search(50 images for each object, negative images were used from INRIA and CALTECH datasets). HMM models were trained using a separate training dataset of videos. The object reactions are learned using the supervised training scheme. In training videos, the frames for the object reaction were manually segmented and the appearance histograms around the hand were used to learn the appearance of object reaction. Additionally our model requires co-occurrence statistics of object-interaction-reaction combinations, distance between grasp location and object center, and reaction time ratios. We used a

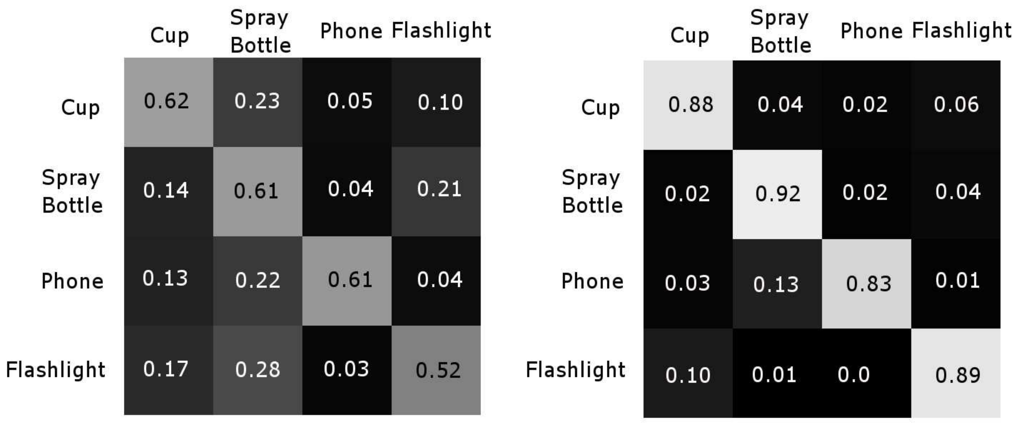
[8] The datasets used in all the experiments are available online and can be downloaded from http://www.umiacs.umd.edu/∼agupta

training dataset of 30 videos of 5 actors performing different types of manipulations on the objects. Training was done in fully supervised manner. All the videos were manually labeled with object locations, hand locations and the type of objects, manipulation and object reactions.

Object Classification: Among the objects used, it is hard to discriminate the spray bottle, flashlight and cup because all three are cylindrical (See Figures 4.16(a),(b)). Furthermore, the spray bottle detector also fired for the handset of the cordless phone (See Figure 4.16(d)). Our approach was also able to detect and classify objects of interest even in cluttered scenes (See Figure 4.16(c)). Figures 4.14(a) and 4.14(b) shows the likelihood confusion matrix for both the original object detector and the object detector in the human-object interaction framework. Using interaction context, the recognition rate of objects at the end of reach locations improved from **78.33%** to **96.67%**[9].

Action Recognition: Of the six activities, it is very hard to discriminate between pouring and lighting on the basis of hand trajectories(See Figure 4.16(a) and (b)). While differentiating drinking from phone answering should be easy due to the differences in endpoint locations, there was still substantial confusion between the two due to errors in computation of hand trajectories. Figure 4.15(a) shows the likelihoods of actions that were obtained for all the videos using hand-dynamics alone. Figure 4.15(b) shows the confusion matrix when action recognition was con-

[9] The recognition rate depicts the correct classification of localized object into one of the five classes: background, cup, spray-bottle, phone and flashlight



	Cup	Spray Bottle	Phone	Flashlight
Cup	0.62	0.23	0.05	0.10
Spray Bottle	0.14	0.61	0.04	0.21
Phone	0.13	0.22	0.61	0.04
Flashlight	0.17	0.28	0.03	0.52

(a) HOG Detector

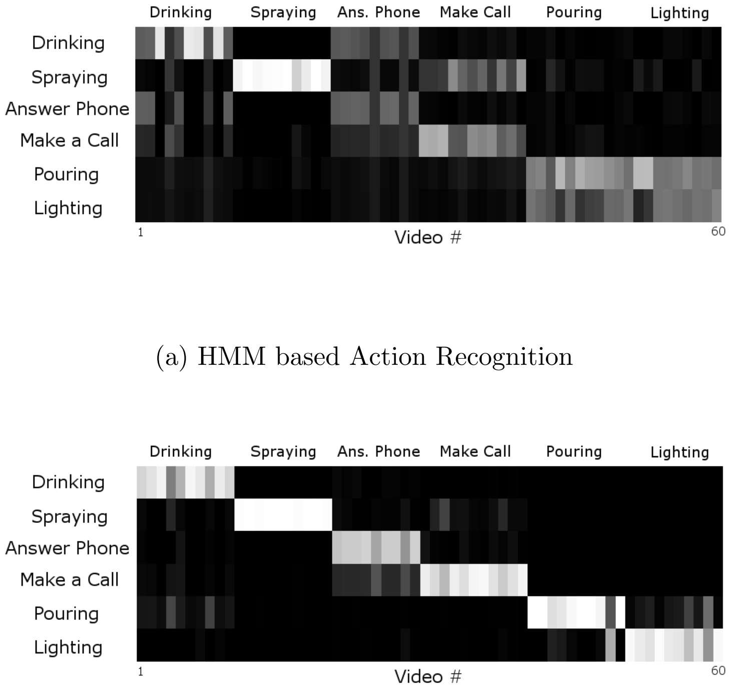
	Cup	Spray Bottle	Phone	Flashlight
Cup	0.88	0.04	0.02	0.06
Spray Bottle	0.02	0.92	0.02	0.04
Phone	0.03	0.13	0.83	0.01
Flashlight	0.10	0.01	0.0	0.89

(b) Using Whole Framework

Figure 4.14: Object Likelihood Confusion Matrix: The i^{th} row depicts the expected likelihood values when i^{th} type of object is present. The right table shows the results of our whole framework, taking into account action, object reaction, and reach motion.

ducted using our framework. The overall recognition rate increased from **76.67%** to **93.34%** when action was recognized using the contextual information from objects and object reactions. While the trajectories might be similar in many cases, the context from object provided cues to differentiate between confusing actions. Similarly in the cases of lighting and pouring, contextual cues from object reaction helped in differentiating between those two actions.

Segmentation Errors: Apart from errors in classification, we also evaluated our framework with respect to segmentation of reach and manipulation motion. The segmentation error was the difference between the actual frame number and



(a) HMM based Action Recognition

(b) HMM based recognition in Interaction Context

Figure 4.15: Comparison of Action Likelihoods without and with contextual information. Each Column represents the normalized likelihood values for six possible actions.

the computed frame number for the end of a reach motion. We obtained the ground truth for the data using manual labellings. Figure 4.17 shows the histogram of segmentation errors in the videos of the test dataset. It can be seen that **90%** of detections were within 3 frames of actual end-frames of reach motion. The average length of the video sequence was approximately 110 frames.

4.4.2 Image Interpretation

Dataset: We evaluated our approach on a dataset which had 6 possible actions: "tennis-forehand", "tennis-serve", "volleyball-smash", "cricket-defensive shot", "cricket-bowling" and "croquet-shot". The images for the first 5 classes were downloaded from the internet and for the sixth class, we used a publicly available dataset [60]. A few images from the dataset are shown in figure 4.19. The classes were selected so that they had significant confusion due to scene and pose. For example, the poses during "volleyball-smash" and "tennis-serve" are quite similar and the scenes in "tennis-forehand" and "tennis-serve" are exactly the same.

Training: We used a fully-supervised approach for training the Bayesian model for image interpretation. We have to learn the parameters for individual likelihood functions and parameters of the conditional probabilities which model the interactions between different perceptual analyses. To learn parameters of individual likelihood functions, we trained individual detectors separately using training images from Google image search(50 images each for every object and 30 silhouettes each for the pose likelihood). Learning parameters corresponding to conditional probabilities requires a separate training dataset of images. Our training dataset consisted of 180 images (30 from each class).

Evaluation: We tested the performance of our algorithm on a dataset of 120 test images (20 from each class). We compared the performance of our algorithm with the performance of models based on isolated components. Figure 4.20 shows the confusion matrix obtained using the full model described in the paper. We also show some failure cases in the figure. Our approach gave some mis-classifications when the scene involved is the same but actions are different such as bowling being classified as batting. This occurs whenever the pose classification algorithm gives a wrong action likelihood (mostly due to faulty segmentation by Grabcut) and the manipulable object detector fails to find any discriminating manipulable object.

Figure 4.21(a) shows the performance of a pose based classification algorithm. We used the pose component of our model to obtain the confusion matrix. As expected, the performance of pose-only model is very low due to similar poses being shared by different actions. For example, there is high confusion between "tennis-serve" and "bowling", since both actions share a high arm pose. Similarly we see confusion between "bowling" and "volleyball". The confusion between "volleyball smash" and "tennis forehand" is mainly due to incorrect segmentations by grabcut.

The comparison between overall performance of our approach and the individual components is shown in figure 4.21(b). The performance of our approach was **78.86%** as compared to **57.5%** by the pose-only model and **65.83%** by the scene-only model.

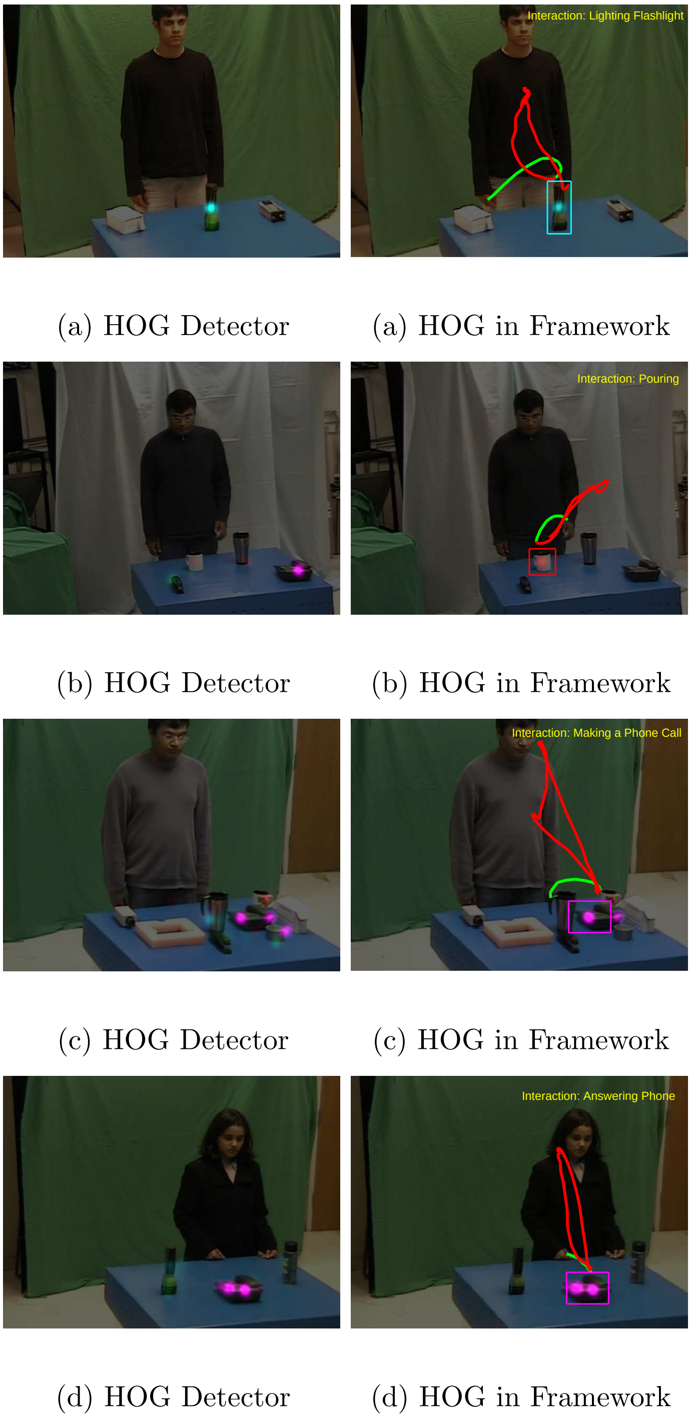
Figures 4.22 and 4.23 shows some examples of correct classification by our algorithm. In both cases, our approach rejects false positives because the belief in the objects fall below the detection threshold when combined with other elements like pose and scene information. For example, in Fig 4.22 the false positives of bats are rejected as they fail to satisfy spatial constraints. Also, in both cases detections related to objects incongruent with scene and action information are also rejected.

Influence of Parameters: We evaluated our system with respect to the parameters of each component of our system. We varied the parameter $\boldsymbol{\alpha_{Scene}}$ used to obtain the scene classification probabilities(Section 4.3.3). Fig 4.24(a) shows that action recognition accuracy increases with increasing $\boldsymbol{\alpha_{Scene}}$, but flattens out after a value of 5. The discriminative power of the scene component lowers with decreasing $\boldsymbol{\alpha_{Scene}}$ and therefore we observe a lower system performance. In our experiments, we use $\boldsymbol{\alpha_{Scene} = 5}$.

Oliva et al. [78] use the WDST (Windowed Discriminant Spectral Template) which describes how the spectral components at different spatial locations contribute to a spatial envelope property, and sample it at regular intervals to obtain a discrete representation. One of the components of their method, $\boldsymbol{w_{Scene}}$, determines the coarseness of this sampling interval. We varied the coarseness of the sampling where smaller $\boldsymbol{w_{Scene}}$ refers to coarser sampling. Figure 4.24(b) shows our performance accuracy with respect to $\boldsymbol{w_{Scene}}$. Our action recognition accuracy reduces for a very coarse sampling of the WDST, but is stable at finer scales. We use $\boldsymbol{w_{Scene} = 4}$ for the experiments.

Our object detection module detects multiple objects in the scene and passes the top few detections on to the bayesian framework. We evaluated our system accuracy with regards to the number of manipulable object detections passed to the bayesian framework. For lower number of detections, the bayesian framework has lower performance due to missing true detections. For higher number of detections, the bayesian framework has lower performance due to the confusion from false positives. This effect is more pronounced for lower $\boldsymbol{\alpha_{Scene}}$ values where the scene component has lower discriminativeness (See Figure 4.24(c)).

Finally, we evaluated our system with respect to the dimensionality of the pose feature vector. This dimensionality is determined by the number of "shape context words" formed in the shape dictionary. Figure 4.24(d) shows the accuracy of our system against the dimensionality of the pose feature vector. As expected, our performance reduces when using a very small number of words. In our experiments, we use a dictionary of 100 visual words resulting in a 100 dimensional pose feature vector.


Interaction: Lighting Flashlight
Interaction: Pouring
Interaction: Making a Phone Call
Interaction: Answering Phone

(a) HOG Detector (a) HOG in Framework

(b) HOG Detector (b) HOG in Framework

(c) HOG Detector (c) HOG in Framework

(d) HOG Detector (d) HOG in Framework

Figure 4.16: Results of object detection in the human-object interaction framework. The likelihoods of the centers of different objects are shown in different colors. The colors red, green, cyan and magenta show the likelihoods of cup, spray bottle, flashlight and phone respectively. (a) A flashlight is often confused as spray bottle by the HOG detector. However, when context from the framework is used there is no confusion. (b) Similarly a cup is often confused with a wide spray bottle. (c) Our detector can find and classify objects in clutter. (d) A spray bottle detector often fires at the handset of cordless phones due to the presence of parallel lines. However, such confusion can be removed using our

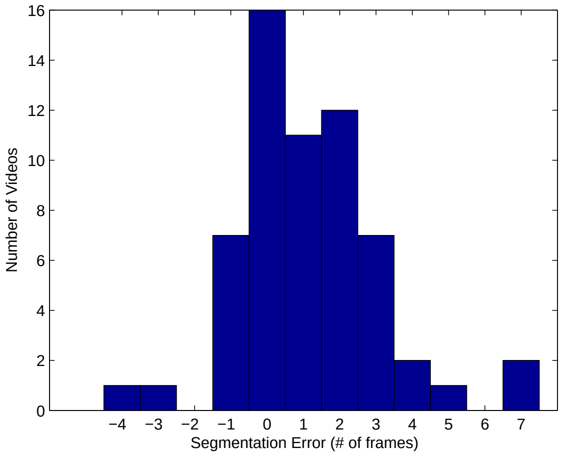


Figure 4.17: Segmentation Error Histogram

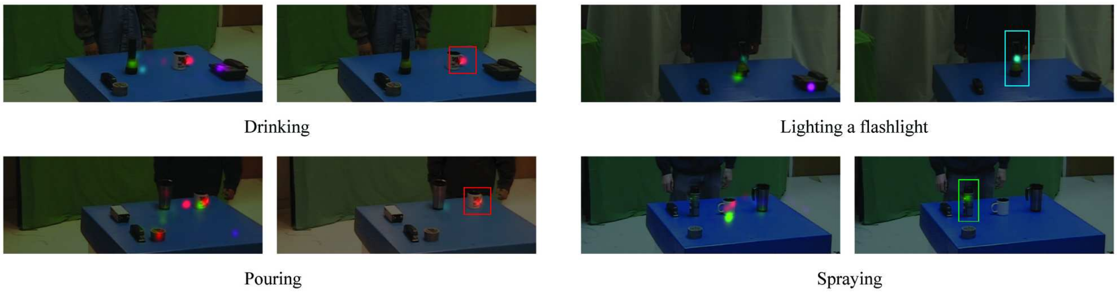


Figure 4.18: Object Recognition using contextual cues from reach, manipulation and object reaction. As before, the colors red, green, cyan and magenta show the likelihoods of cup, spray bottle, flashlight and phone respectively. The activities in the four cases above are: drinking, pouring, lighting, spraying respectively.

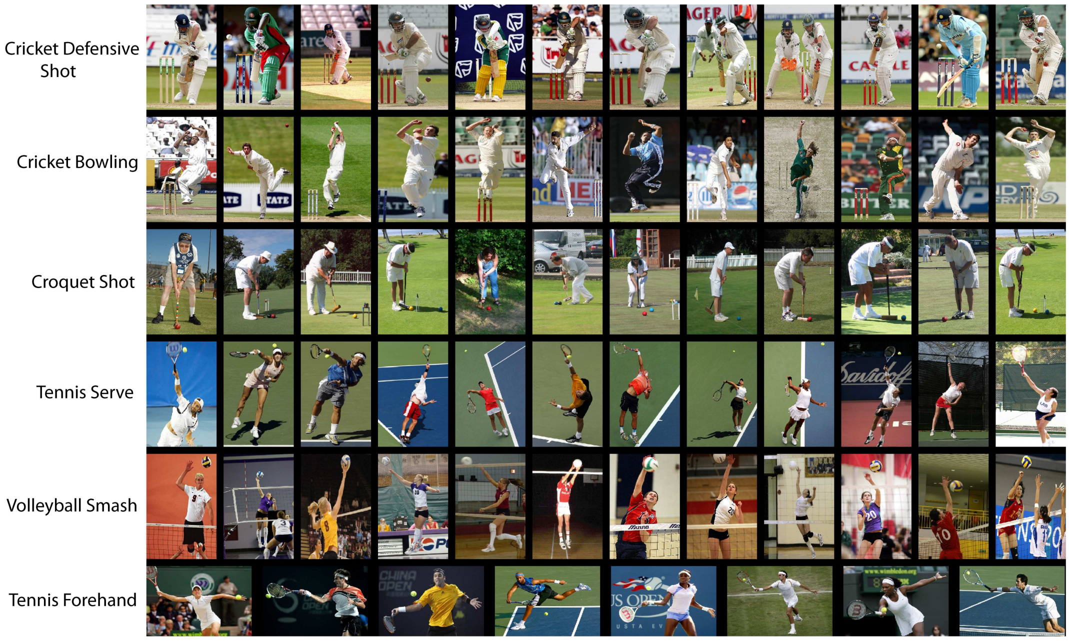
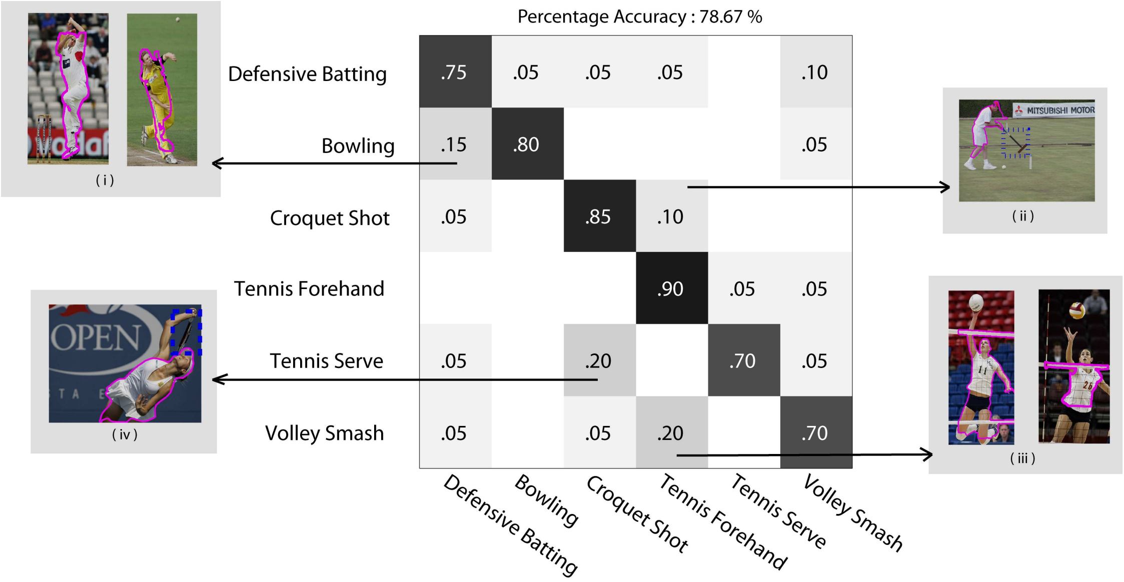


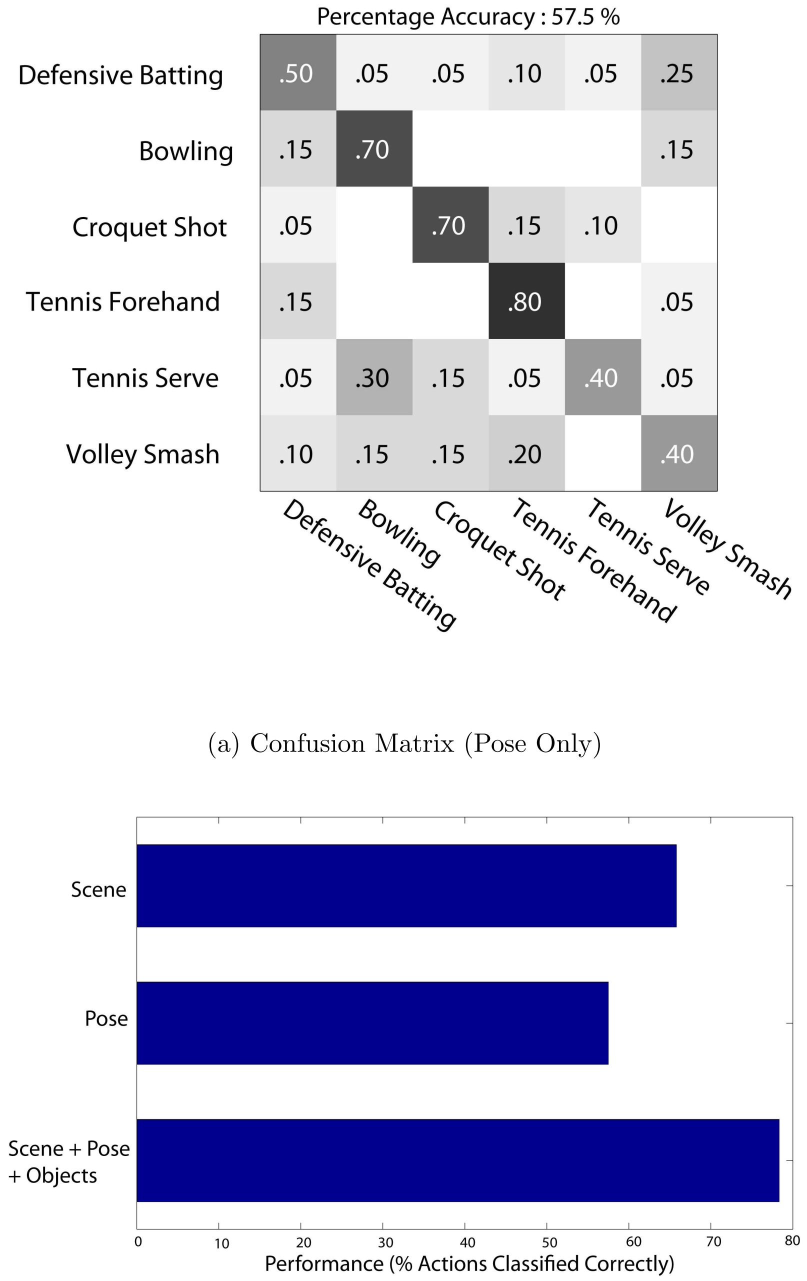
Figure 4.19: Our Dataset.



Percentage Accuracy : 78.67 %

	Defensive Batting	Bowling	Croquet Shot	Tennis Forehand	Tennis Serve	Volley Smash
Defensive Batting	.75	.05	.05	.05		.10
Bowling	.15	.80				.05
Croquet Shot	.05		.85	.10		
Tennis Forehand				.90	.05	.05
Tennis Serve	.05		.20		.70	.05
Volley Smash	.05		.05	.20		.70

Figure 4.20: Confusion Matrix (Full Model): The figure shows the confusion matrix obtained using the full model. We also show some failure cases in the adjoining boxes. (i) The scene in these cases are classified correctly as cricket ground however, due to faulty segmentations the hands of the bowler are missed and the pose is mis-classified as batting. (ii) The pose is again misclassified as that of forehand due to some extra regions added to human segment. The missed detection (shown in dotted blue) of croquet bat also contributes to the miss-classification. (iii) In both the cases the segmentation fails, leading to inclusion of net with the human segment. (iv) Apart from the error in the pose module, the racket is also missed and the ball is not present in the scene.



Percentage Accuracy : 57.5 %

	Defensive Batting	Bowling	Croquet Shot	Tennis Forehand	Tennis Serve	Volley Smash
Defensive Batting	.50	.05	.05	.10	.05	.25
Bowling	.15	.70				.15
Croquet Shot	.05		.70	.15	.10	
Tennis Forehand	.15			.80		.05
Tennis Serve	.05	.30	.15	.05	.40	.05
Volley Smash	.10	.15	.15	.20		.40

(a) Confusion Matrix (Pose Only)

(b) Comparison

Figure 4.21: (a) Confusion Matrix (Pose Only): The confusion matrix is only pose information is used for action classification. (b) Comparative performance of our approach with individual components.

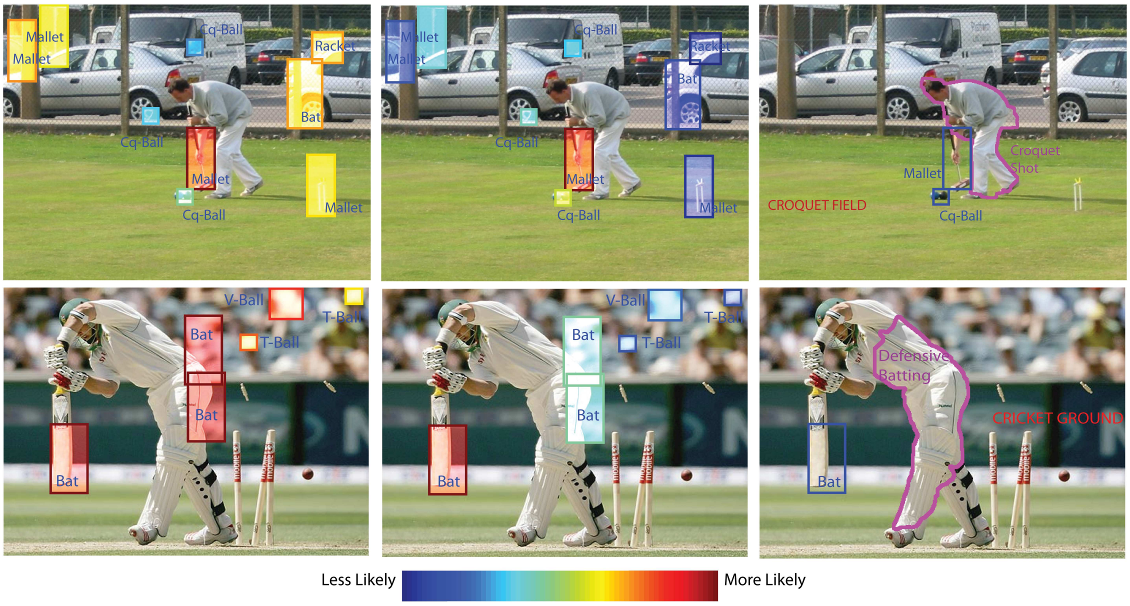


Figure 4.22: Some illustrative examples showing the performance of the system. The left column shows the likelihood of various objects using independent detectors. The colors of the rectangles represent the likelihood probability (red meaning higher probability and blue meaning lower probability). The middle column shows the posterior probabilities after the framework was applied. The right column shows the final result of our approach. In the first example, the detector detects four possible mallets and three possible croquet balls. After applying the spatial constraints, all the false positives are rejected as they fail to satisfy spatial constraints (the other mallets are not connected to a human body and the other balls are above the detected human centroid). In the second example, the false positives of bats are rejected as they fail to satisfy spatial constraints. Also, in both cases detections related to objects incongruent with scene and action information are also rejected. (Note the abbreviations *T-Ball*,*C-Ball*,*V-Ball* and *Cq-Ball* refer to tennis, cricket, volley and croquet balls respectively).

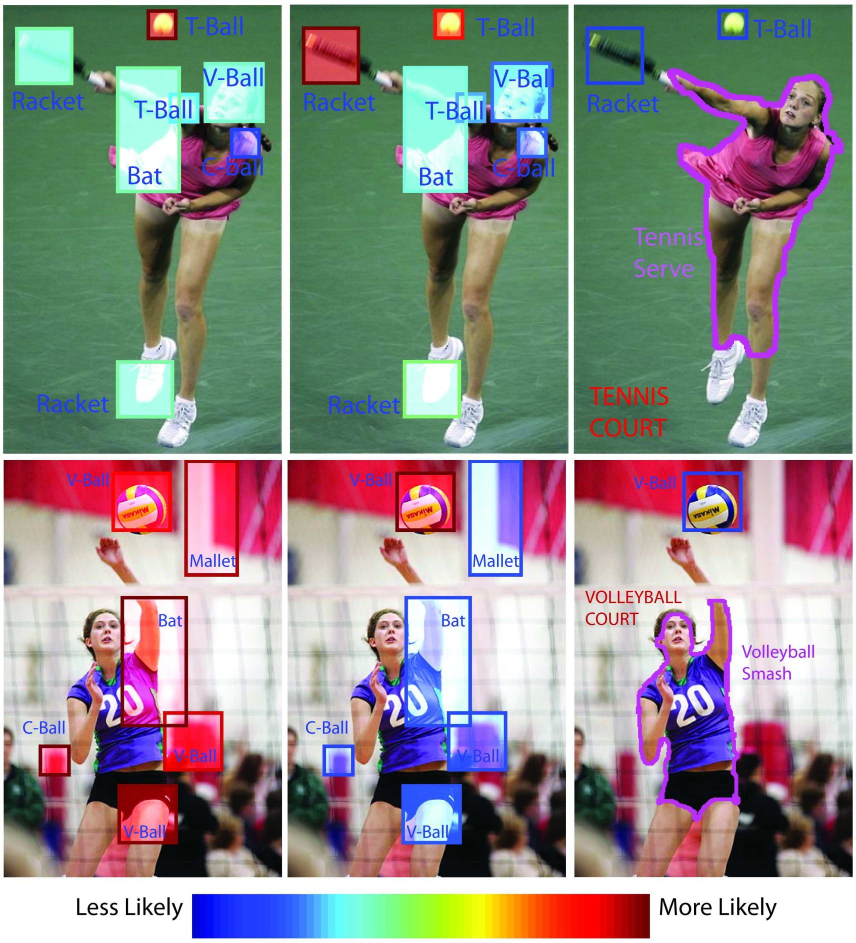
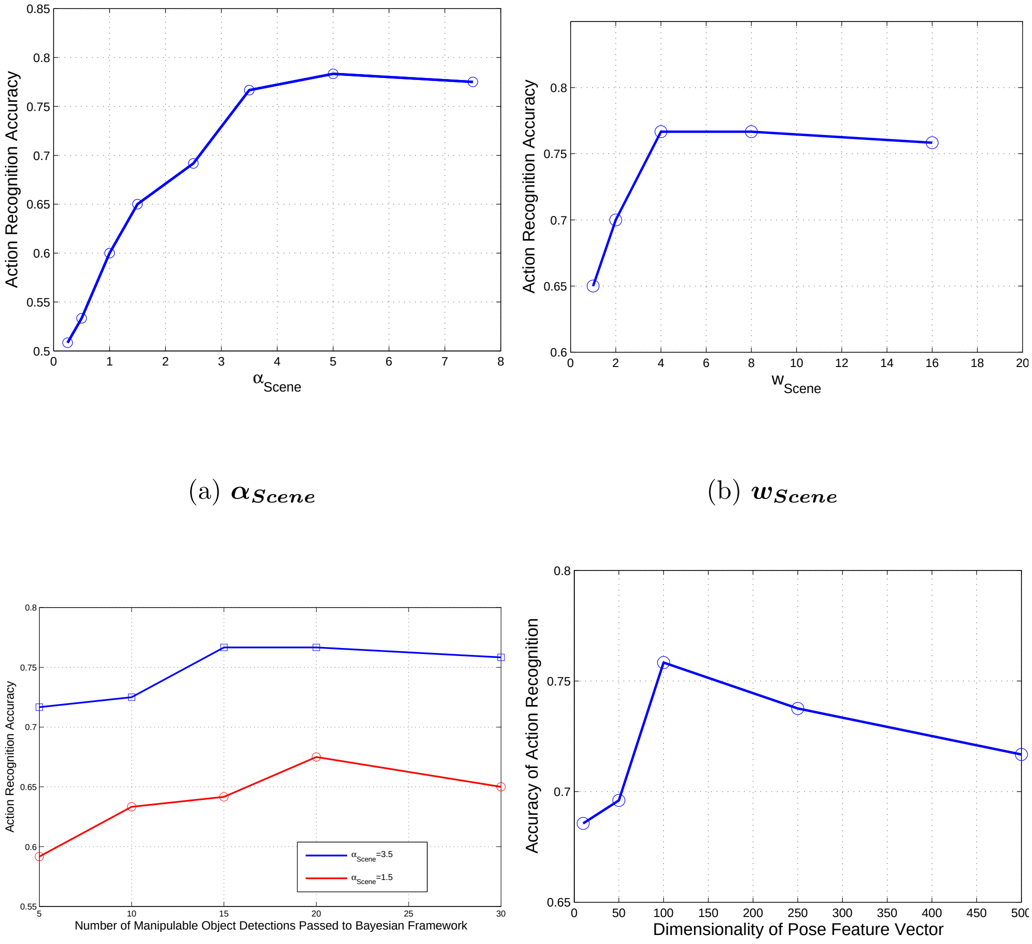


Figure 4.23: Some other examples: In the first case, the tennis racket was detected with a lower likelihood as compared to other objects. After combining information from scene and action the belief in the tennis racket increases since the action and the scene are tennis-serve and tennis court respectively. In the second case, our approach rejects false positives of objects such as a mallet and bat. These objects are rejected as they are not congruent to a volleyball-court and a volleyball-smash action. The false positives in volleyballs are also rejected as they fail to satisfy spatial constraints. Same abbreviations as in figure 4.22.



(a) $\boldsymbol{\alpha_{Scene}}$

(b) $\boldsymbol{w_{Scene}}$

(c) Number of Manipulable Objects

(d) Dimensionality of Pose Features

Figure 4.24: Influence of parameters on the system performance.

Chapter 5

Conclusion

In this thesis, I have explored how language can be integrated with visual learning systems to improve reliability of learning object and action models. This thesis explores how language beyond nouns and verbs can provide top-down grouping information or contextual information which can be used during learning from weakly labeled data. Unlike current approaches, which harness co-occurrence of visual features and nouns/verbs to constrain the learning, our approach uses the internal structure of images and videos to further constrain the learning. In Chapter 2, I have specifically shown how prepositions and comparative adjectives can provide relationship constraints which learned models should satisfy. I have also shown how prepositions and comparative adjectives can be used a as contextual model for scene analysis. In Chapter 3, we go beyond representing contextual model by priors on relationship words and learn a storyline model for videos. This storyline model is represented by an AND-OR graph and the edges in AND-OR graph are based on causal-dependency. AND-OR graphs allow representation of higher order constraints which are necessary for recognition of actions in videos. The storyline model approach goes beyond traditional paradigm of recognizing actions and in this case the problem is simultaneous estimation of storyline and recognition of individual actions. Finally in Chapter 4, I present two Bayesian models for interpretation of human object interactions from videos and static images respectively. Our approach combines the processes of scene, object, action and object reaction recognition. Our Bayesian model incorporates semantic/functional and spatial context for both object and action recognition. Therefore, by enforcing global coherence between different perceptual elements, we can improve the recognition performance of each element substantially.

Bibliography

[1] A. Agarwal and B. Triggs. 3d human pose from silhouettes by relevance vector regression. *IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.

[2] S. Andrews, I. Tsochantaridis, and T. Hoffman. Support vector machines for multiple-instance learning. *NIPS*, 2002.

[3] L. Armitage and P. Enser. Analysis of user need in image archives. *Journal of Information Science*, 1997.

[4] P. Bach, G. Knoblich, T. Gunter, A. Friederici, and W. Prinz. Action comprehension: Deriving spatial and functional relations. *J. Exp. Psych. Human Perception and Performance*, 31.

[5] K. Barnard, P. Duygulu, N. Freitas, D. Forsyth, D. Blei, and M. I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, pages 1107–1135, 2003.

[6] K. Barnard and Q. Fan. Reducing correspondence ambiguity in loosely labeled training data. *CVPR*, 2007.

[7] K. Barnard, Q. Fan, R. Swaminathan, A. Hoogs, R. Collins, P. Rondot, and J. Kaufold. Evaluation of localized semantics: data, methodology and experiments. *Univ. of Arizona, TR-2005*, 2005.

[8] K. Barnard and D. Forsyth. Learning the semantics of words and pictures. *ICCV*, pages 408–415, 2001.

[9] K. Barnard and M. Johnson. Word sense disambigutaion with pictures. *AI*, 2005.

[10] K. Barnard, K. Yanai, M. Johnson, and P. Gabbur. Cross modal disambiguation. *Toward Category-Level Object Recognition*, 2006.

[11] A. Berg and J. Malik. Geometric blur for template matching. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.

[12] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive gmmrf model. *European Conf. on Computer Vision*, 2004.

[13] S. Blakemore and J. Decety. From the perception of action to the understanding of intention. *Nature Reviews Neuroscience*, 2001.

[14] A. Bobick and Y. Ivanov. Action recognition using probabilistic parsing. *CVPR98.*

[15] A. Bobick and A. Wilson. A state-based approach to the representation and recognition of gesture. *IEEE Transactions of PAMI*, 19(12):1325–1337, 1997.

[16] A. Bosch, A. Zisserman, and X. Mu***ñ***oz. Image classification using random forests and ferns. In *IEEE Intl. Conf. on Computer Vision*, 2007.

[17] E. Brill. A simple rule-based part of speech tagger. *ACL*, 1992.

[18] E. Brill. Transformation-based error-driven learning and natural language processing. *Computational Linguistics*, 1995.

[19] P. Brown, S. Pietra, V. Pietra, and R. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 1993.

[20] D. Bub and M. Masson. Gestural knowledge evoked by objects as part of conceptual representations. *Aphasiology*, 20:1112–1124, 2006.

[21] P. Carbonetto, N. Freitas, and K. Barnard. A statistical model for general contextual object recognition. *ECCV*, 2004.

[22] G. Carneiro, A. B. Chan, P. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *IEEE PAMI*, 2007.

[23] L. L. Chao and A. Martin. Representation of manipulable man-made objects in dorsal stream. *NeuroImage*, 12:478–484, 2000.

[24] H. Chen, Z. Jian, Z. Liu, and S. Zhu. Composite templates for cloth modeling and sketching. *CVPR*, 2006.

[25] N. Dalal and B. Triggs. Histogram of oriented gradients for fast human detection. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[26] J. Davis, H. Gao, and V. Kannappan. A three-mode expressive feature model of action effort. In *IEEE Worskshop on Motion and Video Computing*, 2002.

[27] Z. Duric, J. Fayman, and E. Rivlin. Function from motion. *IEEE PAMI*, 18(6):579–591, 1996.

[28] P. Duygulu, K. Barnard, N. Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. *ECCV*, pages 97–112, 2002.

[29] M. Everingham, J. Sivic, and A. Zisserman. Hello! my name is...buffy - automatic naming of characters in tv video. *BMVC*, 2006.

[30] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 2003.

[31] S. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. *CVPR*, 2004.

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

[33] V. Ferrari and A. Zisserman. Learning visual attributes. *NIPS*, 2007.

[34] M. Fleischman and D. Roy. Situated models of meaning for sports video retrieval. *Human Language Tech.*, 2007.

[35] J. Freyd. The mental representation of movement when static stimuli are viewed. *Percept. Psychophys*, 1983.

[36] N. Friedman. The bayesian structural em algorithm. *UAI*, 1998.

[37] N. Friedman and D. Koller. Being bayesian about network structure: A bayesian approach to structure discovery in bayesian networks. *Machine Learning*, 2003.

[38] V. Gallese, L. Fadiga, L. Fogassi, and G. Rizzolatti. Action recognition in premotor cortex. *Brain*, 1996.

[39] S. Gong and T. Xiang. Recognition of group activities using dynamic probabilistic networks. *ICCV*, 2003.

[40] G. Guerra and Y. Aloimonos. Discovering a language for human activity. In *AAAI Work. on Anticipation in Cognitive Systems*, 2005.

[41] A. Gupta and L. Davis. Objects in action: An approach for combining action understanding and object perception. *CVPR'07*.

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

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

[44] A. Gupta, A. Mittal, and L. Davis. Constraint integration for efficient multiview pose estimation with self-occlusions. *IEEE Trans. on PAMI*, 2008.

[45] A. Gupta, P. Srinivasan, J. Shi, and L. Davis. Understanding videos, constructing plots: Learning a visually grounded storyline model from annotated videos. *CVPR*.

[46] H. B. Helbig, M. Graf, and M. Kiefer. The role of action representation in visual object. *Experimental Brain Research*, 174:221–228, 2006.

[47] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.

[48] S. H. J-Frey, F. R. Maloof, R. N.-Norlund, C. Farrer, S. Inati, and S. T. Grafton. Actions or hand-object interactions? human inferior frontal cortex and action observation. *Neuron*, 2003.

[49] T. Jellema and D. Perrett. Cells in monkey sts responsice to articulated body motions and consequent static posture: a case of implied motion. *Neuropsychologia*, 2003.

[50] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. *SIGIR*, 2003.

[51] R. Jin, J. Chai, and L. Si. Effective automatic image annotation via a coherent language model and active learning. *Mutimedia*, 2004.

[52] Z. Kourtzi. But still it moves. *Trends in Cog. Sc.*, 2004.

[53] Z. Kourtzi and N. Kanwisher. Activation in human mt/mst by static images with implied motion. *J. of Cognitive Neuroscience*, 2000.

[54] Y. Kuniyoshi and M. Shimozaki. A self-organizing neural model for context based action recognition. In *IEEE EMBS Conference on Neural Engineering*, 2003.

[55] D. Laganado and A. Solman. Time as a guide to cause. *J. of Exper. Psychology: Learning Memory & Cognition*, 2006.

[56] I. Laptev. On space-time interest points. *IJCV*, 2005.

[57] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. *CVPR'08*.

[58] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. *NIPS*, 2003.

[59] J. Li and J. Wang. Automatic linguistic indexing of pictures by statistical modeling approach. *IEEE PAMI*, 2003.

[60] L.-J. Li and L. Fei-Fei. What, where and who? classifying events by scene and object recognition. *IEEE Intl. Conf. on Computer Vision*, 2007.

[61] L. Lin, H. Gong, L. Li, and L. Wang. Semantic event representation and recognition using syntactic attribute graph grammar. *PRL*, 2009.

[62] R. Mann, A. Jepson, and J. Siskind. The computational perception of scene dynamics. *Comp. Vision and Image Understanding*, 65(2):113–128, 1997.

[63] O. Maron and A. Ratan. Multiple-instance learning for natural scene classification. *ICML*, 1998.

[64] R. Marteniuk, C. MacKenzie, M. Jeannerod, S. Athenes, and C. Dugas. Constraints on human arm movement trajectories. *Canadian Jnl. Psychology*, 1987.

[65] A. McCallum, D. Freitag, and F. Pereira. Maximum entropy markov models for information extraction and segmentation. *ICML*, 2000.

[66] A. D. Milner and M. A. Goodale. *The Visual Brain in Action.* Oxford University Press, 1995.

[67] T. B. Moeslund, A. Hilton, and V. Kruger. A survey of advances in vision-based human motion capture and analysis. *CVIU*, 2006.

[68] D. Moore, I. Essa, and M. Hayes. Exploiting human action and object context for recognition tasks. In *IEEE Intl. Conf. on Computer Vision*, 1999.

[69] Y. Mori, H. Takahashi, and R. Oka. Image to word transformation based on dividing and vector quantizing images with words. *MISRM*, 1999.

[70] H. Murase and S. Nayar. Learning object models from appearance. In *National Conf. on Artificial Intelligence*, 1993.

[71] K. Murphy, A. Torralba, and W. Freeman. Graphical model for scenes and objects. In *NIPS*, 2003.

[72] K. Murphy, A. Torralba, and W. Freeman. Using the forest to see the trees: a graphical model relating features, objects and scenes. *NIPS*, 2004.

[73] H. Nagel. From image sequences towards conceptual descriptions. *Image and Vision Computing*, 1988.

[74] C. Needham, P. Santos, R. Magee, V. Devin, D. Hogg, and A. Cohn. Protocols from perceptual observations. *AI'05*.

[75] K. Nelissen, G. Luppino, W. Vanduffel, G. Rizzolatti, and G. Orban. Observing others: Multiple action representation in frontal lobe. *SCIENCE*, 310:332–336, 2005.

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

[77] N. Nitta, N. Babaguchi, and T. Kitahashi. Extracting actors, actions an events from sports video - a fundamental approach to story tracking. *ICPR*, 2000.

[78] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Intl. Journ. of Computer Vision*, 2001.

[79] J. Pearl. Causality: Models, reasoning, and inference. *Cambridge University Press, 2000.*

[80] J. Pearl. Heuristics: Intelligent search strategies for computer problem solving. *Addison-Wesley*, 1984.

[81] P. Peursum, G. West, and S. Venkatesh. Combining image regions and human activity for indirect object recognition in indoor wide-angle views. In *IEEE Intl. Conf. on Computer Vision*, 2005.

[82] V. Prasad, V. Kellokompu, and L. Davis. Ballistic hand movements. In *AMDO*, 2006.

[83] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *IEEE Intl. Conf. on Computer Vision*, 2007.

[84] C. Rao, A. Yilmaz, and M. Shah. View-invariant representation and recognition of actions. *IJCV*, 2002.

[85] E. Rivlin, S. Dickinson, and A. Rosenfeld. Recognition by functional parts. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 1994.

[86] M. Shah and R. Jain. Motion-based recognition. *Computational Imaging and Vision Series*, 1997.

[87] I. Smyth and M. Wing. *The Psychology of Human Movement.* The Psychology of Human Movement, 1984.

[88] M. Srikanth, J. Varner, M. Bowden, and D. Moldovan. Exploiting ontologies for automatic image annotation. *SIGIR*, 2005.

[89] L. Stark and K. Bowyer. Generic recognition through qualitative reasoning about 3d shape and object function. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 1991.

[90] E. Sudderth, A. Torralba, W. Freeman, and A. Wilsky. Learning hierarchical models of scenes, objects and parts. In *IEEE Intl. Conf. on Computer Vision*, 2005.

[91] J. Sullivan and S. Carlsson. Recognizing and tracking human action. *European Conf. on Computer Vision*, 2002.

[92] S. Todorovic and N. Ahuja. Learning subcategory relevances for category recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.

[93] A. Torralba and P. Sinha. Statistical context priming for object detection. In *IEEE Intl. Conf. on Computer Vision*, 2001.

[94] S. Tran and L. Davis. Visual event modeling and recognition using markov logic networks. *ECCV'08.*

[95] C. Urgesi, V. Moro, M. Candidi, and S. Aglioti. Mapping implied body actions in the human motor system. *J. of Neuroscience*, 2006.

[96] L. Vaina and M. Jaulent. Object structure and action requirements: A compatibility model for functional recognition. *Int. Journal of Intelligent Systems*, 6:313–336, 1991.

[97] A. Vezhnevets and V. Vezhnevets. Modest adaboost' - teaching adaboost to generalize better. *Graphicon*, 2005.

[98] Y. Wang, H. Jiang, M. Drew, Z. Li, and G. Mori. Unsupervised discovery of action classes. *IEEE Conf. on Computer Vision and Pattern Recognitio*, 2006.

[99] A. Wilson and A. Bobick. Parametric hidden markov models for gesture recognition. *IEEE PAMI*, 1999.

[100] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *IEEE Intl. Conf. on Computer Vision*, 2005.

[101] B. Wu and R. Nevatia. Detection and tracking of multiple humans with extensive pose articulation. *IEEE Intl. Conf. on Computer Vision*, 2007.

[102] A. Yilmaz and M. Shah. Actions sketch: A novel action representation. *IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[103] L. Zhu, C. Lin, H. Huang, Y. Chen, and A. Yuille. Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence, competitive exclusion. *ECCV'08.*

[104] Q. Zhu, S. Avidan, M. Ye, and K. Cheng. Fast human detection using a cascade of histograms of oriented gradients. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.

[105] S. Zhu and D. Mumford. A stochastic grammar of images. *Foundations and Trends in Comp. Graphics and Vision*, 2006.