ABSTRACT

Title of dissertation: CONTEXT DRIVEN SCENE UNDERSTANDING

Xi (Stephen) Chen, Doctor of Philosophy, 2015

Dissertation directed by: Professor Larry S. Davis
Department of Computer Science

Understanding objects in complex scenes is a fundamental and challenging problem in computer vision. Given an image, we would like to answer the questions of whether there is an object of a particular category in the image, where is it, and if possible, locate it with a bounding box or pixel-wise labels. In this dissertation, we present context driven approaches leveraging relationships between objects in the scene to improve both the accuracy and efficiency of scene understanding.

In the first part, we describe an approach to jointly solve the segmentation and recognition problem using a multiple segmentation framework with context. Our approach formulates a cost function based on contextual information in conjunction with appearance matching. This relaxed cost function formulation is minimized using an efficient quadratic programming solver and an approximate solution is obtained by discretizing the relaxed solution. Our approach improves labeling performance compared to other segmentation based recognition approaches.

Secondly, we introduce a new problem called object co-labeling where the goal is to jointly annotate multiple images of the same scene which do not have temporal consistency. We present an adaptive framework for joint segmentation and recognition to solve this problem. We propose an objective function that considers not only appearance but also appearance and context consistency across images of the scene. A relaxed form of the cost function is minimized using an efficient quadratic programming solver. Our approach improves labeling performance compared to labeling each image individually. We also show the application of our co-labeling framework to other recognition problems such as label propagation in videos and object recognition in similar scenes.

In the third part, we propose a novel general strategy for simultaneous object detection and segmentation. Instead of passively evaluating all object detectors at all possible locations in an image, we develop a divide-and-conquer approach by actively and sequentially evaluating contextual cues related to the query based on the scene and previous evaluations—like playing a "20 Questions" game—to decide where to search for the object. Such questions are dynamically selected based on the query, the scene and current observed responses given by object detectors and classifiers. We first present an efficient object search policy based on information gain of asking a question. We formulate the policy in a probabilistic framework that integrates current information and observation to update the model and determine the next most informative action to take next. We further enrich the power and generalization capacity of the Twenty Questions strategy by learning the Twenty Questions policy driven by data. We formulate the problem as a Markov Decision Process and learn a search policy by imitation learning.

CONTEXT DRIVEN SCENE UNDERSTANDING

by

Xi Chen

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
2015

Advisory Committee:
Professor Larry S. Davis, Chair/Advisor
Professor Rama Chellappa
Professor James Reggia
Professor Ramani Duraiswami
Professor Héctor Corrada Bravo

Dedication

To my parents who raised me up with unconditional love. To whom makes this dissertation possible.

Acknowledgments

I owe my gratitude to all the people who have made this thesis possible and because of whom my graduate experience has been one that I will cherish forever.

First and foremost I owe my gratitude to my advisor, Professor Larry S. Davis for supporting me to work on this challenging and interesting topic over the past several years. Professor Davis was always available whenever I sought his help and advices. My discussions with him were always encouraging and inspiring. More importantly, he gave me the maximum freedom to explore the exciting unknown. It has been a precious experience to work with and learn from him, which I will continue to benefit from and cherish in my future career.

Secondly, I would like to thank Prof. Rama Chellappa for meaningful comments during my proposal exam and our project meetings. Also, I enjoyed his courses during my graduate studies very much, which also guided my journey in computer vision. I'd also like to show my gratitude to Professor James Reggia, Professor Ramani Duraiswami and Professor Héctor Corrada Bravo for agreeing to serve on my thesis committee and for sparing their invaluable time reviewing the manuscript.

I also would like to thank those collaborators that make this dissertation possible: Dr. Abhinav Gupta and Dr. Arpit Jain offered me a lot of guidance during my early stage of my PhD research which pointed out my direction in my PhD research. He He and I had a lot of helpful discussions to brainstorm and formulate the "20 questions" parts of the thesis. Without their generous help and collaboration, I could not move as far either on my research work or the projects. Special thanks to Dr. Ming-Yu Liu, Dr. Oncel Tuzel,

Prof. Gregory Shakhnarovich and Prof. Qixing Huang for their mentoring during my research internships. I really enjoyed working with them and learned a lot from them. I also thank lots of other group members, Zhuolin, Vlad, Ruiping, Bhejat, Choi, ... for their valuable discussions.

And I would like to thank all my colleagues in the computer vision laboratory that have enriched my graduate life in many ways: Huimin, Ruonan, Guangxiao, Jun-Cheng, Le Kang, Fan, Joe, Muzi, Ang, Brandyn, HyungTae, Ejaz, Sameh, Sravanthi, Varun...I thank lots of my friend in Maryland and in other parts of the world, your accompany made my Ph.D. exploration much more enjoyable.

I owe my deepest thanks to my family - my mother and father who have always stood by me and guided me through my career, and have pulled me through against impossible odds at times. Words cannot express the gratitude I owe them.

It is impossible to remember all, and I apologize to those I've inadvertently left out. This dissertation will be impossible without you.

Lastly, thank you all!

Table of Contents

List of Figures

List of Abbreviations

UMIACS	University of Maryland Institute for Advanced Computer Studies
SVM	Support Vector Machine
IPFP	Integer Projected Fixed Point
QP	Quadratic Programming
CNN	Convolutional Neural Network
mAP	mean average precision

Chapter 1: Introduction

Understanding objects in complex scenes is a central and challenging problem in computer vision. With the surge of real world applications of computer vision, it is essential for computers to better understand the world it perceives from cameras and sensors, especially in scenarios such as self-driving cars and robots. Also it would help people to understand, organize and discover interesting patterns from the tremendous amount of images online. Specifically, given an image, we would like to answer the questions of whether there is an object of a particular category in the image, where is it, and if possible, locate it with a bounding box or pixel-wise labels. To address these questions, the computer vision community has been working on problems defined as object detection, object recognition, image parsing and semantic segmentation. Typical object detection and recognition approaches usually consider only the appearance of the object in a single image. However, due to variation in data distribution, occlusion and viewpoint change, object models may not always capture the appearance of objects and ambiguity arises.

In contrast, in the real world objects never occur in isolation; they co-vary with other objects and particular environments, providing a rich source of contextual associations to be exploited by the visual system. Instead of looking at objects in the world exclusively based on their individual appearance, humans views a scene as a whole picture with rich relations between objects. It has been shown in various studies on visual cognition [1, 2], computer vision [3,4] and cognitive neuroscience [5,6] that contextual information affects the efficiency of the search for and recognition of objects. Such studies are the intuition and inspiration of our work the this thesis to exploit the rich contextual information for scene understanding problems.

In this dissertation, we address the important yet challenging problems of scene understanding by utilizing context information. We focus on the tasks of object recognition and detection. We will first introduce these problems of scene understanding, then discuss the role of context in object recognition and detection. We then present context driven approaches leveraging spatial relationships between objects in the scene to improve both the accuracy and efficiency of scene understanding.

The main contributions of this dissertation are:

- **Object recognition with context in multiple segmentation:** we propose to consider both appearance and context cues in a multiple segmentation framework that can jointly segment and label the image with an efficient optimization scheme.

- **Co-label objects in multiple images:** we propose to label objects in multiple images of the same or similar scene to improve recognition by leveraging the appearance and context consistency across images.

- **Object detection in Twenty Questions:** we propose a novel general strategy for object proposal-based object detection. Instead of passively evaluating all object detectors at all possible locations in an image, we develop a divide-and-conquer approach by actively and sequentially evaluating contextual cues related to the query based on the scene and previous evaluations—like playing a “20 Questions” game—to decide where to search for the object.

1.1 Image Parsing and Object Recognition

We define image parsing to be the task of decomposing an image into its constituent visual patterns. We focus on parsing the image into semantic objects. The problem of image parsing has a long history in computer vision dating back to the 1970’s. Unlike Marr’s sequential processing pipeline, where segmentation from bottom-up cues preceded recognition, Tenenbaum and Barrow proposed Interpretation-Guided Segmentation [7] which labeled image regions using constraint propagation to arrive at a globally consistent scene interpretation. This was followed by development of complete scene understanding systems such as ACRONYM [8] and VISIONS [9]. During the last decade, researchers in visual recognition have made significant advances in object recognition due to better appearance modeling techniques and visual context. These approaches can be broadly categorized into three categories based on how interactions between segmentation and recognition are modeled:

1.1.1 Pixel Based Approaches

These approaches model the problem of visual recognition at the pixel level [10–13] and therefore the problem of segmentation is solved implicitly (neighboring pixels belonging to different class represent boundary pixels). One of the major shortcomings of pixel-based approaches is that many objects (such as cars) are defined in large part by their shape and therefore categorization at the pixel-level using local appearances without global shape analysis performs poorly.

1.1.2 Fixed Segmentation Approaches

These approaches classify individual regions in some fixed image segmentation based on region color, texture and shape [14–16]. However, obtaining semantically meaningful segmentations without top-down control is well beyond the state of the art.

1.1.3 Joint Segmentation and Recognition

These approaches jointly solve segmentation and recognition. Approaches such as [17,18] obtain multiple segmentations of the image and model the problem of segmentation and recognition as the selection of segments based on their matches to semantic classes. On the other hand, approaches such as [19–21] start from an imperfect segmentation and then refine it iteratively by optimizing a cost function defined on segments and appearance matching. One of the shortcomings of these approaches is that they tend to get stuck in local minima due to local refinement. [22,23] proposed super pixel based approaches where the class labels are inferred based on local appearance and context using CRFs. Such approaches fail to incorporate higher level shape information; additionally learning CRF's parameters has proven to be difficult. In [24] segmentation was combined with the responses of sliding window object detectors for image labeling to avoid fragility of segmentation.

1.1.4 Multiple Segmentation using Context

For object recognition, there has been a recent trend to simultaneously address segmentation and recognition. However, these methods use only appearance features to select segments and the best overall labeling is constructed in a greedy manner. They ignore context, which is important for accurate segment selection and labeling.

To address the shortcomings of the previous approaches of image parsing, in the first part of the thesis, we describe an approach to jointly solve the segmentation and recognition problem using a multiple segmentation framework using context. We propose an approach to select the best segmentation and labeling in a single optimization procedure that utilizes context to perform segment selection and labeling coherently. To overcome the fragmentation problem, we allow connected segments to be merged based on local color, texture and edge properties. We also include mid-level cues to constrain the solution space - for example, the segment merging step leads to overlapping segments, and we restrict global solutions to exclude overlapping segments (avoiding the possibility of multiple labeling for pixels). By incorporating contextual relations between region pairs, we find the subset of segments that best explains the image.

We formulate a cost function based on contextual information in conjunction with appearance matching. This relaxed cost function formulation is minimized using an efficient quadratic programming solver and an approximate solution is obtained by discretizing the relaxed solution. Our approach improves labeling performance compared to other segmentation based recognition approaches.

1.2 Object Co-Labeling in Multiple Images

With the recent surge in photos and videos taken from hand-held devices and those shared online, many of which are taken of the same scenes, the need to automatically label objects in such image sets has emerged. Traditional approaches to recognition typically consider only a single test image using appearance and contextual cues. However, modeling relationship between objects is difficult as they are also viewpoint dependent and do not generalize well.

We introduce a new problem called object co-labeling where the goal is to jointly annotate multiple images of the same scene which do not have temporal consistency. We present an adaptive framework for joint segmentation and recognition to solve this problem. We propose an objective function that considers not only appearance but also appearance and context consistency across images of the scene.Our approach improves labeling performance compared to labeling each image individually. We also show the application of our co-labeling framework to other recognition problems such as label propagation in videos and object recognition in similar scenes.

1.3 Object Detection and Localization

1.3.1 Background

Object detection is the task that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. Some common approaches to object detection are based on applying gradient based features over densely sampled sliding windows [60], which are very inefficient since they evaluate up to hundreds of thousands of windows in an image, and false positive detections arise. To reduce the number of windows evaluated, category independent object proposals [61–63] have been proposed which generate a small number of high quality regions or windows that are likely to be objects. These approaches dramatically reduce the number of candidates and reduce false positive detections. Using these object proposals [73, 74, 79] train and apply deep neural network models to learn the feature extractor and classifiers, and achieve state-of-the-art performance on the Pascal VOC detection challenge. However, such category independent proposals do not adapt to different query classes and still lead to a significant amount of unnecessary detector computation.

1.3.2 Context in Object Detection

Context is not only helpful in parsing objects in the scene, but also enhances object detectors by eliminating false positives and improving precision [37]. The sources of contextual information can be in the form of global scene context, ground plane estimation, geometric context in the form of 3D surface orientations, relative location, 3D layout, spatial support and geographic information.

Usually, context-sensitive methods rely on holistic models that consider relations of the query object with all other object classes in the scene at the same time. This is highly inefficient since many non-informative contextual objects have to be queried. We propose a new strategy for simultaneous object detection and segmentation in the scene.

1.3.3 Object Detection in 20 Context Related Questions

Instead of evaluating classifiers for all possible locations of objects in the image, we develop a divide-and-conquer approach by sequentially posing questions for computer to answer given query and the image, like playing a “Twenty Questions” game. Such questions are dynamically selected based on the query, the scene and current observed responses given by object detectors and classifiers. We present an efficient object search policy that considers the most informative questions for both the query and the scene. This policy is driven by a semantic contextual model which sequentially refine the search area for the query. We first formulate the policy in one probabilistic framework that integrates the current information and history observation to update the model and determine the next most informative action to take. We then generalize the decision making capability of the policy by formulating the problem as a Markov Decision Process (MDP) and learn the policy by imitation learning fully driven by data. Experiments show promising results compared with baselines of exhaustive search, searching for objects in random sequences and random locations.

1.4 Organization

The dissertation is organized as follows. In Chapter 2, we present an approach to jointly solve segmentation and recognition problem for scene labeling. We formulate the problem as similar to solving jigsaw puzzle and propose an Integer Programming based solution. In Chapter 3 we study the problem called object co-labeling where the goal is to jointly annotate multiple images of the same scene. In Chapter 4, we propose a sequential approach to detect and locate the query class objects in the image which dynamically selects questions at each step. The policy is based on maximizing the information gain of a question based on the query, the scene and the current observations. In Chapter 5 we formulate the object detection problem as a Markov Decision Process and provide an imitation learning based approach to learn a policy to decide the next context cues to query. In Chapter 6, we conclude and explore future research directions.

Chapter 2: Piecing Together the Segmentation Jigsaw using Context

2.1 Introduction

We describe an approach that jointly segments and labels the principal objects in an image. Consider the image in Figure 2.1. Our goal is to locate and pixel-wise label the principal objects such as car, building, road and sidewalk. One approach is to first segment the image, then perform recognition using appearance and context. However, there are generally no reliable algorithms for segmentation. For example, for the image shown in Figure 2.1, segmentation algorithms will generally not combine the roof and the body of the car into one segment due to differences in appearances. Therefore, there has been a recent trend to simultaneously address segmentation and recognition.

For example, some recent approaches construct the segments by selectively merging superpixels while simultaneously labeling these elements. However, at the superpixel level global image features such as shape cannot be easily employed. So, while these approaches show high performance for "stuff"-like objects such as grass - they often fail to identify objects which require shape cues for identification. To harness shape features, approaches such as [19,20] have instead started with an initial segmentation and then refined these segments iteratively. However, the modifications are generally local in nature and tend to get stuck in local minima.

To overcome these problems, recent approaches have advocated the use of multiple segmentations [18, 25]. Recognition, then, involves selecting the best segments. These methods use only appearance features to select segments and the best overall labeling is constructed in a greedy manner. They ignore context, which is important for accurate segment selection and labeling. For example, the window of the car is labeled as "airplane" because the context from other scene elements such as road, sidewalk and building are ignored.

We propose an approach to select the best segmentation and labeling in a single optimization procedure that utilizes context to perform segment selection and labeling coherently. To overcome the fragmentation problem, we allow connected segments to be merged based on local color, texture and edge properties. We also include mid-level cues to constrain the solution space - for example, the segment merging step leads to overlapping segments, and we restrict global solutions to exclude overlapping segments (avoiding the possibility of multiple labeling for pixels). By incorporating contextual relations between region pairs, we find the subset of segments that best explains the image. For example, in Figure 2.1, our approach correctly selects the combined region of window and body segments and labels it as "car". The labeling of the window segment as "airplane" is not chosen due to contextual constraints from sidewalk, road and building.

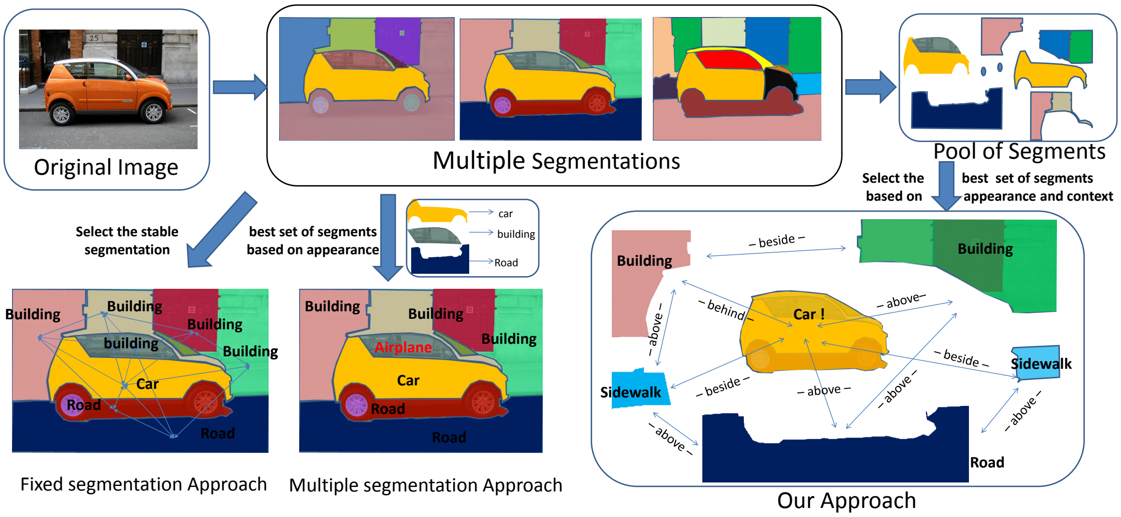

Figure 2.1: Comparison of our approach to fixed and multiple segmentation algorithms. Our approach solves the problem of segmentation and recognition jointly using appearance and context. The figure shows how global contextual relations help to select the whole car segment subset over other fragmented pieces of car, as their association does not satisfy context.

The contributions of our work are: (a) An approach to incorporate contextual information in a multiple segmentation framework, and (b) Increasing the spatial support[1] of image labeling by constructing additional segments from a base pool, at the cost of only a small increase in segment pool size.

2.2 Related Work

The problem of image parsing has a long history in computer vision dating back to the 1970's. Unlike Marr's sequential processing pipeline, where segmentation from bottom-up cues preceded recognition, Tenenbaum and Barrow proposed Interpretation-Guided Segmentation [7] which labeled image regions using constraint propagation to arrive at a globally consistent scene interpretation. This was followed by development of complete scene understanding systems such as ACRONYM [8] and VISONS [9]. During the last decade, researchers in visual recognition have made significant advances in object recognition due to better appearance modeling techniques and visual context. These approaches can be broadly categorized into three categories based on how interactions between segmentation and recognition are modeled:

Pixel Based Approaches: These approaches model the problem of visual recognition at the pixel level [10–13] and therefore the problem of segmentation is solved implicitly (neighboring pixels belonging to different class represent boundary pixels). One of the major shortcomings of pixel-based approaches is that many objects (such as cars) are defined in large part by their shape and therefore categorization at the pixel-level using

[1] Spatial support measures the quality of pool of segments as compared to ground truth. The score is higher if the segments in the ground-truth find segments in the pool with high overlap.

local appearances without global shape analysis performs poorly.

Fixed Segmentation Approaches: These approaches classify individual regions in some fixed image segmentation based on region color, texture and shape [14–16]. However, obtaining semantically meaningful segmentations without top-down control is well beyond the state of the art.

Image Parsing (Joint Segmentation and Recognition): These approaches jointly solve segmentation and recognition. Approaches such as [17, 18] obtain multiple segmentations of the image and model the problem of segmentation and recognition as the selection of segments based on their matches to semantic classes. On the other hand, approaches such as [19–21] start from an imperfect segmentation and then refine it iteratively by optimizing a cost function defined on segments and appearance matching. One of the shortcomings of these approaches is that they tend to get stuck in local minima due to local refinement. [22,23] proposed super pixel based approaches where the class labels are inferred based on local appearance and context using CRFs. Such approaches fail to incorporate higher level shape information; additionally learning CRF's parameters has proven to be difficult. In [24] segmentation was combined with the responses of sliding window object detectors for image labeling to avoid fragility of segmentation.

2.3 Overview

Multiple segmentation approaches construct a pool of initial segments by varying the controlling parameters of a segmentation algorithm or by starting from a coarse segmentation and iteratively refining the segmentation by merging or further segmenting initial segments. They generally assume that each object will be well segmented at some parameter setting or level. [26] pointed out that merging small connected subsets (pairs and triples) of base segments improves recognition performance. However, the algorithm in [26] employed manually choosing the segments to merge. One could simply join all possible pairs and triples of connected segments but this would lead to an explosion in the segment pool size. In contrast, we construct a "good" set of mergings using a classifier which rejects combination which are unlikely to correspond to "complete" objects (section 4).

We organize these segments into a hierarchical segment graph for recognition. The graph structure allows us to impose constraints that reduce the combinatorics of the search process - for example, that a solution cannot include overlapping segments, since this could lead to pixels being given multiple labels.

Given the segment graph, we compute pairwise and higher-order constraints on selection of segments. We then formulate a cost function which accounts for local appearance and enforces pair-wise contextual relationship consistency (such as sky above water, road below car, etc). Directly optimizing this cost function is NP hard so the cost function is approximately minimized by first relaxing the selection problem. The relaxed problem can be solved efficiently by quadratic programming (QP). The relaxed solution is then discretized to obtain the final labeled segmentation (section 5). Finally, we evaluate the performance of our approach with previously reported methods (section 6).

2.4 Constructing the Segment Graph

Obtaining the Initial Segment Pool: We use the hierarchical segmentation algorithm from [27] to construct the segment pool. To increase the robustness of the segmentation algorithm, we use the stability based clustering analysis of [28]. Stability analysis selects segments which are stable under small perturbations (noise) to the image.

In the first step, image is segmented and the segments in the first hierarchical level are added to the segment pool. Then each of these segments is iteratively segmented and the smaller segments are added to the segment pool until any of the following conditions are met. (1) The segment size is too small ($< 2\%$ of total image pixels). (2) The integrated edge strength along the boundary of the segment (obtained by Berkeley edge detector [29]) is below a threshold. (3) The number of leaf nodes in the segment subgraph rooted at the original segment exceeds a threshold.

This procedure gives us initial segment pool over which we will perform segment selection.

Merging Segments: The base segmentation algorithm seldom produces segments that directly correspond to the objects in the image. Hence, we merge small (2 and 3) connected sets of segments from the segment pool to obtain a better collection of segments. But allowing all possible segment merges would explode the size of the pool. To limit the number of pairs and triples merged, we learn a function that scores these small subsets from a training set of fully labeled images.

A Support Vector Regression (SVR) [30] model using radial basis functions is learned from the training images to score potential merges. We compute color, texture and edge features similar to those used by Hoiem et. al. [31] for each segment of an object. Based on these features, the SVR predicts whether the segments should be merged or not. Training images are segmented using the segmentation algorithm described above and a segment pool is obtained for each image. Objects which are broken into multiple segments are determined using the ground truth segmentation. These fragmented objects provide positive examples and the negative examples are obtained using random samplings from the training data. For a testing image, each adjacent pair and connected triple[2] of segments is evaluated for merging using the regression model learned, providing a score for each merging. The pairs and triples with scores above a threshold are added to the segment pool.

[2] triples of segments are constructed by evaluating merging of a segment from the initial pool with an adjacent segment formed from the pairwise merging step.

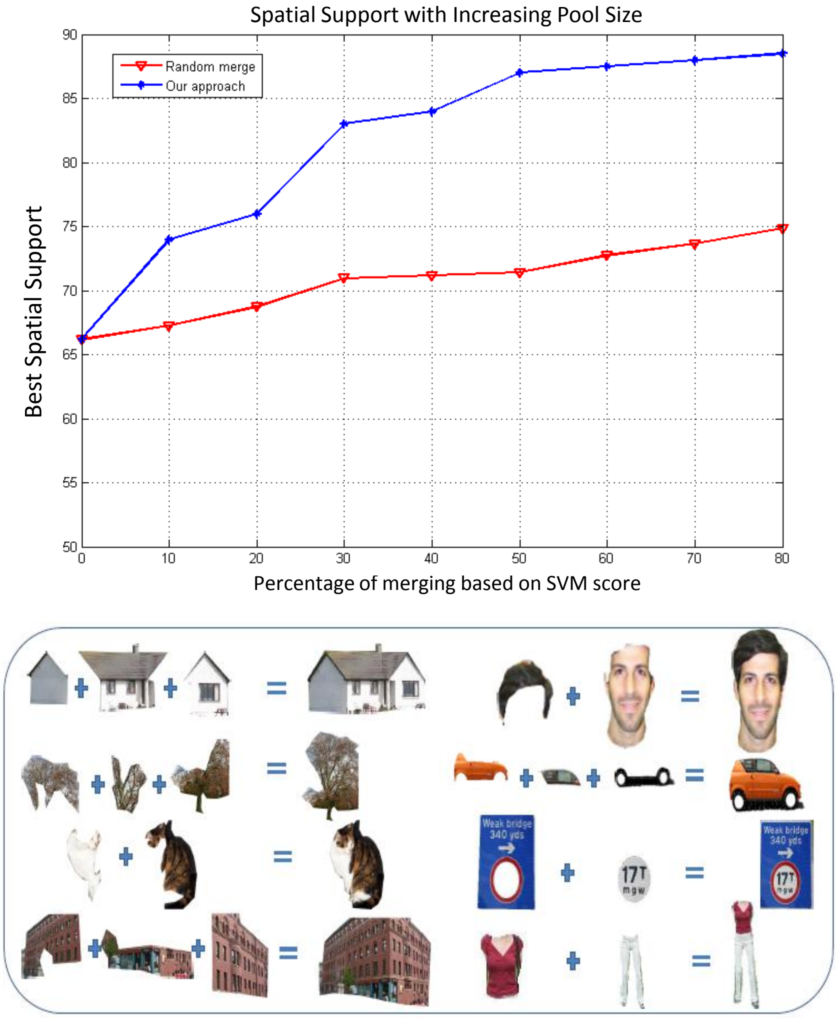

Figure 2.2: Graph on top shows the improvement in spatial support with increase in pool size. Image below the graph shows the instances where SVR model correctly merged fragmented segments of objects in the pool to complete the object segment.

We evaluated the merging scheme on the 256 test images in the MSRC dataset. Figure 2.2 shows the spatial support in the pool with increasing pool size. The pool size is increased by lowering the threshold at which mergings are accepted. To demonstrate that the SVR learns an informative merging function, we compare the spatial support metric when the segment pool is enlarged using random merges (red curve in Figure 2.2). Although spatial support increases (which it obviously must), it does so at a much slower rate than the SVR.

Construction of the Segment Graph: The pool of segments are then arranged in a hierarchical graph structure to which our inference algorithm will subsequently be applied. The graph structure is constructed as follows: The root node is assigned to the whole image. A segment S_i is a child of segment S_j if segment $S_i \subset S_j$. If two segments S_i and S_j are subsets of a S_k then both the segments are children of segment S_k. The segments which have no smaller segment subsets are leaf nodes.

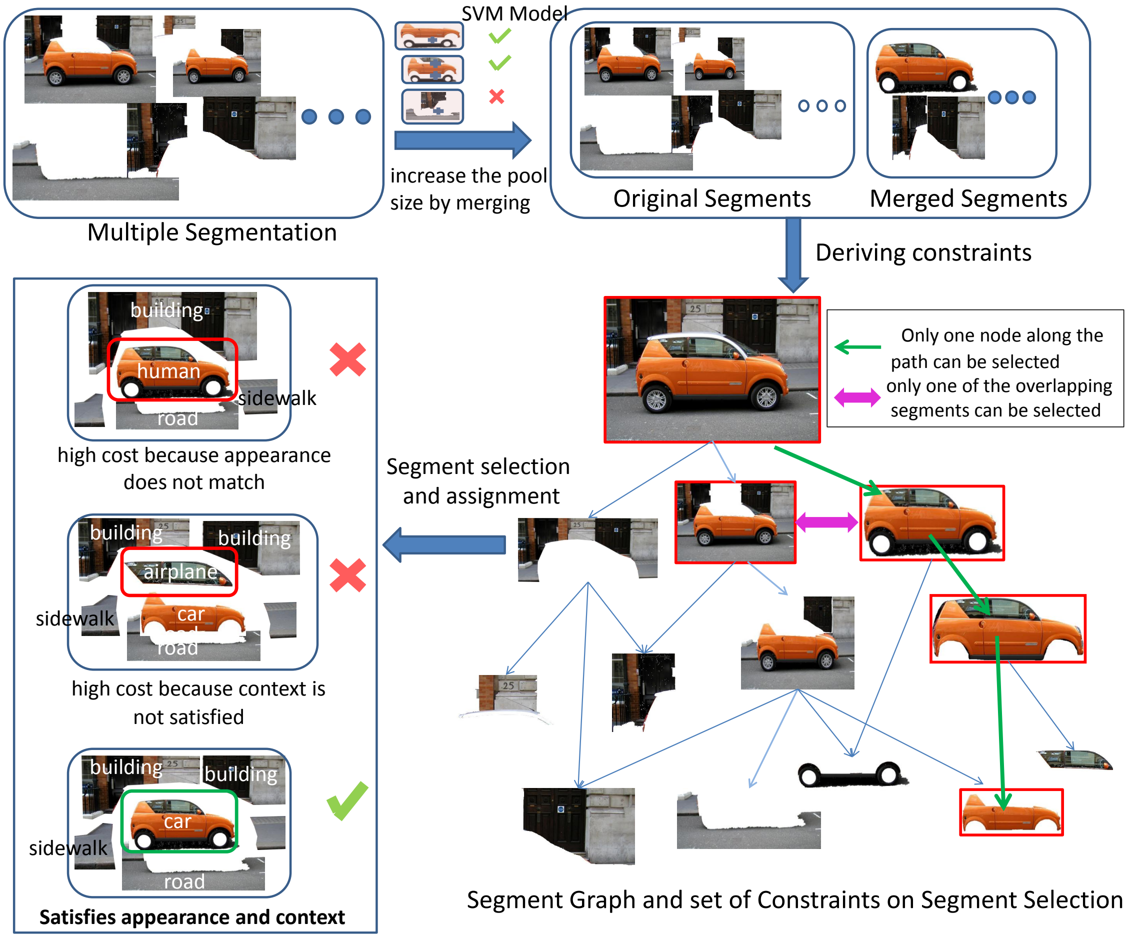
SVM Model
increase the pool size by merging
Multiple Segmentation
Original Segments
Merged Segments
Deriving constraints
Only one node along the path can be selected
only one of the overlapping segments can be selected
building
human
sidewalk
road
high cost because appearance does not match
building
building
airplane
sidewalk
car
road
high cost because context is not satisfied
building
building
car
sidewalk
road
Satisfies appearance and context
Segment selection and assignment
Segment Graph and set of Constraints on Segment Selection

Figure 2.3: Our approach: We first create a pool of segments using multiple segmentations of an image and merging some of the connected pairs and triples of these segments. These segments are arranged in a graph structure where path constraints are used to obtain selection constraints. An example of a path constraint is shown using green edges: only one segment amongst all the segments in the path can be selected. The magenta arrow shows that two segments which overlap cannot be selected simultaneously. Finally, the QP framework is used to find the set of segments, together with their labels, which minimizes the cost function given the constraints

2.5 Piecing together the Segments

Our goal is to select a set of segments from the pool such that each segment has high overlap with a ground-truth segment and is assigned its correct label.

We formulate a cost function which evaluates any possible selection and labeling of segments from the pool. Each segment, S_i in the pool is associated with a binary variable X^i which represents whether or not the segment is selected. With each selected segment we also associate a set of C binary variables, $(X^i_1 ... X^i_C)$, which indicates the label associated with the segment. $X^i_j = 1$ represents that segment i is labeled with class j. Our goal is to choose X^i such that the cost-function $\mathcal{J}$ is minimized, where $\mathcal{J}$ is defined as:

$$\mathcal{J} = \sum_{i,j} -w_1 A_{ij} X^i_j - \sum_{i} w_2 S_i X^i + \sum_{i,j} \sum_{k,l} w_3 X^i_j P_{ijkl} X^k_l \tag{2.1}$$

The cost function consist of three terms. The first term uses an appearance based classifier to match the appearance of selected segments with their assigned labels. The second term is the explanation reward term which rewards the selection of segments proportional to their size. The third term is a context satisfaction term which penalizes assignments which do not satisfy the contextual relationships learned from the training data. We discuss each of these terms below. The weight w_1,w_2,w_3 are obtained by cross validation on a small dataset and for our experiments we use 1, 1.5 and 0.5 respectively.

2.5.1 Constraints on Segment Selection

While there are 2^{N_S} possible selections (where N_S is the number of segments in the pool), not all subsets represent valid selections. For example, if segment i is selected and assigned label j, then other segments which overlap with segment i should not be selected to avoid multiple labeling of pixels. Figure 2.3 shows the overlap constraint by a magenta arrow where the two car segments which overlap cannot be chosen simultaneously. Similarly, two segments along a path from the root to any leaf node cannot be selected together. Figure 2.3 shows one such path constraint in green, where selection of the car and its subset segments simultaneously is prohibited.

These constraints are represented as follows:

$$0 \leq X^i + X^k \leq 1 \;\; \forall (i,k) \in O \tag{2.2}$$

$$0 \leq X^{p_1} + X^{p_2} .... X^{p_m} \leq 1 \;\; \forall p \in \mathcal{P} \tag{2.3}$$

where O represents the set of pairs of regions in the graph that overlap spatially and $\mathcal{P}$ represents the set of paths from the root to the leaves in the segment graph. Additional constraints that are enforced while minimizing the cost function $\mathcal{J}$ include:

$$0 \leq X^i \leq 1 \tag{2.4}$$

$$\sum_j X^i_j = X^i \tag{2.5}$$

These constraints allow only one label to be assigned to each selected segment.

2.5.2 Cost Function

We now explain the individual terms in the cost function.

Appearance Cost: The first term in the cost function evaluates how well the appearance of the selected segment i associated with label j matches the appearance model for class j. For computing A_{ij}, we learn an appearance model from training images using a discriminative classifier over visual features. We use the appearance features from [31] and learn a discriminative probabilistic-KNN model as in [32] for classification.

Explanation Reward: This term rewards selecting a segment proportional to its size, represented by S_i. This term avoids the trivial solution where no segment gets selected by the algorithm.

Contextual Cost: The third term evaluates the satisfaction of contextual relationships for a given selection of segments and their label assignment. We model context by pair-wise spatial and contextual relationships as in [14]. If segment i is assigned to class j and segment k is assigned to class l, P_{ijkl} measures the contextual compatibility based on co-occurrence statistics of classes j and l. We also evaluate spatial contextual compatibility by extracting the pairwise-differential features as in [14] for segments i and k and comparing them with a learned model of differential features for labels (j, l). For example, if the labeling is such that sky occurs below water then the penalty term is kept high and vice-versa. The penalty term is defined as:

$$P_{ijkl} = C_1 \exp(\frac{(d_{i,k} - \mu_{j,l})^2}{2\sigma_{j,l}^2}) + C_2 \exp(-\alpha M_{j,l}) \tag{2.6}$$

where C_1, C_2 and α are constants. $d_{i,k}$ is the differential feature between segment i

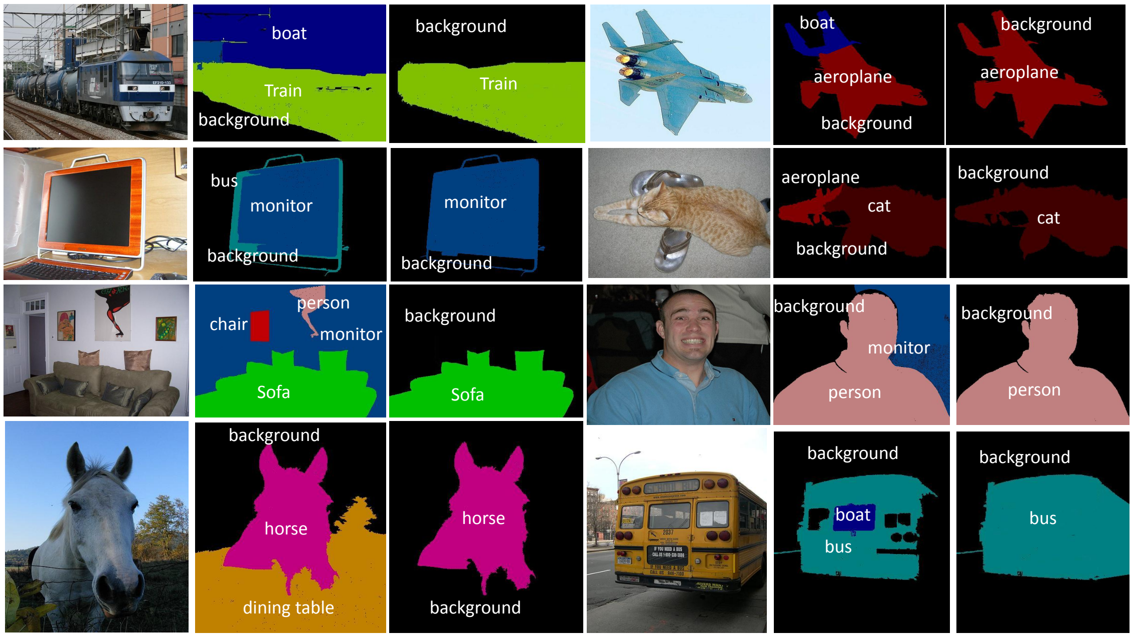

Figure 2.4: PASCAL VOC'09 labeling results. Columns (a) and (d) - original images. Columns (b) and (e) show the performance of appearance based approach without context. Columns (c) and (f) show the performance of our algorithm with context. Best viewed in color.

and segment k. $\mu_{j,l}$ is the mean differential feature obtained from training between class labels j and l. The term $M_{j,l}$ represents the co-occurrence of classes j and l, also obtained from training. We employ eight differential features - Δx,Δy,$\Delta \mu_{red}$,$\Delta \mu_{green}$,$\Delta \mu_{blue}$,$\Delta \mu_{brighter}$, adjacency and overlap.

2.5.3 Optimization

For optimizing the cost function, we relax the binary variables X^i and X^i_j to lie in [0, 1]. We use the Integer Projected Fixed Point (IPFP) algorithm [33] to minimize the cost function.

The solution generally converges in 5-10 steps, which makes it very efficient, while outperforming current state-of-the-art methods for inference. IPFP solves quadratic optimization functions of the form:

$$x'^{*} = argmax(x'^{T} M x') \;\; s.t. \, Ax' = 1, \; x' \geq 0 \tag{2.7}$$

To use the IPFP algorithm, we transform the original equation 1 into 7 through the following substitution: $x' = \left(\begin{smallmatrix} 1 \\ X \end{smallmatrix} \right)$ and $M = \left(\begin{smallmatrix} 0 & (A+S)^T/2 \\ (A+S)/2 & -P \end{smallmatrix} \right)$. The path constraints discussed in section 5.1 are incorporated as constraints in a linear solver during step 2 of the optimization algorithm.

In the second step, the relaxed solution is then discretized to obtain an approximate solution. Here, higher probability segments are selected first and assigned their class labels as long as segment selection constraints are satisfied.

2.6 Experiments

We evaluated the performance of our algorithm on three standard dataset: Label Me subset (used in [16]), PASCAL VOC 2009 [34] and MSRC [11].

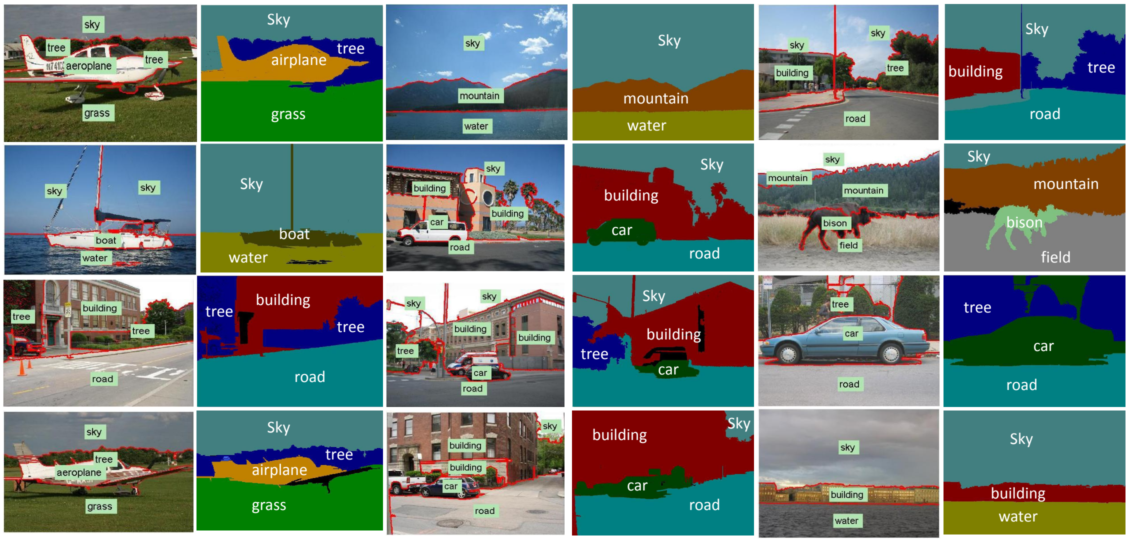

Figure 2.5: LabelMe dataset results - columns 1, 3 and 5 show the original image with object labels obtained by our algorithm and columns 2, 4 and 6 show the corresponding image segmentation.

Table 2.1: Performance comparison of our algorithm against previous approaches on PASCAL VOC09 dataset.

	Hierarchical CRF [23]	Hierarchical CRF with CO [23]	Ours w/o Context w/Merging	Ours w/Context w/o Merging	Ours Context w/Merging
Background	77.7	82.3	76.4	61.2	**85.8**
Aeroplane	38.3	**49.3**	25.6	37.3	39.8
Bike	9.6	**11.8**	8.0	5.5	7.6
Bird	**24.0**	19.3	14.2	20.6	18.4
Boat	35.8	37.7	**47.3**	36.0	45.0
Bottle	**31.0**	30.8	8.1	14.6	8.4
Bus	59.2	**63.2**	30.5	30.8	44.6
Car	36.5	46.0	53.7	55.3	**66.1**
Cat	21.2	23.7	50.1	46.8	**54.2**
Chair	8.3	10.0	**18.6**	10.6	11.2
Cow	1.7	0.5	9.1	4.2	**10.3**
Table	22.7	23.1	48.5	40.2	**52.7**
Dog	14.3	14.1	10.9	11.3	**15.2**
Horse	17.0	22.4	15.8	17.3	**23.5**
MBike	26.7	33.9	33.8	29.0	**39.2**
Person	21.1	35.7	47.3	36.1	**50.8**
Plant	15.5	**18.4**	10.2	9.1	11.5
Sheep	16.3	12.1	15.7	29.3	**31.5**
Sofa	14.6	**22.5**	11.2	12.8	19.8
Train	48.5	**53.1**	48.6	47.4	40.4
TV	33.1	37.5	35.2	38.2	**48.9**
Average	27.3	30.8	29.5	28.3	**34.5**

Table 2.2: Performance comparison of our algorithm against other approaches on LabelMe dataset.

	Texton-boost	**MRF based**	**Jain et.al. [16]**	**Ours(no Context,Merging)**	**Ours(Context,no Merging)**	**Ours(Context, Merging)**
pixel wise	49.75	54.2	59.0	65.23	71.9	75.6
class wise	20	30.2	–	38.5	43.5	45

LABEL-ME: [16] used a subset of LABEL ME containing 350 images - 250 training and 100 testing. The dataset contains 19 classes. Performance is measured using the two standard measures from [16]. For comparison, we also evaluate four approaches in addition to those compared in [16] (1) Our multiple segmentation framework, but without contextual information. (2) A fully connected MRF-model similar to [15], which performs recognition using context on a fixed segmentation obtained using stability analysis. (3) A Texton-boost approach [3] without the CRF model, and 4) our method applied to the initial segment pool, but without the SVR merged segments.

Figure 2.5 shows a few qualitative examples of our approach. When context is not utilized many small segments are mislabeled and matched to wrong object classes. However, when context is added many of these errors are eliminated.

Table 2.2 shows the quantitative performance of our approach compared with these four methods and [16] using the two standard evaluation metrics. Our approach has a pixel-wise accuracy of 75.6%; when only appearance is used the performance falls to 65.23%. This shows that contextual information is critical not only for recognition but also for segment selection. As expected, the fixed segmentation MRF model has a low pixel-wise accuracy of 54.2%. The publicly available version of Texton-boost achieves just 49% pixel-wise accuracy. This is because Texton-boost relies on pixel-based appearance models. These are adequate for modeling regions like ‘grass’ and ‘sky’ but perform poorly for objects whose recognition requires cues such as shape.

[3] http://jamie.shotton.org/work/code/

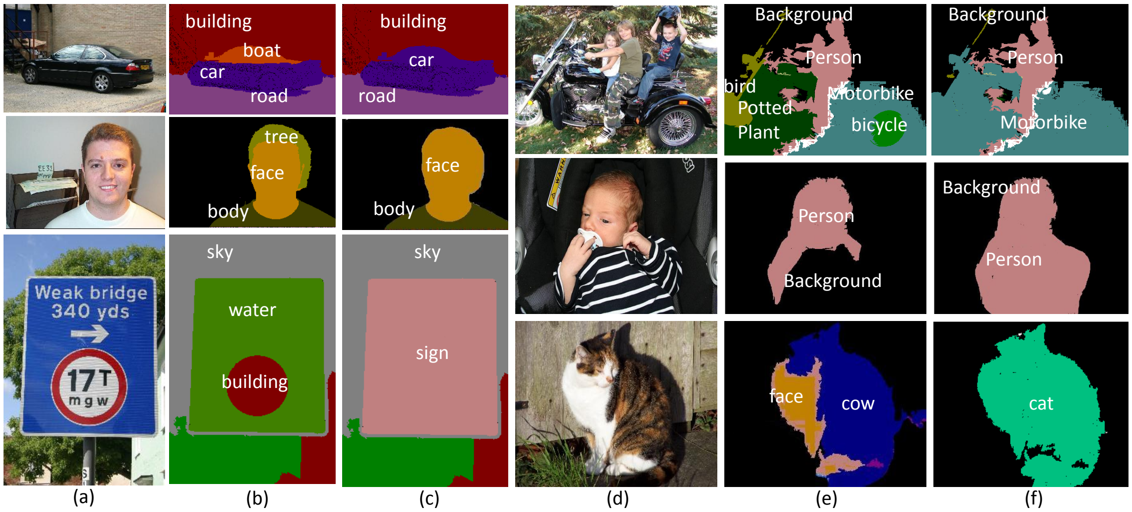

Figure 2.6: Qualitative results of our algorithm with and without merging. Columns (a) and (d) are original images. Columns (b) and (e) show the labeling performance without merging. Columns (c) and (f) show performance with merging. Best viewed in color.

PASCAL VOC 2009: The PASCAL VOC 2009 dataset [34] consists of 1499 images which is split into 749 images for training and 750 images for validation. We follow the protocol used by [23] to compare against the state of the art, and use the same evaluation metric as [23]. Table 5.1 shows the class wise performance of our approach compared with the other approaches. Our approach outperforms previous approaches on many classes which shows that it generalizes to a large number of object classes. Our better performance on classes like Car, Cat, Horse, Sheep, Cow, Monitor, Dog and Person supports our contention that a multiple segmentation approach performs better on object classes for which shape is important. Table 5.1 also shows that both context and merging improves recognition by choosing segments which have better spatial support.

Figure 2.4 shows some qualitative results on VOC 2009. Columns (b) and (e) show the labeling performance of our algorithm solely based on appearance. The algorithm using only appearance leads to a variety of errors such as the wing of the airplane being labeled as boat, the ground in the horse image as dining table, and the painting above the sofa as a person. Columns (c) and (f) show the performance of our approach with context. Figure 2.6 compares qualitative results of our algorithm with and without mergings and elucidates the importance of merging for better recognition. For example, in the sign image, the parts of the sign board are labeled as water and building but after merging them, it is correctly labeled as sign board.

MSRC dataset: Our algorithm achieved 75% (pixel-wise) and 68.7%(classwise) on the MSRC dataset, which is comparable to state-of-the-art results except [23]. MSRC is relatively simple and does not significantly benefit from the use of multiple segmentations. Our approach performs better than [23] for classes like bird, car and cow, where multiple segmentation and merging helps by creating segments whose shapes are closer to class models, but performs poorer on “stuff” classes such as grass and sky.

2.7 Conclusion

We described an approach for simultaneous segmentation and labeling of images using appearance and context. The optimization criteria developed was solved by relaxing the discrete constraints and employing a quadratic programming method. The relaxed solution was then discretized (and additional constraints were introduced) using a greedy algorithm. Experiments on three well studied datasets demonstrated the advantages of the method.

Chapter 3: Object Co-Labeling in Multiple Images

3.1 Introduction

With the recent surge in photos and videos taken from hand-held devices and those shared online, many of which are taken of the same scenes, the need to automatically label objects in such image sets has emerged. Traditional approaches to recognition typically consider only a single test image [35]. However, due to variation in data distribution, occlusion and viewpoint change, object models may not always capture the appearance of objects and ambiguity arises. Recently, context information has also been modeled to capture relationships between objects at the semantic level to reduce such ambiguities [28, 36, 37]. However, modeling relationship between objects is difficult as they are also viewpoint dependent and do not generalize well [16].

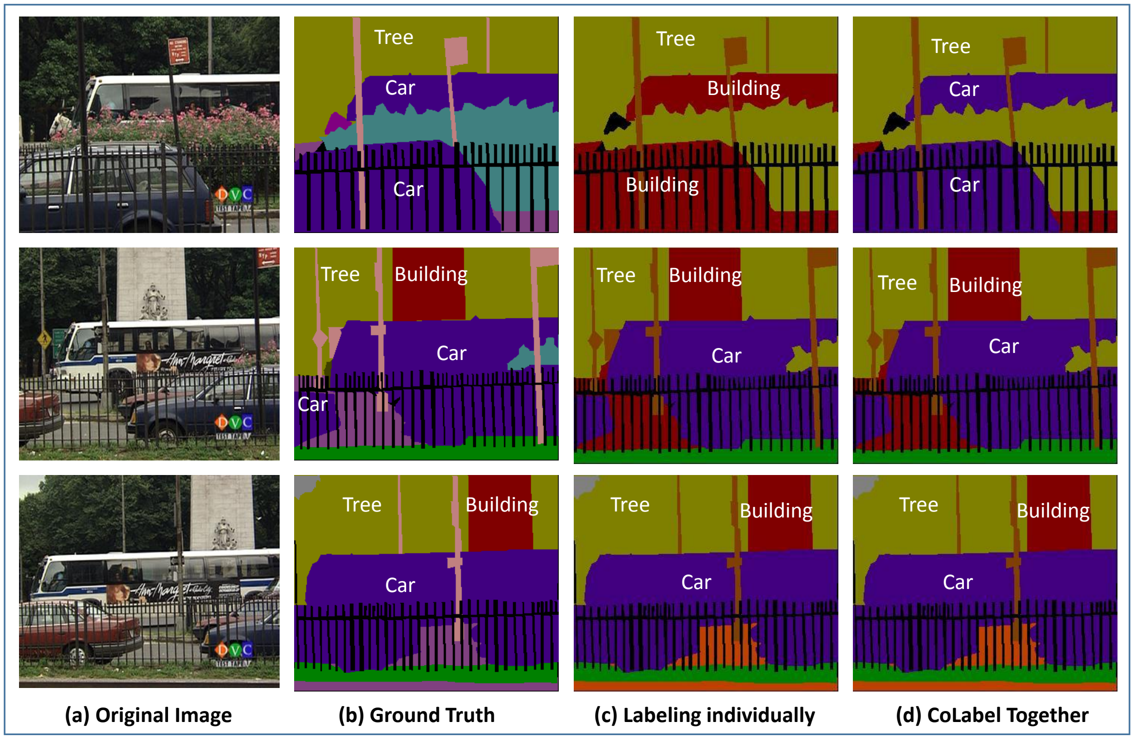

Figure 3.1: Object Co-labeling for multiple images. Column (a) shows the original images taken of the same scene, (b) is the groundtruth labeling, (c) the results of single image parsing and (d) our co-labeling framework. Given images of the same scene, object co-labeling can correctly label the major objects even when appearance models failed to parse images.

Consider the example in Figure 3.1. These images of vehicles are taken of the same scene but at different times. Due to occlusion, the learned vehicle model does not classify vehicles in all the frames correctly. However, if we consider these images together, we find that they share similar appearance and space-time consistency with the surrounding objects and background. So, even though a region in one image may not look like a car to the car detector, it might be visually similar to the region in another image corresponding to the same car that the car detector responds to strongly . This example explains the motivation of our work. We are trying to answer the question - “Can we do better inference using information from other images of the objects in related scenes?” We introduce a new problem that we call *object co-labeling*. Given a set of images of the objects in the same scene, the goal of co-labeling is to locate and pixel-wise label the principal objects such as car, building and road in all images. We will demonstrate that our framework generalizes well to other similar applications, such as label propagation in videos and semantic segmentation and annotation in similar scenes.

We propose an approach to select the best segmentation and labeling in a single optimization procedure that utilizes low-level information across all the images to perform segment selection and object labeling coherently. We build on the multi-segmentation frame work proposed in [38]. We first segment the images; to overcome the fragmentation problem, we allow connected segments to be merged based on local color, texture and edge properties. We then include the mid-level cues to constrain the solution space - for example, the segment merging step leads to overlapping segments, and we restrict global solutions to exclude overlapping segments (avoiding the possibility of multiple labeling for pixels). By incorporating label coherence between region pairs with low level correspondence determined by SIFT flow [39], we find the subset of segments that best explains the images. For example, in Figure 3.1, the bus in the first image is labeled as “Building” when labeled in isolation. However, due to its strong correspondence with other bus segments in other images of the scene, it is correctly labeled as bus by our approach.

The contributions of our paper are: (a) a general framework for object co-labeling which allows us to segment and pixel-wise label objects from multiple images; (b) a novel approach to co-label objects in a multiple segmentation framework in multiple images, (c) a novel objective function that can be optimized efficiently to perform segment selection and co-labeling across all images in the same scene . (d) a co-labeling framework that can be generalized to other recognition tasks such as label propagation in video sequences and semantic segmentation and labeling of object categories in similar scenes.

3.2 Related Work

Our paper is related to and inspired by several other problems in computer vision.

Scene alignment: One of the basic problems of computer vision in multiple images is scene alignment. Optical flow is proposed in [40] for the correspondence problem between two adjacent frames in dynamic scenes in video sequences. It is a dense sampling in the temporal domain to align temporally continuous frames. In order to cope with more general scene matching problems, [39] proposed SIFT Flow. It establishes dense correspondence of SIFT descriptors in different images by using discontinuity-preserving optical flow. This method provides a robust low-level correspondence for images and is shown to outperform traditional optical flow algorithms. But it is based on a pixel-level matching that can not capture the object level information, and thus only works when two images are very similar.

Image Parsing: Many approaches have been proposed to annotate image for scene understanding. A common pipeline is to first segment the image and then infer labels by adding contextual relations between segments [16, 36, 41]. To overcome fixed segmentation issues, joint segmentation and recognition frameworks have been proposed [20, 38, 42]. [43] proposed a framework to label road scenes, where they learn models from limited training data and adapt to new scenes during testing. Some approaches bypass the segmentation step and directly transfer labels from training data. [44] further extends SIFT Flow for non-parametric scene parsing by retrieving images of similar scenes in the training set and transfer the labels to the query in a Markov Random Field. However, they need to retrieve over 20 training images per query, and only work well when the retrieved training images are similar to the query. [45] proposed a non-parametric image parsing algorithm that also doesn't require training. They transfer information at the superpixel level using complex features.

Label Propagation: The goal of label propagation is to automatically annotate the video given a few annotated frames. [46] proposed an Expectation Maximization (EM) based approach to automatically propagates labels through frames. [47] uses a weighted combination of motion, appearance and spatial continuity evidence to propagate labels in frames and uses graph cut to minimize the energy. However, all these approaches assume temporal consistency and small object motion which is not true in our case. [48] proposed an approach that first classifies images then propagates labels based on a similarity metric, but their goal is medical image classification. Our problem is more general as it requires annotating images which do not share temporal consistency and we are annotating multi-class pixel labels in complex scenes.

Cosegmentation Our work is inspired by some of the recent works in "co-segmentation" [49, 50]. In co-segmentation, the goal is to automatically segment images of similar scenes in a joint manner. [51] extended the cosegmentation approach to multi-class segmentation. We extend this problem of co-segmenting images to co-labeling images. Th multiple foreground co-segmentation (MFC) problem studied by [52], [53], [54] and [55] is similar in spirit where the goal is to segment K foreground objects in M images. In these works, objects (girl and baby) belonging to same semantic class (person) can be labeled as different foregrounds in MFC problem. So cosegmentation does not address the problem of labeling the foreground semantically. Moreover, the goal of co-segmentation problem is not scene understanding. Our problem is dense pixel labeling and scene understanding, where we model the relationships between semantic objects.

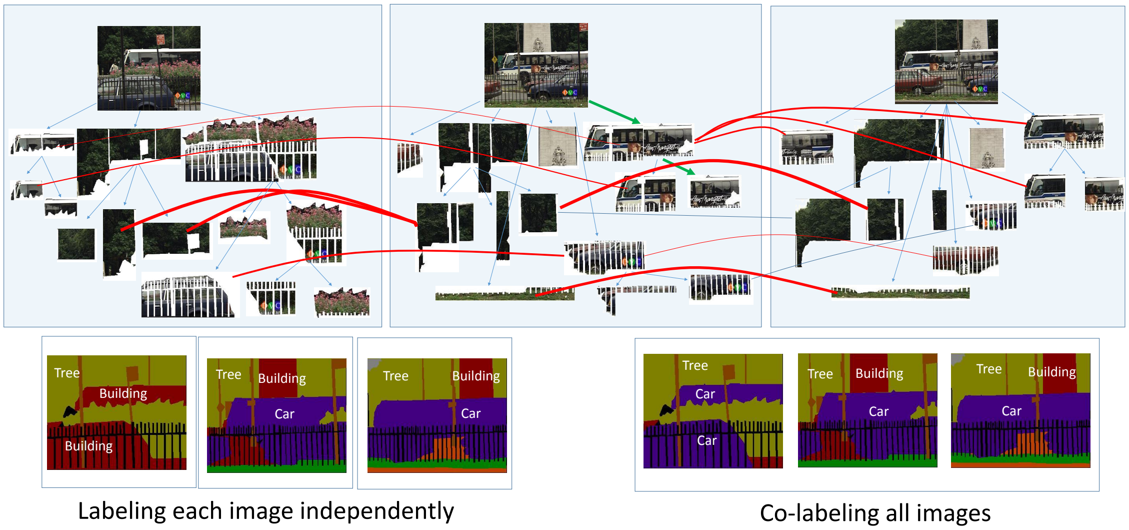

Figure 3.2: Our approach: We first create a pool of segments using multiple segmentations for each image. These segments are arranged in a graph structure where path constraints are used to obtain selection constraints. An example of a path constraint is shown using green edges: only one segment amongst all the segments in the path can be selected. We then define a co-labeling consistency cost based on the strength of SIFT flow connection between segments in different images, shown as the red edges in the figure, where the width of each edge denotes the SIFT flow similarity. Finally, a QP framework is used to find the set of segments, together with their labels, which minimizes the cost function given the constraints

3.3 Overview

Our framework is shown in Figure 3.2. Our approach starts by constructing a pool of multiple segments for each image. Our multiple segmentation approach, similar to [56], constructs a pool of initial segments by varying the controlling parameters of a segmentation algorithm or by starting from a coarse segmentation and iteratively refining the segmentation by merging or further segmenting initial segments. In contrast to [56] where segments are manually chosen to merge, we construct a good set of mergings [38] using a classifier which rejects combinations which are unlikely to correspond to complete objects (section 3.4). The final segment graph is organized in a hierarchical manner to impose constraints for selection.

Given the segment graph, we then compute the pairwise low level correspondences between each pair of segments in different images using SIFT flow. The two images may contain different object instances captured from different viewpoints, placed at different spatial locations or may be taken at different scales. In addition, some objects present in one image might be missing in other images. Thus, it is suitable for our co-label task to establish correspondences between different segments. However, to our knowledge, there are very few works to use SIFT flow to establish superpixel level correspondence. We first compute SIFT flow between each pair of images. Then the correspondence between each pair of segments in the two images is estimated as the ratio of number of pixels in one segment flowing to the other.

After computing the low level correspondence graph, we formulate a cost function which accounts for local appearance and enforces pairwise consistency of segments between the images. Directly optimizing this cost function is NP hard. Therefore, the cost function is approximately minimized by first relaxing the selection problem. The relaxed problem is solved efficiently by quadratic programming (QP). The relaxed solution is then discretized to obtain the final labeled segmentation (section 3.5.2). Finally, we evaluate the performance of our approach with previously reported methods (section 4.5).

3.4 Constructing the Segment Graph

We used the hierarchical segmentation algorithm from [57] to construct the segment pool. We then learn a merging function as described in [38] to obtain a better pool of segments using color, texture and edge features similar to those used by Hoiem et. al. [31] for each segment of an object. This learned merging function improves the spatial support of our segments and now the goal is to select subset of segments which are consistent across images. We organize the segments of each image into a hierarchical segment graph for recognition. The graph structure allows us to impose constraints that reduce the combinatorics of the search process - for example, that a solution cannot include overlapping segments, since this could lead to pixels being given multiple labels. Pairwise constraints on selection of segments are computed given the segment graph. The path constraints in the segment graph hierarchy guarantee that each pixel in the images is labeled once and only once.

3.5 Colabeling segments

In the co-labeling stage, the goal is to select and label segments from all the images at the same time. Given a pool of segments from all images and the hierarchical graph of multiple segmentations for each image, our goal is to select a set of segments from the pool such that each segment has high overlap with a ground-truth segment and infer the best labels that are consistent across all images.

We formulate a cost function which evaluates the subset selection and labeling of segments from the pool. Given M test images $\mathcal{I} = I_1, ..., I_M$, each segment, S_i in the pool is associated with a binary variable X^i which represents whether or not the segment is selected. With each selected segment we also associate a set of C binary variables, $(X^i_1...X^i_C)$, where $X^i_j = 1$ represents that segment i is labeled with class j. Our goal is to choose X^i such that the cost-function $\mathcal{J}$ is minimized, where $\mathcal{J}$ is defined as:

$$\mathcal{J} = -w_1 \sum_{I_n} \sum_{i,j} A_{ij} X^i_j + w_2 \sum_{I_n, I_m} \sum_{i,j} \sum_{k,l} X^i_j P_{ijkl} X^k_l \tag{3.1}$$

where $I_n, I_m \in \mathcal{I}, m \neq n$ and $S_i \in I_n$, $S_k \in I_m$.

The cost function consist of two terms. The first term uses an appearance based classifier to match the appearance of selected segments with their assigned labels. The second term is a label consistency constraint which gives high penalty to the segments in two images that do not have a strong low level connection. We discuss each of these terms below. The weights w_1 and w_2 are obtained by cross validation on a small dataset and for our experiments we use 1 and 0.1 respectively.

3.5.1 Constraints on Segment Selection

While there are 2^{N_S} possible selections (where N_S is the number of segments in the pool), not all subsets represent valid selections. For example, if segment i is selected and assigned label j, then other segments which overlap with segment i should not be selected to avoid multiple labeling of pixels. Similarly, two segments along a path from the root to any leaf node cannot be selected together. Figure 3.2 shows one such path constraint in green, where selection of the bus and its subset segments simultaneously is prohibited.

These constraints are represented as follows:

$$0 \leq X^i + X^k \leq 1 \;\; \forall(i,k) \in O_n \tag{3.2}$$

$$0 \leq X^{p_1} + X^{p_2} .... X^{p_m} \leq 1 \;\; \forall p \in \mathcal{P}_n \tag{3.3}$$

where O_n represents the set of pairs of regions in the graph that overlap spatially and $\mathcal{P}_n$ represents the set of paths from the root to the leaves in the segment graph in image $I_n \in \mathcal{I}$. Additional constraints that are enforced while minimizing the cost function $\mathcal{J}$ include:

$$0 \leq X^i \leq 1 \tag{3.4}$$

$$\sum_j X^i_j = X^i \tag{3.5}$$

These constraints allow only one label to be assigned to each selected segment.

3.5.2 Cost Function

We now explain the individual terms in the cost function.

Table 3.1: Our superpixel features

Feature Descriptors		
Type	Name	Dimension
Color	RGB	3
	HSV values	3
	Hue	6
	Saturation	4
Texture	DOOG filters and stats	15
	Texture Histogram	100×2
Shape and Location	Normalized x and y	8
	Bounding box size relative to image size	2
	Segment size ratio to the area of the image	1
SIFT	SIFT Histogram	100×2

Appearance Cost: The first term in the cost function evaluates how well the appearance of the selected segment i associated with label j matches the appearance model for class j. For computing A_{ij}, we learn an appearance model from training images using a discriminative classifier over visual features. We use the appearance features for superpixels from [45] and learn a discriminative probabilistic-KNN model as in [32, 58] for classification.

Consistency Cost: The second term evaluates the satisfaction of label consistency between segments in different images. Given the SIFT flow of each pair of images in the test set, we can obtain a correspondence strength for each pair of segments between them. Then we assign a cost according to these edge strengths between segments, giving high penalty to those with weak correspondence and reward to those with strong correspondence. The penalty term is defined as:

$$P_{ijkl} = \exp(\frac{-\alpha\phi(S_i, S_k)^2}{2\sigma^2}) \tag{3.6}$$

where α and σ are constants ($\alpha = 0.05$ and $\sigma = 0.5$ in our experiments). $\phi(S_i, S_k)$ is the low level similarity between segment S_i and segment S_k in two different images. We estimate the SIFT flow similarity between superpixels in the following way. Let $f_{S_i \mapsto S_k} : \mathbb{R}^2 \mapsto \mathbb{R}^2$ be the SIFT flow from S_i to S_k, then ϕ is defined as the ratio of the number of pixels in S_i flowing to S_k:

$$\phi(S_i, S_k) = \frac{||f_{S_i \mapsto S_k}(S_i) \cap S_k||_0}{\max\{||S_i||_0, ||S_k||_0\}} \tag{3.7}$$

3.5.3 Optimization

To optimize the cost function, we relax the binary variables X^i and X^i_j to lie in [0, 1] and use the Integer Projected Fixed Point (IPFP) algorithm [33] to minimize the cost function. The solution generally converges in 10-15 steps which is reasonable for the problem size. IPFP solves quadratic optimization functions of the form:

$$x'^{*} = \arg\max(x'^{T} M x') \;\; s.t. \, Ax' = 1, \; x' \geq 0 \tag{3.8}$$

To use the IPFP algorithm, we transform the original equation 1 into 8 through the following substitution: $x' = \left(\begin{smallmatrix} 1 \\ X \end{smallmatrix} \right)$ and $M = \left(\begin{smallmatrix} 0 & A^T/2 \\ A/2 & -P \end{smallmatrix} \right)$. The path constraints discussed in section 5.1 are incorporated as constraints in a linear solver during step 2 of the optimization algorithm. In the second step, the relaxed solution is discretized to obtain an approximate solution. Here, higher probability segments are selected first and assigned class labels as long as segment selection constraints are satisfied.

The optimization scheme above is efficient for inference. The bottleneck of the running time is the calculation of SIFT flow dense matching, which takes approximately 3 to 5 seconds for each pair of images of size 640×320. Once the SIFT flow is precomputed, the batch inference of about 100 images in a sequence takes about 3 seconds with our Matlab implementation.

3.6 Experiments

We evaluate our co-labeling algorithm on three tasks: co-labeling objects in the same scene, where we train the appearance model on fully-annotated multiclass training image sets then test on images of the same scenes; object segmentation and recognition in similar scenes, where the objects in each subset are in the same categories and in similar but different scenes; and multiclass label propagation in video sequences, in which the labels of one or two frames are given and the labels are propagated to the rest of the frames in video sequences. In these experiments, we use the combination of features from [45] and [31]. Table 3.1 shows our features setting in detail.

3.6.1 Co-label Objects in the Same Scenes

In this task, we use the SUNY Buffalo 24-class Dataset [47], one of the multiclass video pixel label propagation benchmarks, to evaluate our co-labeling algorithm. This dataset contains 8 video clips with 70 to 88 frames each, with pixel-wise labeled groundtruth. Each clip is taken in one scene with either the camera or the objects moving. To show that our algorithm can work without temporal adjacency, we evenly sampled 20 frames in each video to form our test data. Tabel 3.2 shows our colabel results on the 8 subsets. We compare our results with the label propagation algorithm in [47] and show improved results compared to theirs. Figure 3.3 and Table 3.2 shows some qualitative and quantitative results on this data. Our colabeling algorithm achieves better performance in 7 out of 8 subsets, and outperforms the benchmark method in global and classwise measures. Moreover, for all subsets, our colabeling algorithm outperforms the accuracy of single image parsing, showing that co-labeling improves label accuracy for objects in the same scene.

	Bus	Container	Garden	Ice	Paris	Salesman	Soccer	Stefan	Global
[47]	35.86	76.66	66.09	60.84	67.98	**77.95**	84.15	59.93	63.46
Single label	70.14	83.16	69.68	88.91	61.76	70.66	82.49	85.01	79.86
Colabel	**75.75**	**89.97**	**74.24**	**90.41**	**68.52**	75.68	**87.43**	**90.04**	**84.33**

Table 3.2: Quantitative results of colabeling objects in the same scene on SUNY Buffalo dataset, compared with the method in [47].

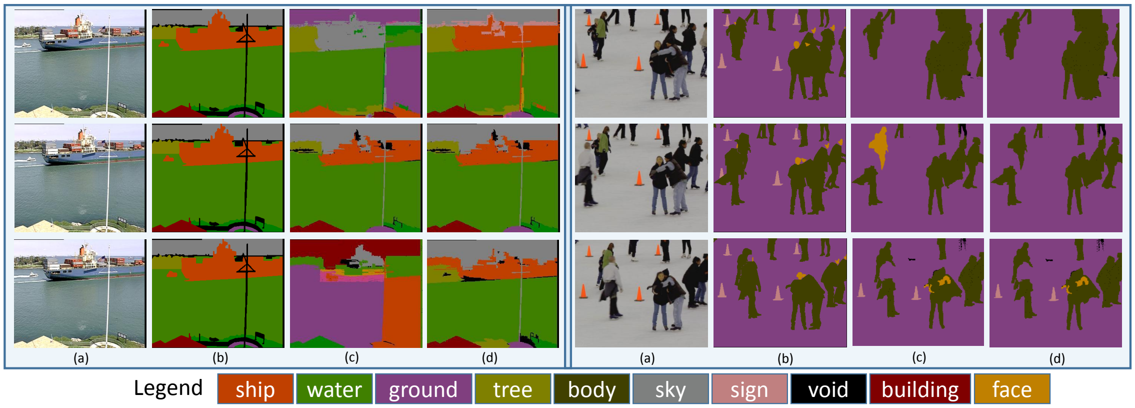

Figure 3.3: Qualitative labeling results of two subsets in SUNY Buffalo datasets. Columns (a) to (d) correspond to the original image, groundtruth labeling, single image parsing results and co-labeled results. Best viewed in color.

	Bus	Container	Garden	Ice	Paris	Salesman	Soccer	Stefan	Global
[47]	40.59	**94.7**	64.12	60.04	**67.95**	**76.75**	82.25	59.59	66.7471
Single label	68.74	81.03	67.89	74.47	60.73	64.77	83.21	85.44	75.52
Colabel	**72.61**	86.15	**70.23**	**81.68**	66.49	72.76	**88.08**	**89.13**	**80.36**

Table 3.3: Quantitative results of video label propagation on SUNY Buffalo dataset, compared with the method in [47].

	25 Frames	50 Frames	100 Frames
Single Label	76.99	76.24	73.11
Label Propagation using Colabel	**81.76**	**81.33**	**77.40**

Table 3.4: Performance of label propagation in CamSeq01.

3.6.2 Label Propagation in Video Sequences

Our co-labeling framework can be applied to label propagation in video sequences. In this task, the labels of the first two frames are given for each video sequence, then propagated to the remaining frames. Instead of using a fully-connected pairwise cost, we only have edges between adjacent frames in video label propagation. We test on two video datasets with semantic pixel labels. The first is the SUNY Buffalo dataset we used for colabeling. The second is the CamSeq01, a 101-frame sequence from the CamVid dataset [59]. Table 3.3 and Table 3.4 show quantitative results on each dataset. We can see that even without explicitly modeling temporal consistency, our co-labeling algorithm still outperforms the baseline video label propagation algorithm on both datasets.

3.6.3 Semantic Segmentation in Similar Scenes

Our co-labeling algorithm is also capable of labeling objects of the same semantic category in different but similar scenes. In this experiment, we test on a subset of the MSRC 21-class dataset [11]. The data is divided in a standard train-test split, but we further divide the original 21 categories in test sets into finer subsets that shares a similar scene type to form our MSRC Co-label dataset (See Figure 3.4). Table 5.3 and Figure 3.4 shows the quantitative and qualitative results of our framework, compared to labeling each image individually. This shows that our co-labeling algorithm works not only for objects in the same scene but can also generalize to object segmentation and recognition in different but similar scenes in a challenging multi-class multi-object dataset.

We compare our supervised joint object segmentation and recognition in multiple images of similar scene type with the recent work in [51]. They proposed an approach for weakly supervised multiclass cosegmentation, where the images of the same object categories (and mostly share a similar scene) are jointly segmented and classified given weak labels. We evaluate the performance of [51] in the fully supervised multi-class segmentation and classification task on our MSRC co-labeling dataset. We first cosegment image sets in similar scenes using the settings in [51] then perform multi-class recognition using the same feature as in Table 3.1. We tried different values of K and the results are in Table 3.6. We can see that our co-labeling algorithm outperforms their performance in the object co-labeling task. Moreover, [51] needs users to provide semantic class labels of the test set of images. In contrast users in our approach just need to feed in a collection of test images of similar scenes without the need to provide class labels. So our approach

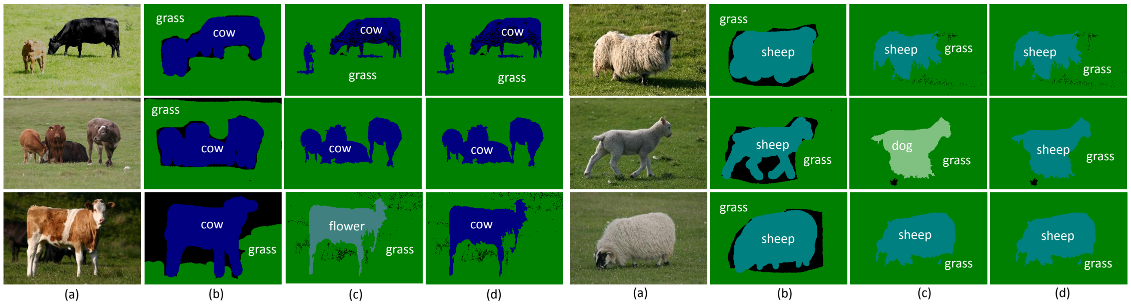

Figure 3.4: Qualitative labeling results of two subsets in our MSRC co-label dataset. Columns (a) to (d) correspond to the original image, groundtruth labeling, single image parsing results and co-labeled results. Best viewed in color.

requires less user interactions while achieving higher pixelwise accuracy compared to that in [51].

	Single Label	Colabel
grass	92.2	92.2
cow	42.5	**58.5**
sky	**99.5**	99.1
house	60.0	**66.3**
tree	**77.9**	64.3
sheep	46.1	**56.3**
flower	43.0	**55.9**
ground	**82.3**	78.8
book	51.0	**81.1**
dog	25.7	**40.6**
body	**38.5**	35.4
head	50.3	**58.0**
car	**58.3**	55.4
bike	69.3	**72.7**
plane	35.3	**39.0**
global	67.6	**72.1**
classwise	58.1	**63.6**

Table 3.5: Quantitative results of colabeling objects in our MSRC Co-labeling datasets.

	[51],$K = 4$	[51],$K = 6$	Colabel
grass	81.9	84.1	**92.2**
cow	15.0	13.2	**58.5**
sky	97.0	95.5	**99.1**
house	36.5	31.2	**66.3**
tree	27.7	50.1	**64.3**
sheep	9.1	22.2	**56.3**
flower	34.9	30.1	**55.9**
ground	25.0	78.7	**78.8**
book	33.5	35.9	**81.1**
dog	0.0	16.6	**40.6**
body	3.4	**46.6**	35.4
head	0.1	45.8	**58.0**
car	8.2	17.4	**55.4**
bike	7.0	14.4	**72.7**
plane	3.2	25.6	**39.0**
global	45.8	51.1	**72.1**
classwise	25.5	40.5	**63.6**

Table 3.6: Quantitative results of colabeling objects in our MSRC Co-labeling datasets, compared with [51]

3.7 Conclusion

We addressed the problem of object co-labeling, in which we aim to segment and label multiple images of the same (or similar) scene(s) joinly. We propose a framework that can jointly perform segment selection and labeling using appearance and low level SIFT flow correspondence. The optimization criteria developed was solved by relaxing the discrete constraints and employing a quadratic programming method. Experiments on three well studied datasets demonstrated the advantages of the method.

Chapter 4: Searching for Objects with Information Gain-based Twenty Questions Strategy

4.1 Introduction

Object detection and segmentation in complex scenes is a central and challenging problem in computer vision. Given an image, for example, Figure 5.1, our goal is to answer the query: is there a car in the scene, and if yes, to locate it with a bounding box or pixel-wise labels. This problem is usually tackled by running multiple object detectors exhaustively on densely sampled sliding windows [60] or category-independent object proposals [61–63]. Such methods are time-consuming since they need to evaluate a large number of object hypotheses. In addition, due to variations in data distribution, occlusion and viewpoint change, object models may not always capture the appearance of objects and ambiguity arises. In the example of Figure 5.1, since the viewpoint and the scale of the cars are not similar to those in common training images, it is difficult for the car detector to recognize and locate them.

Instead of checking all hypotheses indiscriminately and exhaustively, humans only look for a set of related objects in a given context. Context information has been modeled to capture relationships between objects at the semantic level to reduce ambiguities from unreliable detection results [64–66]. For example, because roads and buildings often co-occur with cars, knowing the existence of these objects can help us infer the locations of cars. Usually, context-sensitive methods consider relations of the query object with all other object classes in the scene at the same time. This is highly inefficient since many non-informative contextual objects have to be queried. For instance, in Figure 5.1, knowing the top of the scene is sky is not very helpful to distinguish whether there is a car or a boat since both can be under the sky; while observing a road instead of water in the lower part gives a strong indication of the existence of cars. And if we know there is road, we do not need to ask about water. We note that the set of related object classes and the order of asking questions about them is dynamic given a specific query in the scene and knowledge of previous observations. This motivates us to raise the question: can we utilize context information to locate query objects more accurately and more efficiently?

In this chapter, we propose a context-driven strategy to sequentially and dynamically select the most informative object class to detect, and outputs detection result of the query object when evidence from the context is strong enough. Our detector selection strategy adapts to different queries and scenes. Specifically, at each step, we make a decision about which detector to run based on responses from previous object detectors and contextual classifiers. Our spatial-aware contextual model then refines the search area for the query object and updates posterior probabilities for each class. This process is iterated until there is little gain in further evaluating new detectors. Finally, we run the query object detector in the output search area and use a unified probabilistic model to combine the result with previously obtained contextual information. To the best of our knowledge, this work is the first to handle the challenging task of simultaneous object detection and

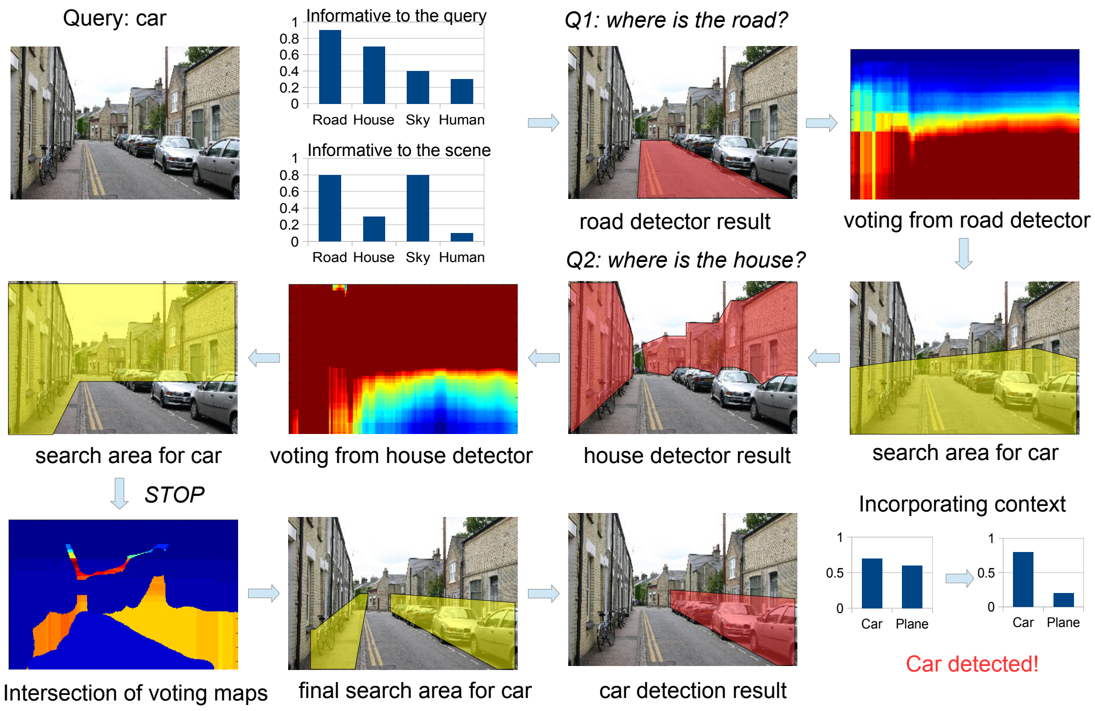

Figure 4.1: Illustration of our sequential search for objects in 20 context driven questions.

segmentation in complex scenes using an active divide-and-conquer approach fully driven by semantic context.

The contributions of this chapter are:

- a dynamic, closed-loop policy to decide the most informative action at each step considering both the query object and the scene
- a general and unified probabilistic framework incorporating responses from multi-class object detectors and contextual classifiers to update the model and conduct inference
- a data-driven context model that not only encodes co-occurrence but also spatial relations by efficient weighted vote maps from exemplars.

4.2 Related Work

Sequential Testing. The "20 question" approach to pattern recognition dates back to Blanchard and Geman [67], motivated by the large number of possible explanations in scene interpretation. They formally studied coarse-to-fine search in the theoretical framework of sequential hypothesis testing, and proposed optimal strategies considering both the cost and effectiveness of each test. Although they did not consider contextual information, their work provides a theoretical foundation for the design of sequential algorithms.

There are several works [68] on classifying objects by running classifiers sequentially in an active order. [69] proposed an information gain based approach to iteratively pose questions for users and incorporate human responses and computer vision detector results for fine-grained classification. [70] formulated object classification as a Markov decision process, where actions are the detector to deploy next. The model maintains a belief of object classes and keeps updating it based on new observations. They used reinforcement learning to train the detector selection policy, which becomes expensive when the number of classes and data size is large due to exploration. However, these approaches only focus on classifying objects. They have not addressed the challenging problem of simultaneous segmentation and localization of objects in a multi-class scene as we do in this chapter, and did not exploit inter-object context.

[71] applied a sequential decision making framework to window selection. The next window is selected based on votes of previously evaluated windows. However, the voting process needs to look up nearest neighbors in hundreds of thousands of exemplar window pairs in the training set because their context is at the exemplar/instance level, which is highly inefficient. In contrast, our context modeling is semantically aware so we do not compute nearest neighbors over hundreds of thousands of windows in a high dimensional descriptor space to retrieve the voters, we only need votes from a few regions within the search space of context class instead of sampling hundreds of windows in [71]. Our context model achieves good accuracy while greatly reducing computational complexity.

Object Detection. A common approach to object detection is based on applying gradient based features over densely sampled sliding windows [60].Such methods achieve good results on classes like human and vehicles, but they are very inefficient since they evaluate thousands of windows in an image, and false positive detections arise. To reduce the number of windows evaluated. [72] proposed a subwindow search based on a branch-and-bound scheme and only evaluates the high scoring windows. Recently, category independent object proposals [61–63] have been proposed to generate a small number of high quality regions or windows that are likely to be objects. These approaches dramatically reduce the number of candidates and reduce false positive detections. Using these object proposals [73, 74] train and apply deep neural network models on large datasets to learn the feature extractor and classifiers, and achieve state-of-the-art performance on the Pascal VOC detection challenge.

Object Recognition using Context. Context has been shown to improve object recognition and detection. Model-based approaches learn the appearance of semantic categories and relations among them using a parametric model. In [11, 64–66, 75], CRF models are used to combine unary potentials based on visual features extracted from superpixels with neighborhood constraints and low level context. Inter-object context in the scene has also been shown to improve recognition [38, 65]. Most of these context models are used as post-detection smoothing after all classifiers are run as unary potentials, and then they are jointly incorporated in inference regardless of their importance to different kinds of objects and scenes. Our framework, in contrast, evaluates the informativeness of context in an active loop before classifications of all objects are made, and goes beyond simple co-occurrence statistics.

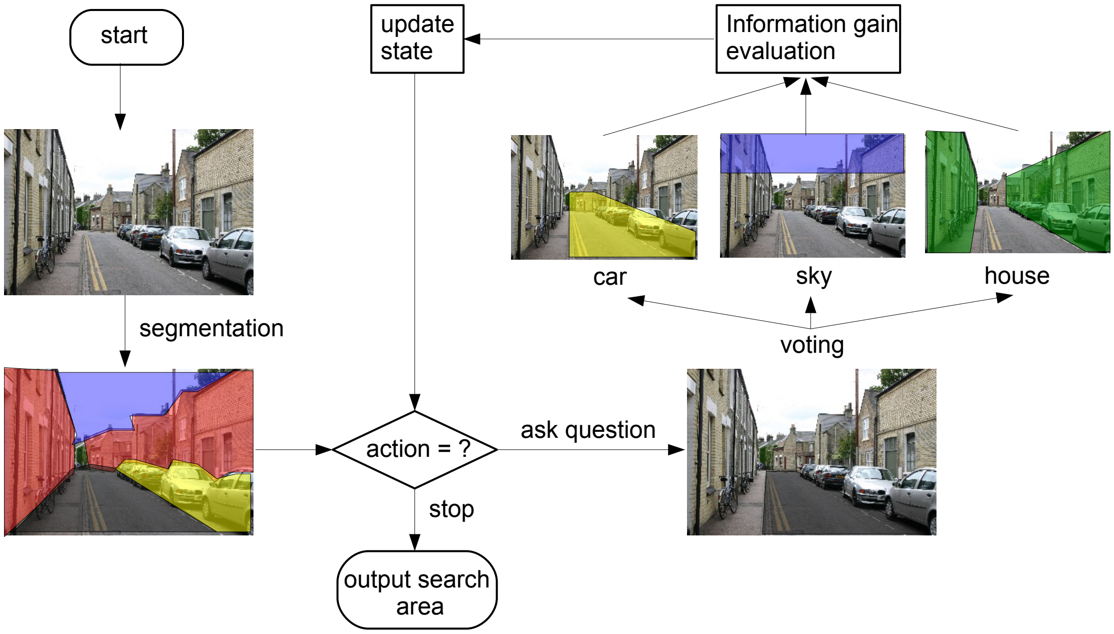

Figure 4.2: Flow chart of our context driven object searching. We first generate region hypotheses using object proposal algorithms, then the policy evaluates the current state and iteratively selects the most informative question considering both the scene and the query. Afterwards, the possible search locations are updated and the posterior probabilities of each category are updated for the next state.

4.3 Overview

Our framework is shown in Figure 5.2. Given an image X and a query object class c_q, our goal is to determine whether and where it occurs in the image by sequentially posing and answering questions about object and context. To reduce the searching complexity, our approach is based on object segment hypotheses generated by stable segmentation algorithms [27] or category independent object proposals [63]. Features for the scene and objects classification are precomputed. Initially classes with high cooccurrence with the query class will be assigned to the "20 questions set" $U = \{q_1, q_2, ..., q_M\}$ where M is the maximum number of questions to ask. Our goal is to learn a closed-loop policy that makes decisions dynamically given computed context information. A *closed-loop* policy is one that takes the previous feedbacks into consideration, while an *open-loop* policy is one that makes decisions independently from the previous actions [70]. We design a dynamic *closed-loop* policy to select the most informative next action by exploiting signal in the current scene and integrating observations over time. After the action is taken, the policy updates the posterior probability of the query class and the observed classes given the observations of responses so far, and evaluates the information gain. If the information gain is small, we use the output policy (See Section 4.4.4), to output the reduced possible search space for the query object, and then run the object detector for the query class together with the obtained context probabilities. We describe our search policy and context modeling in Section 4.4.1 and Section 5.4.2 respectively, then present the implementation details in Section 4.5.2. Finally we evaluate the performance of our approach in Section 4.5.

4.4 Search for Objects in 20 Questions

Given an image $X = \{x(i, j)\}$ and a query object class c_q, our goal is to output the detection and localization results of the query class in the scene. We model it in a reinforcement learning framework, where we design two policies: one is for sequential searching driven by context, and the other is the output policy. In this framework, we have a *state* $s(t) = (X, R^t) \in \mathcal{S}$ that includes the image X and the observation R^t at time t. Our search policy $\pi(s(t), a^t) : \mathcal{S} \to \mathcal{A}$ maps the current state to the next *action* a^t in the action space $\mathcal{A}$.

4.4.1 Context Driven Search Policy

Our search policy is shown in Algorithm 1. During test time, our policy repeatedly selects an action $a_t \in \mathcal{A}$, executes it, and obtains response r_t at time step t, and then selects the next action. The set of actions $\mathcal{A} = \{a_1, ..., a_C, STOP\}$, where a_i :=“*ask questions* q_i”, and q_i :=“*where is class* c_i”, or stop and apply the output policy to evaluate the final results once the information gain is too small.

Formally, at each time step t, we select a question q_t and take action a_t to evaluate it. Let $R^t = \{r_1, r_2, ..., r_t\}$ be the observations of responses to the actions taken at time $1...t$, where the response $r_t = p(c_t|X)$ is the detection or classification probability of class c_t corresponding to question q_t. We propose a policy to use maximum information gain of both the scene and the query object as the criterion to select q_t. Such criteria are commonly used in decision theory, require no training and few parameters, and can easily adapt to different query classes and scenes.

Algorithm 1 Context Driven Object Search Policy

```
Input: $X, c_q$
Initialization: $R^0 = \emptyset$,$\mathcal{A} = \{a_1, ..., a_C, STOP\}$, $H(c|X, R^t) = 0$, $X_c = X, c = \{1, .., C\}$
while $|\mathcal{A}| > 0$ and  do
    $c_t, \mathcal{I}^* = \arg\max_{c_i,\mathcal{I}} \mathcal{I}(c; c_i|X, R^t)$
    if $\mathcal{I}^* < \delta$ then
        $a_t :=$ STOP
        break
    end if
    $a_t :=$ run classifier/detector of $c_t$
    $\mathcal{A} \leftarrow \mathcal{A} \backslash a_t$
    for $X_s \in X_{c_t}$ do
        $p(c_t|X_s) \leftarrow a_t$
    end for
    for $c \in C$ do
        for $s \in X_{c_t}$ do
            for $i$-th region pair $(s^i_{ct}, s^i_c) \in$ training set do
                $p(c|c_t, X_s) \leftarrow$ vote$(s^i_{ct}, s^i_c)$
            end for
        end for
        $p(c_t|c, X) \leftarrow p(c|c_t, X)p(c_t|X)$
        $p(R^{t+1}|c, X) \leftarrow \prod_{i=1}^{t} p(c_i|c, X)$
        $p(c|X, R^{t+1}) \leftarrow p(R^{t+1}|c, X)p(c|X)$
        $H(c|X, R^{t+1}) \leftarrow Entropy(p(c|X, R^{t+1}))$
    end for
    $r_t :=$ observation after $a_t$
    $R^{t+1} = R^t \cup r_t$
    $\mathcal{I} \leftarrow p(r_t|X, R^t)(H(c|X, R^t \cup r_t) - H(c|X, R^t))$
end while
Output $p(c_q|X, R^t)$
```

We define $\mathcal{I}(c; r_i|X, R^t)$, the expected information gain of posing the additional question q_i as follows:

$$\mathcal{I}(c; r_i|X, R^t) \quad = \quad \lambda \mathcal{I}_{c_q}(c; r_i|X, R^t) + (1-\lambda)\mathcal{I}_{scene}(c; r_i|X, R^t) \tag{4.1}$$

where $\mathcal{I}_{c_q}(c; r_i|X, R^t)$ and $\mathcal{I}_{scene}(c; r_i|X, R^t)$ are expected information gain for the query class and scene respectively. λ is the parameter to balance query and the scene information, which can be learned or determined via cross validation.

4.4.1.1 Information about the Query

To select the most informative question that leads to an answer regarding the query object, we define the information gain for the scene based on posing question q_i as:

$$\begin{aligned} &\mathcal{I}_{c_q}(c_q; r_i|X, R^t) \\ &= \mathbb{E}_r[KL(p(c_q|X, r_i \cup R^t)||p(c_q|X, R^t))] \\ &= p_{c_q}(r_i|X, R^t)(H(c|X, r_i \cup R^t) - H(c|X, R^t)) \end{aligned} \tag{4.2}$$

where $H(c_q|X, R^t)$ is the entropy of $p(c_q|X, R^t)$

$$H(c_q|X, R^t) = -\sum_{c_q=0}^{1} p(c_q|X, R^t) \log p(c_q|X, R^t) \tag{4.3}$$

4.4.1.2 Information about the Scene

Besides the query class, our information gain based policy also considers context consistency in the scene and selects the context class that is both relevant and easy to find in the scene for making a decision about the query. The information gain for the scene by posing question q_i is defined as:

$$\begin{aligned} &\mathcal{I}_{scene}(c; r_i|X, R^t) \\ &= \mathbb{E}_r[KL(p(c|X, r_i \cup R^t)||p(c|X, R^t))] \\ &= p_{scene}(r_i|X, R^t)(H(c|X, r_i \cup R^t) - H(c|X, R^t)) \end{aligned} \tag{4.4}$$

where $H(c|X, R^t)$ is the entropy of $p(c|X, R^t)$

$$H(c|X, R^t) = -\sum_{c=1}^{C} p(c|X, R^t) \log p(c|X, R^t) \tag{4.5}$$

In the following sections we will discuss each term in the information gain and how they integrate observation of detector responses and context over time and space to finalize the decision.

4.4.2 Updating Responses and Context

To make use of the observed responses from taken actions, we propose a simple framework that alternates between exploration in action space and updating the current state. The key is to compute $p(c|X, R)$, where R is a sequence of responses from the actions taken:

$$p(c|X, R^t) = \frac{p(R^t|c, X)p(c|X)}{Z} \tag{4.6}$$

where $Z = \sum_c^C p(R^t|c, X)p(c|X)$ is the partition function. This term evaluates the probability of the true class c given observed responses and the current image. The responses of detectors depend on different query classes c and the specific image X. We assume the detectors are trained independently per category, thus the aggregated responses can be modeled as:

$$p(R^t|c, X) = \prod_t p(r_t|c, X) \tag{4.7}$$

where $p(r_t|c, X) = p(c_t|c, X)$ is context and scene dependent model for having detection response r_t at time step t. We present the details about how we model context in Section 5.4.2.

Similarly, we define $p(r_i|X, R^t)$ in Equation 4.2 and 4.4 as

$$p_{c_q}(r_i|X, R^t) = \sum_{c_q=\{0,1\}} p(r_t|c_q, X)p(c_q|X, R^t) \tag{4.8}$$

and

$$p_{scene}(r_i|X, R^t) = \sum_{c_q=1}^{C} p(r_t|c, X)p(c|X, R^t) \tag{4.9}$$

4.4.3 Context Modeling

Since our task is not only to detect the object but also refine the search space of the query in the image as accurately as possible, conventional modeling of context as simple co-occurrence statistics is inadequate. Instead we present a data-driven location aware approach to represent the spatial correlation between the objects and the scene.

Here we formulate the context $p(c_t|c, X)$ as a posterior of the probabilistic vote map $p(c|c_t, X_s)$ defined on each pixel $(x_i, x_j) \in X$ over the image, and the responses of class c_t after action a_t:

$$p(c_t|c, X) = \sum_{s \in X_{c_t}} p(c|c_t, X_s)p(c_t|X_s) \tag{4.10}$$

Given a refined search space $X_{c_t} \in X$ of a context class c_t at time t, we formalize $p(c|c_t, X)$ as a weighted vote from the cooccurring region pairs of class c_t and c in training scenes. Let $(s_{c_t}^i, s_c^i)$ be the i-th pair of co-occurring regions of class c_t and c, and $b_{c_t}^i$ and b_c^i be the corresponding bounding boxes. We can now define the probabilistic vote map $p(c|c_t, X)$ as:

$$p(c|c_t, X_s)_{s \in X_{c_t}} = \frac{1}{Z_c} \sum_i W(s_{c_t}^i, s; \theta^W).T(b_{c_t}^i, b_c^i) \tag{4.11}$$

where $s \in X_{c_t}$ is a region within the search space of the context class c_t. Z_c is the normalization function. $W(.)$ is a kernel measuring similarity of region s with a training region s_i. $T(b_{c_t}^i, b_c^i)$ models the transformation from $b_{c_t}^i$ to b_c^i, including translation and scaling. Figure 4.3 shows a few examples of the vote maps. We can see that with the exemplar based and semantically aware voting, the resulted vote maps give more accurate search area of the query objects.

The final context probabilistic vote map is given by

$$p(c_t|c, X) = \sum_{s \in X_{c_t}} p(c_t|X_s) \sum_i W(s_{c_t}^i, s; \theta^W).T(b_{c_t}^i, b_c^i) \tag{4.12}$$

where $p(c_t|X_s)$ is the probabilities of s as class c_t after taking the action a_t to run classification at time t.

4.4.4 Output Policy

Our search policy stops when the information gain is smaller than a threshold δ (set at 0.1 per pixel in our implementation). If the classifier of the query is already run in oberservation R^t, then it will directly output the probability $p(c_q|X, R^t)$ as the detection result. Otherwise the policy will output the $p(c_q|X, R^t)$ as the reduced search area for the query class detector, and run the detector only over the area for the detection results.

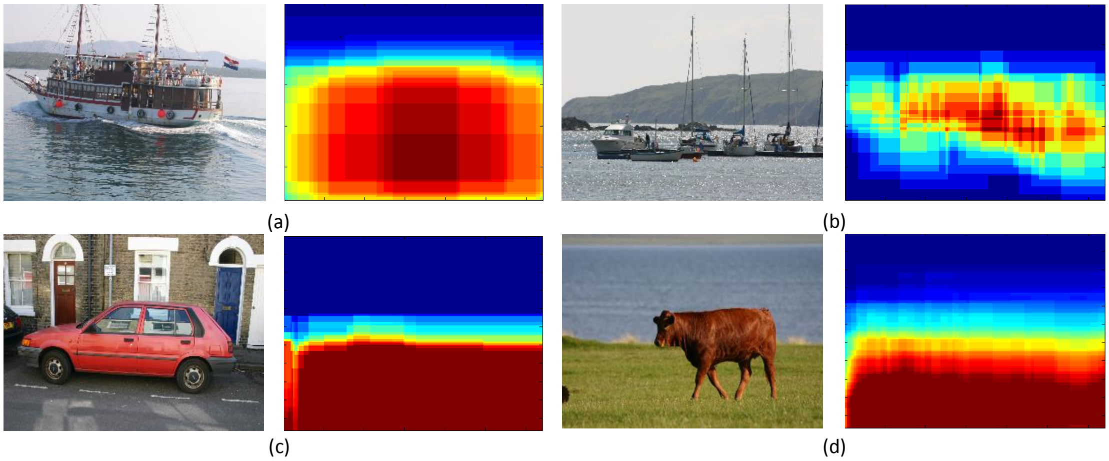

Figure 4.3: Examples of our context vote maps. Each pair of images corresponds to the original image and the vote-based probability map of object location from observed context. From (a) - (d) are the vote maps from water to boat, sky to boat, road to car and grass to cow, respectively. Best viewed in color.

4.5 Experiment

4.5.1 Dataset and Evaluation Metrics

We evaluate the efficacy of our sequential object detection approach on the MSRC dataset [11], which is a multi-class dataset with full annotation, containing multiple objects and context classes in the images. We select the object classes as the queries and encode the context in a lookup table of their features and pairwise transformation between each cooccurring region pairs.

The results are evaluated using the AP^r and AP$^r_{vol}$ measures, similar to [74], where the AP^r score is the average precision of whether a hypothesis overlaps with the groundtruth instance by over 50%, and the AP$^r_{vol}$ is the volume under the precision recall (PR) curve, which is more suitable for the simultaneous segmentation and detection task.

4.5.2 Implementation Details

4.5.2.1 Object Hypotheses

We use the category independent object proposals generated using the algorithm in [63]. Since our context driven search policy reduces the search space, we only need 30-50 object hypotheses in one image. Because these object hypotheses mainly cover the objects in the image, we also generate other regions for context classes using the finest level of the multiple segmentations similar to that in [38]. The total number of superpixels is around 50 in each images.

4.5.2.2 Feature Representation and Classification

We extract region features and classify them for object classes using the deep neural network model in [74] fine-tuned on Pascal VOC 2012. For context classifiers we use a subset of the appearance features for superpixels from [45] and learn one-vs-all SVM models for classification. The features we use are shown in Table 4.1.

4.5.3 Baselines

We compare our method to three baselines: exhaustively running classifiers from [74] on all the regions and output classification scores; searching for objects in randomly sampled location; and searching for the object by asking questions in random order, regardless of the query and the scene.

Table 4.1: Our superpixel features for context classification

Type	Name	Dimension
Color	RGB	3
	HSV values	3
	Hue	6
	Saturation	4
Texture	DOOG filters and stats	15
	Texture Histogram	100 × 2
	Normalized x and y	8
Shape and Location	Bounding box size relative to image size	2
SIFT	SIFT Histogram	100 × 2

	[74]	Ours	[74]	Ours	[74]	Ours
	GT	GT	MCG N=20	MCG N=20	MCG N=40	MCG N=40
cow	60.7	**71.3**	72.8	**75.2**	77.6	**79.6**
sheep	92.9	92.9	80.2	**83.0**	80.8	80.8
bird	53.2	**66.7**	40.7	**43.8**	51.2	**59.2**
chair	92.9	92.9	**29.3**	26.6	39.3	**39.6**
cat	100.0	100.0	60.0	**78.9**	60.0	**78.9**
dog	100.0	100.0	68.8	**83.4**	70.9	**84.0**
boat	**63.6**	60.7	21.4	**22.9**	22.3	**27.1**
body	41.0	**45.0**	21.2	**26.0**	**30.1**	24.2
car	76.3	**77.4**	44.5	**51.4**	52.8	**59.3**
bike	100.0	100.0	11.0	**16.7**	18.1	**25.0**
plane	92.9	92.9	36.4	**37.5**	45.7	**54.8**
mean	79.4	**81.8**	44.2	**49.6**	48.9	**55.7**

Table 4.2: Comparison with exhaustive detection on MSRC object classes in AP^r. All numbers are %.

	[74]	Ours	[74]	Ours	[74]	Ours
	GT	GT	MCG	MCG	MCG	MCG
			N=20	N=20	N=40	N=40
cow	48.2	**57.0**	55.0	**59.1**	57.0	**61.1**
sheep	85.8	**89.7**	65.2	**70.3**	64.0	**65.9**
bird	55.2	**65.4**	**38.9**	36.6	41.8	**48.3**
chair	85.0	**90.3**	**39.0**	38.6	42.9	**47.8**
cat	97.8	**100.0**	52.5	**60.5**	51.7	**62.1**
dog	99.1	**100.0**	59.0	**60.0**	59.8	**61.3**
boat	62.9	**62.9**	28.6	**30.1**	28.1	**31.4**
body	34.7	**40.1**	22.4	**24.4**	**24.3**	23.3
car	71.4	**79.8**	41.6	**43.1**	45.0	**46.1**
bike	90.6	**95.4**	33.4	**33.8**	33.4	**37.4**
plane	83.1	**89.2**	40.0	**41.9**	40.9	**45.4**
mean	74.0	**79.1**	43.2	**45.3**	44.5	**48.2**

Table 4.3: Comparison with exhaustive detection on MSRC object classes in AP$^r_{vol}$. All numbers are %.

4.5.4 Results

4.5.4.1 Comparison with Exhaustive Search

Table 4.2 and Table 4.3 shows the quantitative results on the MSRC object classes in AP^r and AP^r_{vol} metrics respectively, comparing our sequential object search approach with an exhaustive search using the same classifiers trained in [74], where “GT” denotes using ground truth segmentation as input object proposals and “MCG” are the one generated from [63]. We can see that given the correct localization provided by groundtruth regions, the baseline classifier achieved high scores similar to our context driven search method. But when performing detection using object proposals in multiple locations, our context driven approach outperforms the baseline exhaustive approach. We also notice that our performance using 20 MCG proposals even outperforms that of the baseline using 40 regions, in both AP^r and AP^r_{vol}. This shows that our method is more efficient in searching by greatly reducing the number of object proposals needed while achieving higher accuracy.

Figure 4.4 shows some qualitative results for detection and segmentation of the MSRC object classes. We can see that using our sequential search approach, the localization of objects is more accurate because of a refined search area, while the ambiguities have been reduced given observed context.

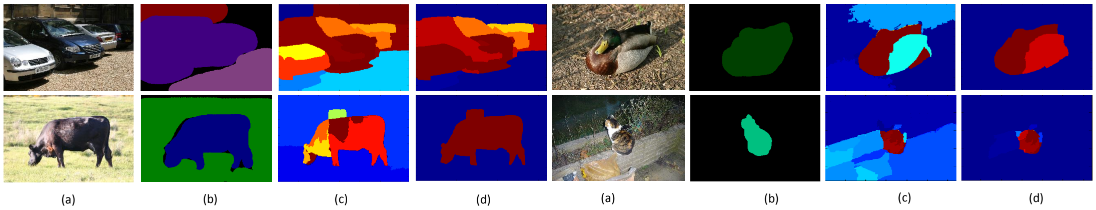

Figure 4.4: Qualitative results for detection and segmentation of the MSRC object classes. Columns (a) to (d) correspond to the original image, groundtruth label, probability map of the query object given by exhaustive search and by our sequential search respectively. The probability map from red to blue corresponds to the probability from high to low. Best viewed in color.

4.5.4.2 Comparison with Random Search

Table 4.4 shows the quantitative results on the MSRC object classes comparing our approach with random location searching and random order searching, in AP^r and AP$^r_{vol}$ metrics respectively. The random location search is performed by running detectors on randomly sampled locations, while the random sequence search is searching for objects by asking a fixed number of randomly selected questions and updating the search location sequentially. Given the same object proposals, we can see that random search approaches lead to more false object locations and gives poor detection and segmentation results. We can also see that without our context driven search model, the random sequential search policy gives the poorest results, showing the importance of having context in the searching process.

	Mean AP^r	Mean AP$^r_{vol}$
Random Sequence	5.9	8.2
Random Location	7.7	8.9
Ours	**55.7**	**48.2**

Table 4.4: Comparison with random search in AP^r and AP$^r_{vol}$ on MSRC. All numbers are %

4.6 Conclusion

We presented an efficient object search policy that determines the most informative questions for both the query and the scene. This policy is driven by a semantic context model using location voting maps. We formulate the policy in one probabilistic framework that integrates current information and the history of observations to update the model and determine the next most informative action to take. Experiments show the efficacy of our algorithm compared with baselines of exhaustive search and searching for objects in random locations or in a random sequence.

Chapter 5: Learning to Detect Objects in Twenty Questions

5.1 Introduction

Object detection and segmentation in complex scenes is a central and challenging problem in computer vision and robotics. This problem is usually tackled by running multiple object detectors exhaustively on densely sampled sliding windows [60] or category-independent object proposals [61–63]. Such methods need to evaluate a large number of object hypotheses indiscriminately, and can easily introduce false positives if exclusively considering local appearance.

Instead of checking all hypotheses exhaustively, humans only look for a set of related objects in a given context [1]. Context information is an effective cue for humans to detect low-resolution or small objects in cluttered scenes [76]. Many contextual models have been proposed to capture relationships between objects at the semantic level to reduce ambiguities from unreliable independent detection results. However, such methods still need to evaluate the high order co-occurrence statistics and spatial relations of the query object with *all* other object classes in the scene, some of which may not be informative and even introduce unwanted confusion.

By contrast, humans do not process the whole scene at once: human visual perception is an active process that sequentially samples the optic array in an intelligent, task-specific way [77]. Research in neuroscience has revealed that when humans search for a target, those objects that are associated to the query will reinforce attention with the query and weaken recognition of unrelated distractions [78]. For instance, in Figure 5.1, when we search for cars, knowing the top of the scene is sky does not help distinguish whether the image contains a car or a boat since both are equally likely to be under the sky; on the other hand, observing a road instead of water in the lower part gives a strong indication of the existence of cars. Therefore, in order to find cars, humans tend to first look for roads instead of sky; additionally, if we cannot find cars on the road, we may want to look beside the buildings because cars are likely to park next to them. This motivates us to raise the question: *can object detection algorithms decide where to look for objects of a query class more efficiently and accurately by exploring a few related context cues dynamically, similar to humans?*

To this end, we propose a generic strategy for object proposals-based object detection to explore the search space dynamically based on learned contextual relation, which achieves better speed-accuracy tradeoff. We formulate the object detection problem as a Markov Decision Process (MDP), and use imitation learning to learn a context-driven policy that sequentially and dynamically selects the most informative context class to explore based on past observations, and gradually refine the search area for the query class.

We show our framework in Figure 5.2. Specifically, like playing a 20 Questions game, at each step the policy asks for information about a context class such as road or building based on the query (e.g. car) and responses from previous contextual classifiers. We then run the detector/classifier of the selected context class. Based on the responses, we further refine the search area for the query class using spatially-aware contextual models. This process of contextual querying and search area refinement is repeated until the policy determines that sufficient contextual information has been gathered and decides to stop. Finally, we run the query object detector in the refined search area and output the result. Besides asking for contextual information, our policy can reject a query early to avoid unnecessary computation if it determines that there is little chance of the query object being in the scene. The early rejection decision can be taken even before running any object detector; therefore we can eliminate a large amount of unnecessary computation.

To demonstrate the efficacy of our idea, we implement our algorithm based on the Simultaneous Detection and Segmentation (SDS) [74] framework, but our algorithm is generic and can be extended to different object proposal-based methods such as [79] and [80]. Object detection experiments on the PASCAL VOC dataset show that our algorithm produces a search area that has better overlap with the target object by leveraging its context, thus significantly eliminating 45% of object proposals and 36% of total evaluation time compared to an exhaustive detection approach. Even with less computation, our method achieves mean average precision (mAP) higher than the exhaustive search method. To the best of our knowledge, this is one of the first few approaches that solve the challenging task of simultaneous object detection and segmentation in complex scenes in an MDP framework by actively acquiring and leveraging task-specific contextual information.

5.2 Related Work

Sequential Testing. The “20 question” approach to pattern recognition dates back

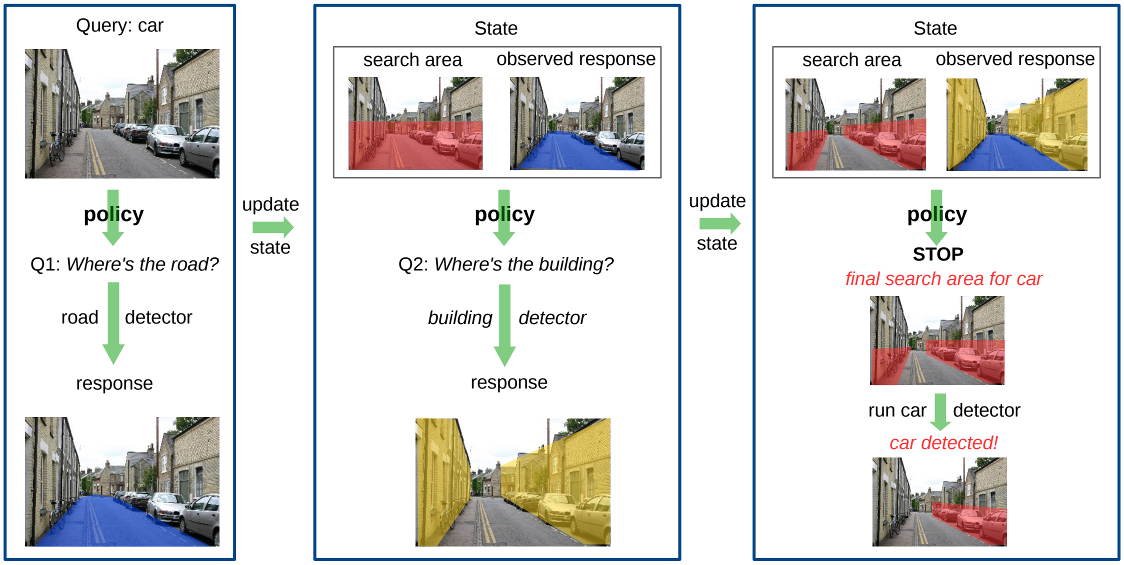

Figure 5.1: Illustration of our sequential search for query objects in 20 context-driven questions.

to [67], motivated by the scene interpretation problem with a large number of possible explanations. Their work provides a theoretical foundation for the design of sequential algorithms. "20 questions" approaches recently have been used to generate questions for users in applications such as image binary segmentation [81] and "visual Turing test" [82]. But such methods involve humans in the loop during test time, which is expensive and hard to scale up. There have been recent attempts to model the computational processes of visual attention [83] for object recognition. Such methods focus on low level salience and are tested in simple scenarios such as MNIST dataset.

There are several models [68] of objects classification that operate by running classifiers sequentially in an active order. [69] proposed an information gain based approach to iteratively pose questions for users and incorporates human responses and computer vision detector results for fine-grained classification. [70] formulated object classification as a Markov decision process to select classifiers under given time constraints. However, these approaches only focus on classifying objects. They have not addressed the challenging problem of simultaneous segmentation and localization of objects in a multi-class scene as we do in this work, and did not exploit inter-object spatial context.

Object Detection. Some common approaches to object detection are based on applying gradient based features over densely sampled sliding windows [60], which are very inefficient since they evaluate up to hundreds of thousands of windows in an image, and false positive detections arise. To reduce the number of windows evaluated, category independent object proposals [61–63] have been proposed which generate a small number of high quality regions or windows that are likely to be objects. These approaches dramatically reduce the number of candidates and reduce false positive detections. Using these object proposals [73, 74, 79] train and apply deep neural network models to learn the feature extractor and classifiers, and achieve state-of-the-art performance on the Pascal VOC detection challenge. However, such category independent proposals do not adapt to different query classes and still lead to a significant amount of unnecessary detector computation.

Object Recognition using Context. Context has been shown to improve object recognition and detection. In [11, 66], CRF models are used to combine unary potentials based on visual features extracted from superpixels with neighborhood constraints and low level context. [84] shows that using contextual information can improve object detection using CRF models. However these approaches evaluate the high order co-occurrence statistics with *all* other object classes appearing in the scene altogether, some of which may not be informative. Our framework, in contrast, only evaluates the most related context in an active sequence before classifications of all objects are made, and goes beyond simple co-occurence statistics. [71] applied a sequential decision making framework to window selection by voting for the next window. However, the voting process needs to look up nearest neighbors in hundreds of thousands of exemplar window pairs in the training set because their context is purely based on appearance similarity at the instance level, which is highly inefficient. By contrast, our model is based on context between semantic classes, which greatly reduces computational complexity.

5.3 Problem Formulation

Given an image X and a query class c_q ($q \in 1, .., C$, where C is the total number of object classes), we detect instances of the query class by sequentially choosing one context class to detect, and reduce the search area for the query class based on the responses of the context class detectors. The sequential decision-making problem can be formulated as a Markov Decision Process (MDP).

Definition 1. The **Object Detection MDP** is defined by the tuple $(\mathcal{S}, \mathcal{A}, T(.), R(.), \gamma)$:

- The **state** $s_t = (X^t, O^t)$, where X^t is the search area for the query at time t (initially X^0 is the entire image X), $O^t = \{o_1, o_2, \ldots, o_t\}$ is a sequence of observed *responses* from applied contextual classifiers;
- The **action** set $\mathcal{A} = \{a_1, \ldots, a_C, \mathit{Stop}, \mathit{Reject}\}$, where a_i corresponds to running the detector of class c_i, *Reject* corresponds to deciding that the query class does not occur in the image and terminate the process, and *Stop* terminates querying context classes and applies the detector of the query class on the current search area;
- The **state transition** function $T(s'|s, a)$ defines a next-state distribution after action a is taken in state s;
- The **reward** function $R(s, a) \rightarrow \mathbb{R}$ evaluates how good it is to take action a in state s;
- The **discount** factor γ is a constant controlling the tradeoff between greedily maximizing the immediate reward and the long term expected reward.

We define the *reward* R as the immediate gain in an intersection/union model of the search space:

$$R(s_t, a_t) = \frac{X^{t+1} \cap X_q}{X^{t+1} \cup X_q} - \frac{X^t \cap X_q}{X^t \cup X_q} \tag{5.1}$$

where X^{t+1} is the updated search area after executing action a_t in state s_t, determined by the context models described in the Approach section. X_q is the groundtruth mask of the query object instances in the image.

The query agent follows a *policy* $\pi : S \rightarrow A$ that determines which action to take in a given state. Given an optimal policy π^* which yields a state-action sequence that maximizes the discounted cumulative reward, the optimal Q-value is recursively defined as $Q^*(s_t, a_t) = R(s_t, a_t) + \gamma \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1})$, where a_t is chosen by π^* and γ is the discount factor. Our goal is to learn the optimal policy for the object detection MDP.

5.4 Approach

We show our framework in Figure 5.2. Given a query, we first generate object hypotheses as well as a small number of regions corresponding to contextual classes, then the policy sequentially either a) rejects the occurrence of the query, b) poses a question about a context class, or c) stops and runs the query detector. After an action is taken, the search locations are updated based on the responses and new posterior probabilities of each category are computed to update the state. In this section, we first present the imitation learning algorithm for learning a policy that plays the twenty questions game; we then describe how to refine the search area of the query given responses of contextual classifiers evaluated by previous questions.

5.4.1 Learning the Policy by Imitation

Typically, MDPs are solved by reinforcement learning (e.g., Q-learning [85], REINFORCE [86]). However, given our exponential search space, such trial-and-error approaches can take too long to converge. It is also difficult to specify a reward function for the policy because the underlying true reward is hard to model. We therefore take the imitation learning approach [87], where we assume direct supervisory signals from an oracle are available and learn to mimic the oracle's behavior.

Assuming we know the optimal Q-values, the optimal policy is straightforward

$$\pi^*(s) = \arg\max_{a \in A} Q^*(s, a). \tag{5.2}$$

To learn Q^*, we assume the optimal Q-values are given by an *oracle* at training time; thus we reduce to learning a linear approximation:

$$Q^*(s, a) = \theta_\pi^T \phi(s, a), \tag{5.3}$$

where $\phi(s, a) = \phi((X^t, O^t), a)$ is a feature representation of the state consisting of the search area X^t and observations O^t after executing actions $a_1, ..., a_t$. This can be solved by standard supervised learning algorithms.

We compute the oracle's action sequence by breadth-first search with pruning. The action sequence that maximizes the discounted cumulative reward in the terminal state is selected as the oracle's action sequence. We then collect examples $\{(s_t, a_t, Q_t)\}$ from the oracle's trajectory for policy training. However, collecting examples from the oracle's trajectory only may result in mismatch in distributions of training and test data, since the learned policy may go to states the oracle never visited. To solve the mismatch problem,

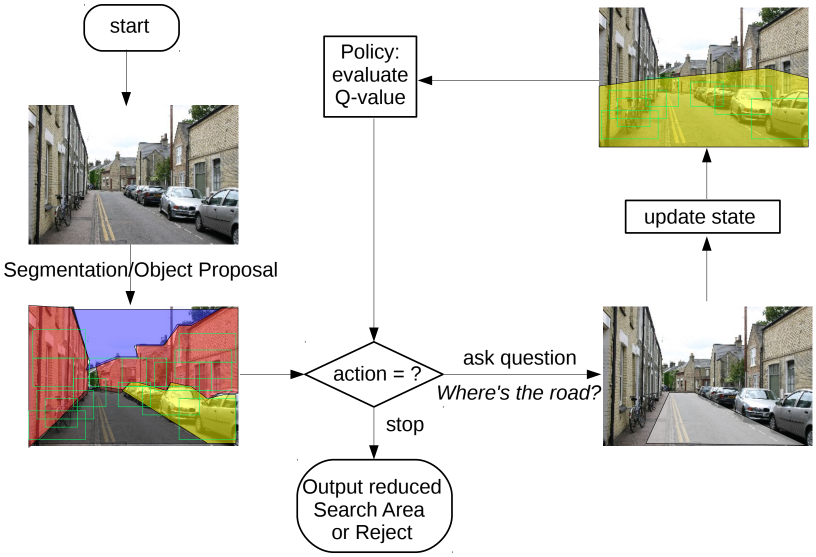

Figure 5.2: **Flowchart of our context driven object searching algorithm**.We first generate region hypotheses using object proposal algorithms, then the policy evaluates the current state and iteratively selects the action maximizing the Q-value function. Afterwards, the possible search locations are updated and the posterior probabilities of each category are evaluated for the next state.

we encourage exploration by searching multiple times with random starting states. Due to the large number of negative states for rejection, we sample the negative states by early pruning when the immediate reward is negative, which imitates the action of early rejection. After example collection, we train the policy (predict the optimal Q-values) by ridge regression.

5.4.2 Context Modeling

Since our task is not only to detect instances of the query object but also to refine the search space of the query in the image as accurately as possible, conventional modeling of context as simple co-occurrence statistics is inadequate. Instead we present a data-driven location-aware approach to represent the spatial correlation between the objects and the scene.

We capture the spatial relationships in a non-parametric manner. Figure 5.3 illustrates our model. During training, the bounding box of a region s^i indexed by i is represented by $b^i = (x^i, y^i, \sigma^i)$ with x, y as its center location and σ as the scale w.r.t. the image. For each pair of co-occurring regions belonging to class c_k and c_l respectively, we index this pair as j and store corresponding displacement vector $T_j = T(b^j_k, b^j_l)$ which includes translation $(\Delta x, \Delta y)$ and change of ratio in two directions between the two boxes.

During test time, we define $X_c \subset X$ as the *exploration area* for context which excludes the observed regions of other contextual classes in the image. Let $s^i \subset X_c$ be the context region i in a test image. Given an action a_k to detect context class c_k at time t, to model the context between class c_k and another class c_l, we model the context $p(c_k|c_l, X)$ as the posterior of the probabilistic vote map $p(c_l|c_k, X_c)$ defined for each pixel in the image, and the responses of class c_i after action a_i:

$$p(c_k|c_l, X) = p(c_k|c_l, X_c) = \sum_{s^i \subset X_c} \frac{p(c_l|c_k, s^i)p(c_k|s^i)}{p(c_l|s^i)} \tag{5.4}$$

where $p(c_k|s^i)$ is the probabilities of s^i as class c_k after taking the action a_k to run classification at time t.

We can now define the probabilistic vote map $p(c_l|c_k, s^i)$. Let (s_k^j, s_l^j) be the j-th training pair of co-occurring regions of class c_k and c_l, and b_k^j and b_l^j be their corresponding bounding boxes. Let $s_k^i \subset X_c$ be the context region i detected as class c_k in the test image. We retrieve those training pairs (s_k^j, s_l^j) between class c_k and c_l and compute the RBF kernel $W(.)$ measuring the similarity of the features of train/test segments of class c_k as $W(s_k^i, s_k^j; \theta^W)$, where θ^W is the kernel parameter. We then formalize $p(c_l|c_k, X_c)$ as a weighted vote from the co-occurring region pairs of classes c_k and c_l in training scenes.

$$p(c_l|c_k, s^i) = \frac{1}{Z_c} \sum_i \sum_j W(s_k^i, s_k^j; \theta^W).T(b_k^j, b_l^j) \tag{5.5}$$

where Z_c is the normalization function.

5.4.3 Update Responses and Search Area

After taking action a_t and receiving response $o_t = p(c_t|c, X)$ from context class c_t, we integrate the response into observations from the previous sequence of actions. Assuming the detectors and context classifiers are trained independently per category, the

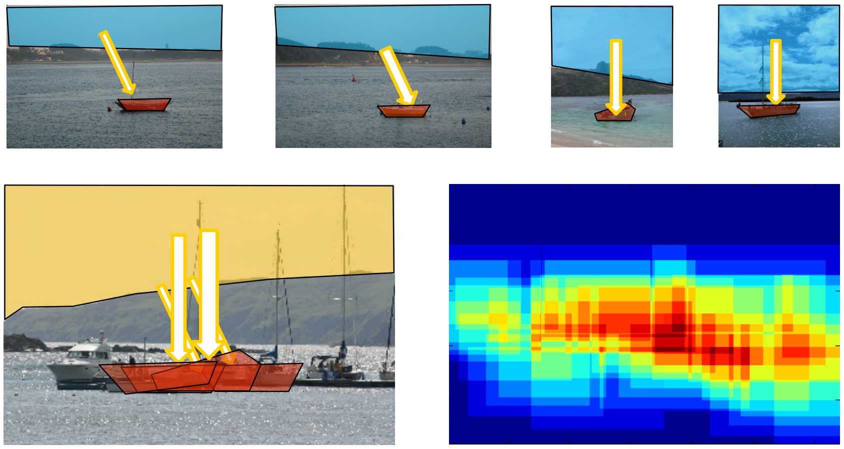

Figure 5.3: **Our context voting model**. The first row shows example training pairs of the sky and the boat. The second row shows the test image and the weighted voting map. The arrows denote applying the weighted displacement vectors $T(b_k^j, b_l^j)$ from the training pairs to the test pairs of sky and boat (highlighted in yellow and blue respectively).

aggregated responses can be modeled as:

$$p(O^t|c, X) = \prod_t p(c_t|c, X) \tag{5.6}$$

We then update the search area for the query class c_q in a probabilistic framework:

$$p(c_q|X, O^t) = \frac{p(O^t|c_q, X)p(c_q|X)}{Z} \tag{5.7}$$

where $Z = \sum_c^{Ct} p(O^t|c, X)p(c|X)$ is the partition function, $p(c|X)$ is obtained by taking actions and running context classifiers over the context segments, and $C^t = \{c_1, c_2, ..., c_t, c_q\}$ are the set of observed contextual classes.

5.5 Implementation Details

5.5.1 Object Proposals

We use MCG object proposals from [63] as object candidates. Since the object proposals mainly cover the objects, we also generate a small number (20∼30 per image) of segments using the stable segmentation algorithm from [28] to cover regions corresponding to contextual classes. To reduce computational overhead, our context voting step uses only the stable segments. The stable segmentation gives a coarse level of object/context division and reduces the computational complexity of context voting compared to the large number of finer object proposals, while still maintaining semantic spatial information.

5.5.2 Datasets

We conduct our experiments on the Pascal VOC dataset [88], a standard benchmark for object detection. Since the original dataset does not provide annotation for segmentation of contextual classes, we train our policy using the Pascal Context dataset [84] which fully annotates every pixel of the Pascal VOC 2010 train and validation sets, with additional contextual classes such as sky, grass, ground, building, etc. We use the 33 context classes from [84] and train our policy on the Pascal Context training set, and test our algorithm and baselines on the validation set. We also test our policy on the MSRC dataset [11] to show our algorithm can generalize to different data.

5.5.3 Feature Representation

To classify object proposals, we extract region features and classify them using the deep neural network model in [74] fine-tuned on Pascal VOC 2012. For the policy action classifiers, we use the same model to extract features for states represented by the masks of search area X^t and observed area O^t in state s_t, then concatenate the features as inputs to the policy. For context classifiers we use a subset of the appearance features for superpixels from [45] and learn one-vs-all SVM models for classification.

5.6 Experiments

5.6.1 Baselines

We compare with two recent popular exhaustive detection baselines, RCNN [73] and SDS [74]. RCNN adapts the CNN pertained for image classification [89] to the task of object detection by fine-tuning the network on warped object bounding boxes, then applies the network to extract CNN features on each object proposal for detection. SDS further extends RCNN to the task of segmentation by training and testing on region-based proposals. Both approaches need to extract features and run class-specific detectors exhaustively on all object proposals. We implement our algorithm based on the SDS framework. We also compare with random search which randomly samples the same number of object proposals for detection, window selection driven by context in [71], detection using object proposals in selective search [62] and objectness [90]. For average precision we also compare with a recently proposed contextual model in [84] which considers global and local context in a Markov Random Field framework based on a deformable part-based (DPM) model. This model has high computational cost since it needs to evaluate hundreds of thousands of windows as well as the context deformation term between all context boxes in the graph.

5.6.2 Speed-accuracy tradeoff

Figure 5.4 shows on the Pascal VOC 2010 dataset the average precision (mAP) performance VS the (amortized) number of detectors/classifiers evaluated on the object proposals. The amortized number of proposals consists of not only the resulting proposals for the query, but also the average overhead evaluation including context classifiers and the Q-value evaluations on the state masks, so it reflects the total computational cost. Our algorithm has significantly reduced both the number of object proposals for the query and the total computation time. Compared to the SDS, the reduction of the proposals for the object is 45%, and the overall reduction of time is 36%. Empirically it takes SDS about 13.3s to evaluate features for 2000 proposals for a class. With our algorithm, the average number of object proposal drops 45% resulting in computation of around 7.1s, plus about 0.8s for evaluating Q-values and 0.6s for context detectors. This is 36% reduction in amortized run time. With increasing numbers of object proposals, our algorithm can achieve even better results than exhaustive methods due to the reduction of false positives. We also see the random search approach performs poorly, showing the effectiveness of our context driven search approach.

In comparison to [71] which closely relates to our approach, context class lookup in [71] between 2.55 and 5.7s+0.26s to update the vote map, while our method only takes 0.6s, achieving 7x∼10x speedup. Although we use MCG object proposals that are already highly precise in object location, we still achieve 45% reduction on average.

5.6.3 Detection precision

Table 5.1 shows the classwise mAP of our 20 questions approach with other context based methods and their corresponding baselines. We compare our model with SDS and RCNN as well as [84] denoted as "Pascal 20/30 Context" in the table, and deformable part-based model with context denoted as “DPM(+context)”. Both the SDS and the 20 question methods start with 2000 object proposals per image. Our 20 question detection approach outperforms exhaustive search baselines SDS and RCNN as well as DPM based context approaches while reducing 45% of proposals.We can see that classes that empirically appear with more context with other objects in the scenes have significant gain in precision over exhaustive search, such as boat, car, chair, cow, sofa etc..

5.6.4 Search space accuracy

To measure the quality of our predicted search areas, we evaluate the mean intersect vs. union (IU) of the search area produced by our 20 questions approach with the groundtruth objects. We also compare with the search area of the original detector, produced by the union of the object proposals with high scores. The mean IU of the original detectors, our 20 questions approach and the oracle are 64.12%, 73.9% and 78.2% respectively. We can see that our approach significantly improves the accuracy of overlap between the predicted search area and the target query object. We also find that the mean IU of the 20 questions search space is close to that predicted by the oracle trajectory, which shows that our imitation learning has learned a good policy that closely mimics the oracle’s behavior.

5.6.5 Simultaneous detection and segmentation

Given that we employ segment based object proposals generated by [63], our detection system can also perform segmentation. We compare our algorithm with [74] in

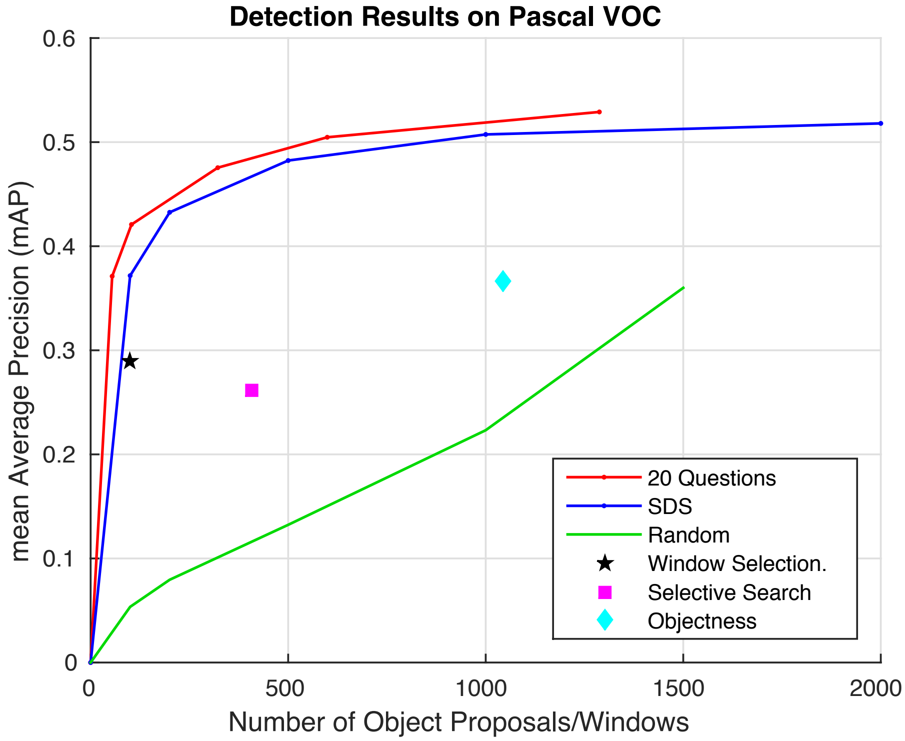

Figure 5.4: **Speed-accuracy tradeoff** mAP vs. number of amortized evaluated object proposals on Pascal VOC dataset. Best viewed in color.

Table 5.1: Avg. detection precision of ours and other algorithms on PASCAL VOC10 dataset.

	DPM	DPM+	Pascal	Pascal	RCNN	SDS	SDS+20Q
		33 Context	20 Context	33 Context			
Plane	44.3	46.4	46.9	49.8	69.9	67.3	66.8
Bike	51.3	50.8	50.1	48.8	64.2	63.6	64.3
Bird	7.1	7.5	9.2	12.0	48.0	47.1	48.9
Boat	8.0	8.2	9.5	10.8	30.2	33.1	36.1
Bottle	21.8	21.2	30.1	29.1	26.9	34.3	32.2
Bus	56.0	55.3	57.2	55.2	63.3	67.2	67.7
Car	41.2	41.6	44.1	45.6	56.0	55.8	56.5
Cat	18.4	20.0	30.7	32.0	67.6	74.6	70.4
Chair	13.8	14.7	12.7	14.2	26.8	24.9	28.1
Cow	11.7	11.8	15.1	12.6	44.7	44.8	58.3
Table	10.4	11.6	12.9	13.7	29.6	35.7	37.2
Dog	13.5	13.9	14.2	16.6	61.7	62.7	60.5
Horse	38.3	37.9	35.6	39.8	55.7	62.5	64.7
MBike	42.7	40.2	44.8	44.2	69.8	64.8	65.9
Person	44.6	45.1	44.0	45.1	56.4	59.1	52.1
Plant	3.7	4.2	4.9	8.2	26.6	26.9	26.7
Sheep	27.0	24.1	30.6	35.3	56.7	54.2	57.8
Sofa	24.3	27.6	20.1	26.0	35.6	40.7	46.6
Train	38.0	40.8	42.2	42.3	54.4	61.3	62.9
TV	32.2	33.9	34.8	34.3	57.7	55.7	53.3
Mean	27.4	27.8	29.5	30.8	50.1	51.8	52.9

Table 5.2: AP^r performance on PASCAL VOC10 dataset.

	SDS	**SDS+20Q**
Plane	68.2	66.7
Bike	52.1	55.0
Bird	51.6	52.2
Boat	30.7	33.3
Bottle	34.2	32.1
Bus	66.7	67.2
Car	52.4	53.7
Cat	70.9	66.9
Chair	21.0	24.2
Cow	39.6	53.8
Table	30.7	32.0
Dog	58.8	56.9
Horse	55.2	54.1
MBike	54.0	53.3
Person	55.5	49.6
Plant	25.0	25.1
Sheep	56.4	58.1
Sofa	33.3	35.1
Train	61.2	61.3
TV	58.4	55.5
Mean	48.8	49.3

Table 5.3: AP^r performance on MSRC dataset.

	cow	sheep	bird	chair	cat	dog	boat	body	car	bike	plane	mean
SDS	87.4	87.6	**49.6**	**52.2**	75.0	72.3	49.2	62.5	73.0	80.7	93.8	71.9
SDS+20Q	**88.4**	**93.8**	45.8	48.3	**82.6**	**76.7**	**51.7**	**65.5**	**79.0**	**85.2**	**95.7**	**73.2**

the simultaneous detection and segmentation task using the AP^r metric proposed in [74]. Table 5.2 and Table 5.3 show the performance on Pascal VOC10 and the MSRC datasets respectively. We outperforms the SDS approach on both datasets, showing our 20 questions algorithm can generalize well from the detection to the segmentation task, as well as generalize to other datasets such as MSRC.

Chapter 6: Conclusion

6.1 Summary

In this dissertation, we studied the problem of employing context for scene understanding. We addressed the core problems of scene understanding and computer vision including object recognition, object detection and semantic segmentation, and explored different approaches to leverage contextual information to enhance improve recognition and detection.

In Chapter 2, we presented an approach to jointly solve the segmentation and recognition problem using a multiple segmentation framework. We formulated the problem as segment selection from a pool of segments, assigning each selected segment a class label. Previous multiple segmentation approaches used local appearance matching to select segments in a greedy manner. In contrast, our approach formulates a cost function based on contextual information in conjunction with appearance matching. This relaxed cost function formulation is minimized using an efficient quadratic programming solver and an approximate solution is obtained by discretizing the relaxed solution. Our approach improved labeling performance compared to other segmentation based recognition approaches.

In Chapter 3, we introduced a new problem called object co-labeling where the goal is to jointly annotate multiple images of the same scene which do not have temporal consistency. We presented an adaptive framework for joint segmentation and recognition to solve this problem. We proposed an objective function that considers not only appearance but also appearance and context consistency across images of the scene. A relaxed form of the cost function is minimized using an efficient quadratic programming solver. Our approach improved labeling performance compared to labeling each image individually. We also showed the application of our co-labeling framework to other recognition problems such as label propagation in videos and object recognition in similar scenes. Experimental results demonstrated the efficacy of our approach.

In Chapter 4 and Chapter 5 of the dissertation, we proposed a novel general strategy for object proposal-based object detection. Instead of passively evaluating all object detectors at all possible locations in an image, we developed a divide-and-conquer approach by actively and sequentially evaluating contextual cues related to the query based on the scene and previous evaluations—like playing a "20 Questions" game—to decide where to search for the object. The policy driven by a semantic contextual model sequentially refines the search area for the query.

We formulated this strategy in two different ways. In Chapter 4, we presented an efficient object search policy that considers the most informative questions for both the query and the scene. The policy sequentially selects the question by maximizing the information gain based on the query, the scene and current observed responses given by object detectors and classifiers. This approach has few parameters and requires no learning. In Chapter 5, we further enhanced the framework by modeling object detection as a Markov Decision Process. The policy is learned by imitation learning fully driven by data. We apply the 20 questions approach in the recent framework of simultaneous detection and segmentation. Experimental results on the Pascal VOC dataset showed that our algorithm reduces about 45% of the object proposals and 36% of average evaluation time while improving average precision compared to exhaustive search. Our learned search policy also achieves better speed-accuracy tradeoff than random search.

6.2 Future Research Directions

In future, we would like to continue the research of scene understanding in these possible directions:

- General 20 Questions: we would like to introduce more general form of the questions to extract information and guide the scene understanding process. For example, instead of only asking binary questions about where is a context class, we can design an algorithm for attributes based questions such as "is it composed of metal material?"
- Visual Common Sense: we would like to incorporate high level knowledge mined from language and text databases to help computers reason about the visual world. For example, knowing a cat likes to lie on soft surface and bed has soft surface, we can infer the location of cat lying in a bedroom scene even if we do not have training samples showing a cat lying on the bed.
- Subgraph selection in 20 questions: currently in our 20 questions framework, at each step only one context question can be selected. This is not very stable because a wrong selection and response of context class may negatively impact decisions

afterwards. This approach will also be intractable when the number of possible classes are very large. So instead of querying one context class only, we would like to select multiple context classes as questions that are most informative to the query, and consider the interactions between the context classes themselves as well as between context and the object.

Bibliography

[1] Irving Biederman, Robert J Mezzanotte, and Jan C Rabinowitz. Scene perception: Detecting and judging objects undergoing relational violations. *Cognitive psychology*, 14(2):143–177, 1982.

[2] tephen E Palmer. The effects of contextual scenes on the identification of objects. *Memory & Cognition*, 3:519–526, 1975.

[3] Antonio Torralba. Modeling global scene factors in attention. *JOSA A*, 20(7):1407–1418, 2003.

[4] Antonio Torralba, Aude Oliva, Monica S Castelhano, and John M Henderson. Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. *Psychological review*, 113(4):766, 2006.

[5] Moshe Bar and Elissa Aminoff. Cortical analysis of visual context. *Neuron*, 38(2):347–358, 2003.

[6] Moshe Bar. Visual objects in context. *Nature Reviews Neuroscience*, 5(8):617–629, 2004.

[7] J. M. Tenenbaum and H. G Barrow. Experiments in interpretation guided segmentation. *Journal of Artificial Intelligence*, 8(3):241–274, 1977.

[8] R. Brooks, R. Greiner, and T. Binford. Model-based threedimensional interpretation of two-dimensional images. *In Proc. Int. Joint Conf. on Art. Intell.*, 1979.

[9] A. Hanson and E. Riseman. Visions: A computer system for interpreting scenes. *In Computer Vision Systems.*, 1978.

[10] Xuming He, Richard Zemel, and Deb Ray. Learning and incorporating top-down cues in image segmentation. In *ECCV*. 2006.

[11] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Computer Vision–ECCV 2006*, pages 1–15. Springer, 2006.

[12] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *Proc. IEEE CVPR*. 2008.

[13] Zhuowen Tu. Auto-context and its application to high-level vision tasks. In *CVPR*. 2008.

[14] Abhinav Gupta and Larry S. Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *Proc. ECCV*. 2008.

[15] C. Galleguillos, A. Rabinovich, and Serge Belongie. Object categorization using co-occurrence, location and appearance. In *Proc. CVPR*. 2008.

[16] Arpit Jain, Abhinav Gupta, and Larry S. Davis. Learning what and how of contextual models for scene labeling. In *ECCV*. 2010.

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

[18] Tomasz Malisiewicz and Alexei A. Efros. Recognition by association via learning per-exemplar distances. In *CVPR*. 2008.

[19] Stephen Gould, Tianshi Gao, and Daphne Koller. Region-based segmentation and object detection. In *NIPS*. 2009.

[20] M Pawan Kumar and Daphne Koller. Efficiently selecting regions for scene understanding. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3217–3224. IEEE, 2010.

[21] Li-Jia Li, Richard Socher, and Li Fei-Fei. Towards total scene understanding:classification, annotation and segmentation in an automatic framework. In *CVPR*. 2009.

[22] Lubor Ladicky, Chris Russell, Pushmeet Kohli, and Philip H. S. Torr. Associative hierarchical crfs for object class image segmentation. In *ICCV*. 2009.

[23] Lubor Ladicky, Chris Russell, Pushmeet Kohli, and Philip H.S. Torr. Graph cut based inference with co-occurrence. In *ECCV*. 2010.

[24] A. Torralba, K.P. Murphy, and W.T. Freeman. Contextual models for object detection using boosted random fields. In *NIPS*. 2005.

[25] Abhinav Gupta, Alexei Efros, and Martial Hebert. Block world revisited: Image understanding using qualitative geometry and mechanics. In *In ECCV*. 2010.

[26] Tomasz Malisiewicz and Alexei A. Efros. Improving spatial support for objects via multiple segmentations. In *BMVC*. 2007.

[27] Eitan Sharon, Meirav Galun, Dahlia Sharon, Ronen Basri, and Achi Brandt. Hierarchy and adaptivity in segmenting visual scenes. *Nature*, 442(7104):719–846, June 2006.

[28] Andrew Rabinovich, Andrea Vedaldi, Carolina Galleguillos, Eric Wiewiora, and Serge Belongie. Objects in context. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[29] D. Martin, Fowlkes C., and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Tran. on PAMI*, 26:530–549, 2004.

[30] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines. 2001.

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

[32] Prateek Jain and Ashish Kapoor. Active learning for large multi-class problems. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 762–769. IEEE, 2009.

[33] Marius Leordeanu, Martial Hebert, and Rahul Sukthankar. An integer projected fixed point method for graph matching and map inference. In *Advances in Neural Information Processing Systems*, pages 1114–1122, 2009.

[34] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A Zisserman. The pascal visual object classes challenge 2009 (voc2009) results.

[35] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 264–271, 2003.

[36] Abhinav Gupta and Larry S. Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *European Conference on Computer Vision (ECCV)*, pages 16–29, 2008.

[37] Santosh Kumar Divvala, Derek Hoiem, James H Hays, Alexei A Efros, and Martial Hebert. An empirical study of context in object detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1271–1278. IEEE, 2009.

[38] Xi Chen, Arpit Jain, Abhinav Gupta, and Larry S Davis. Piecing together the segmentation jigsaw using context. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2001–2008. IEEE, 2011.

[39] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):978–994, 2011.

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

[41] Carolina Galleguillos, Brian McFee, Serge Belongie, and Gert Lanckriet. Multiclass object localization by combining local contextual interactions. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 113–120. IEEE, 2010.

[42] Li-Jia Li, Richard Socher, and Li Fei-Fei. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2036–2043. IEEE, 2009.

[43] Evgeny Levinkov and Mario Fritz. Sequential bayesian model update under structured scene prior for semantic road scenes labeling. In *Computer Vision, 2013. ICCV 2013. IEEE International Conference on*. IEEE, 2013.

[44] Ce Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing: Label transfer via dense scene alignment. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1972–1979, 2009.

[45] Joseph Tighe and Svetlana Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *Computer Vision–ECCV 2010*, pages 352–365. Springer, 2010.

[46] Vijay Badrinarayanan, Fabio Galasso, and Roberto Cipolla. Label propagation in video sequences. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3265–3272. IEEE, 2010.

[47] A Y C Chen and J J Corso. Propagating multi-class pixel labels throughout video frames. In *Image Processing Workshop (WNYIPW)*, pages 14–17, 2010.

[48] Tomáš Kazmar, Evgeny Z Kvon, Alexander Stark, and Christoph H Lampert. Drosophila embryo stage annotation using label propagation. In *Computer Vision, 2013. ICCV 2013. IEEE International Conference on*. IEEE, 2013.

[49] Carsten Rother, Tom Minka, Andrew Blake, and Vladimir Kolmogorov. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 993–1000, 2006.

[50] Gunhee Kim, Eric P Xing, Li Fei-Fei, and Takeo Kanade. Distributed cosegmentation via submodular optimization on anisotropic diffusion. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 169–176. IEEE, 2011.

[51] Armand Joulin, Francis Bach, and Jean Ponce. Multi-class cosegmentation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 542–549. IEEE, 2012.

[52] Gunhee Kim and Eric P Xing. On multiple foreground cosegmentation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 837–844. IEEE, 2012.

[53] Michael Rubinstein, Armand Joulin, Johannes Kopf, and Ce Liu. Unsupervised joint object discovery and segmentation in internet images. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1939–1946. IEEE, 2013.

[54] Tianyang Ma and Longin Jan Latecki. Graph transduction learning with connectivity constraints with application to multiple foreground cosegmentation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013.

[55] Jifeng Dai, Ying Nian Wu, Jie Zhou, and Song-Chun Zhu. Cosegmentation and cosketch by unsupervised learning. In *14th International Conference on Computer Vision*, 2013.

[56] T. Malisiewicz and A. A. Efros. . improving spatial support for objects via multiple segmentations. In *British Machine Vision Conference(BMVC)*, 2007.

[57] Eitan Sharon, Meirav Galun, Dahlia Sharon, Ronen Basri, and Achi Brandt. Hierarchy and adaptivity in segmenting visual scenes. *In the journal of Nature*, 442(7104):719–864, 2006.

[58] Prateek Jain and Ashish Kapoor. Probabilistic nearest neighbor classifier with active learning. *Microsoft Research, Redmond*.

[59] Gabriel J. Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV (1)*, pages 44–57, 2008.

[60] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

[61] Joao Carreira and Cristian Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1312–1328, 2012.

[62] Koen EA Van de Sande, Jasper RR Uijlings, Theo Gevers, and Arnold WM Smeulders. Segmentation as selective search for object recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1879–1886. IEEE, 2011.

[63] Pablo Arbeláez, Jordi Pont-Tuset, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping. CVPR, 2014.

[64] Stephen Gould, Richard Fulton, and Daphne Koller. Decomposing a scene into geometric and semantically consistent regions. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1–8. IEEE, 2009.

[65] Carolina Galleguillos and Serge Belongie. Context based object categorization: A critical survey. *Computer Vision and Image Understanding*, 114(6):712–722, 2010.

[66] Lubor Ladicky, Chris Russell, Pushmeet Kohli, and Philip HS Torr. Graph cut based inference with co-occurrence statistics. In *Computer Vision–ECCV 2010*, pages 239–253. Springer, 2010.

[67] Gilles Blanchard and Donald Geman. Hierarchical testing designs for pattern recognition. *Annals of Statistics*, pages 1155–1202, 2005.

[68] Tianshi Gao and Daphne Koller. Active classification based on value of classifier. *NIPS*, 2011.

[69] Steve Branson, Catherine Wah, Florian Schroff, Boris Babenko, Peter Welinder, Pietro Perona, and Serge Belongie. Visual recognition with humans in the loop. In *Computer Vision–ECCV 2010*, pages 438–451. Springer, 2010.

[70] Sergey Karayev, Tobias Baumgartner, Mario Fritz, and Trevor Darrell. Timely object recognition. *NIPS*, 2012.

[71] Bogdan Alexe, Nicolas Heess, Yee Whye Teh, and Vittorio Ferrari. Searching for objects driven by context. *NIPS*, 2012.

[72] Christoph H Lampert, Matthew B Blaschko, and Thomas Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(12):2129–2142, 2009.

[73] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition*, 2014.

[74] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision (ECCV)*, 2014.

[75] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild.

[76] Devi Parikh, C Lawrence Zitnick, and Tsuhan Chen. Exploring tiny images: the roles of appearance and contextual information for machine and human object recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(10):1978–1991, 2012.

[77] Jiri Najemnik and Wilson S Geisler. Optimal eye movement strategies in visual search. *Nature*, 434(7031):387–391, 2005.

[78] Elisabeth Moores, Liana Laiti, and Leonardo Chelazzi. Associative knowledge controls deployment of visual selective attention. *Nature neuroscience*, 6(2):182–189, 2003.

[79] Ross Girshick. Fast r-cnn. *arXiv preprint arXiv:1504.08083*, 2015.

[80] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.

[81] Christian Rupprecht, Loic Peter, and Nassir Navab. Image segmentation in twenty questions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3314–3322, 2015.

[82] Donald Geman, Stuart Geman, Neil Hallonquist, and Laurent Younes. Visual turing test for computer vision systems. *Proceedings of the National Academy of Sciences*, 112(12):3618–3623, 2015.

[83] Marc'Aurelio Ranzato. On learning where to look. *arXiv preprint arXiv:1405.5488*, 2014.

[84] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 891–898. IEEE, 2014.

[85] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[86] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[87] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM, 2004.

[88] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[89] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[90] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. What is an object? In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 73–80. IEEE, 2010.