Context and Subcategories for Sliding Window Object Recognition

Santosh K. Divvala

CMU-RI-TR-12-17

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Robotics

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

August 2012

Thesis Committee
Martial Hebert, Co-Chair
Alexei A. Efros, Co-Chair
Takeo Kanade
Deva Ramanan, University of California at Irvine

Keywords: Object Category Detection, Sliding Window, Context, Subcategories

To my Parents and Teachers.
(without whom nothing would have been possible)

Abstract

Object recognition is one of the fundamental challenges in computer vision, where the goal is to identify and localize the extent of object instances within an image. The current de facto standard for building high-performance object category detectors is the sliding window approach. This approach involves scanning an image with a fixed-size rectangular window and applying a classifier to the features extracted within the sub-image defined by the window. In this thesis, we study two important factors influencing the performance of the approach.

First is the role played by context, where information outside the sliding window is used to rescore the detections output by the local window classifier. Context helps to suppress detections in regions that are less probable to contain an object and encourages those that are more plausible. In the first part of this thesis, we enumerate different sources and uses of context, and comprehensively evaluate their role in a benchmark detection challenge. Our analysis demonstrates that carefully used contextual cues serve not only to improve performance of local classifiers, but also to make their error patterns more meaningful and reasonable. Our analysis also provides a basis for assessing the inherent limitations of the existing approaches as well as the specific problems that remain unsolved.

The second factor is the role played by subcategories, where information within the sliding window is used to split the training data into smaller groups, for learning multiple classifiers to model the appearance of an object category. The smaller groups have reduced appearance diversity and thus lead to simpler classification problems. In the second part of this thesis, we analyze different schemes to generate subcategories and find that unsupervised feature-space clustering produces well-performing subcategory classifiers. Beyond performance gains, subcategories are attractive for their conceptual simplicity and computational tractability. For example, we find that careful use of subcategories can potentially replace the need for deformable parts within the state-of-the-art deformable parts model detector for many object categories. Data fragmentation is an important problem associated with subcategory-based methods. We present a novel approach that circumvents this problem by allowing different subcategories to share each other's training instances.

Acknowledgements

I have been very fortunate to be co-advised by two computer vision superstars, Martial Hebert and Alexei (Alyosha) Efros. I would like to thank Martial for being a great adviser, both on academic as well as non-academic matters. He has always supported me at critical junctures of my PhD and helped me survive them. There are many qualities that I would like to imbibe from him. Working with Alyosha has been a memorable experience. His charismatic style of advising has helped me chase hard research problems that I would have otherwise hesitated to pursue. I would like to thank him for making me a well-rounded researcher.

I would also like to thank Takeo Kanade. He has been my role model ever since I have first read about him. He has always been and will continue to remain my source of inspiration and motivation for pursuing fundamental research in Artificial Intelligence. I would like to thank Deva Ramanan for providing his key insights into my work and offering suggestions that led to improvements.

CMU has been a fantastic environment for pursuing my PhD. It has been a great place that has helped me connect with talented individuals from various backgrounds. I would like to thank Ankur Dutta and Tomasz Maliswiecz. The interactions I have had with them on various aspects of computer vision, robotics, and life in general, have been very enriching and will be memorable. I would also like to thank Balaji Sukumar and Umashankar Nagarajan for all the long discussions I have had with them on various aspects of career and life. I would like to thank my racquetball buddy Brian Becker and the Hebert dinner group (Kate Taralova, Dan Munoz, Ed Hsiao, Scott Satkin) for making my life in RI enjoyable. I am also glad to have interacted with JF Lalonde, Sanjeev Koppal, Ram Ravichandran, Minh Nguyen, Debadeepta Dey, Henry Kang and Mark Desnoyer during the course of my life at RI.

I have been fortunate to experience two great internships. I would like to thank Larry Zitnick, Simon Baker and Ashish Kapoor at Microsoft Research, and Mei Han and Sergey Ioffe at Google for their support.

Contents

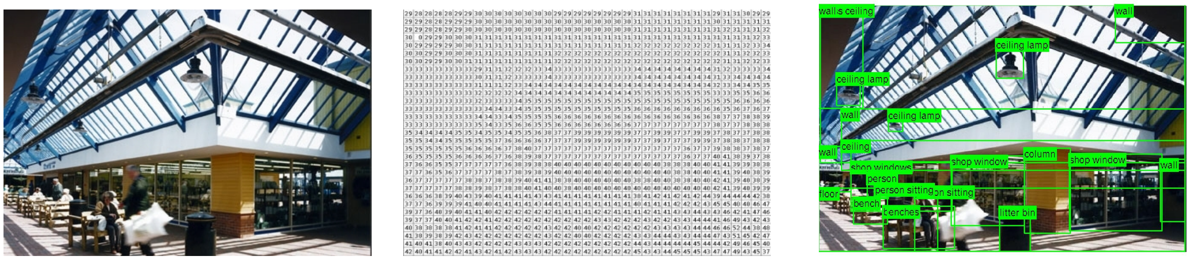
List of Figures

List of Tables

Chapter 1

Introduction

> "Of all the visual tasks we might ask a computer to perform, object recognition remains the most challenging; no one has yet constructed a system that approaches the performance level of a two year-old child. There is not even any consensus among researchers on when this level of performance might be achieved."
>
> Richard Szeliski, Computer Vision: Algorithms and Applications (2010)



(a) Single Input Image (b) What a robot perceives (c) What we want it to perceive

Figure 1.1: The object recognition problem: Given a single image, we humans are able to effortlessly identify and localize the extent of different objects within it. However a robot has a tough time parsing the raw input of numbers to generate meaningful representation of the image.

Object recognition is one of the fundamental challenges in computer vision, where the goal is to identify and localize the extent of object instances within an image, without the help of a human expert (See Figure 1.1). Despite decades of research, robustly identifying familiar objects (e.g., cat, person, sofa) in images and videos is still far beyond the capabilities of today's vision systems. Truly successful recognition systems will have a high impact in application domains as varied as personal robotics, human-computer interaction, health care, scientific image analysis, surveillance, biometrics, and image retrieval. The most popular and the current de facto standard for building high-performance object detectors is the sliding window approach. In this work, we explore this approach in greater detail, and study two important factors influencing its performance, namely context and subcategories.

1.1 Background

Object recognition is one of the fascinating abilities already possessed by humans at childhood. With a glance of an object, we are able to robustly identify it despite the appearance variation due to change in illumination, texture, pose, or occlusions. Moreover, we can easily generalize from looking at a set of objects to recognizing objects that have never been seen before. However, it is a challenging task to devise vision systems that match the cognitive abilities of humans.

General object recognition falls into two broad categories, namely instance recognition and category recognition. The former involves recognizing known rigid 2D or 3D objects such as locations or planar objects, potentially being viewed from a novel viewpoint, against a cluttered background, and with partial occlusions. The latter is the much more challenging problem of recognizing any instance of a particular general class such as “cat”, “car”, or “bicycle”. This thesis will primarily focus on the latter problem.

Several researchers have extensively studied the recognition problem for over four decades [37, 154]. Significant efforts have been devoted to develop a variety of representation schemes and algorithms aimed at recognizing generic objects in images taken under different imaging conditions (e.g., viewpoint, illumination, and occlusion). Central to all recognition approaches is how the regularities of images, taken under different lighting and pose conditions, are extracted and recognized.

Early attempts at object recognition were focused on using geometric models of objects to account for their appearance variation due to viewpoint and illumination change. The key idea is that the geometric description of a 3D object allows the projected shape to be accurately predicated in a 2D image under perspective projection, thereby facilitating the recognition process using edge or boundary information. Much attention has been devoted to extracting geometric primitives (e.g., lines, circles, etc.) that are invariant to viewpoint change. Nevertheless, it has been shown that such primitives can only be reliably extracted under limited conditions such as under controlled variation in lighting and viewpoint [103].

While much of the early work relied almost exclusively on geometric methods, modern recognition techniques are appearance-based, in which methods from statistical pattern recognition are applied to image descriptors. Tremendous progress has been achieved in the past few years, due in large part to the integration of new data representations with the effective models of classification procedures developed in the statistical machine-learning community. Classifiers such as k-nearest neighbor, support vector machines, and boosting have been applied to recognize objects from images [121, 148].

Amongst the numerous appearance-based recognition approaches, the sliding window approach is the most popular and the current de-facto standard [25, 43, 115]. The standard approach involves scanning an image with a fixed-size rectangular window at multiple scales and applying a classifier to the features extracted within the sub-image defined by the window. The higher scoring windows are hypothesized to be more probable of containing an object. Multiple detections may occur near the target region and are merged to obtain the final bounding box. The classifier that scores each window is trained using a supervised machine learning algorithm that is provided with positive (ground-truth) windows that contain the object instance and negative windows that do not have any overlap with an object instance.

The sliding window approach is currently the best performing approach in both recognition competitions (e.g., PASCAL VOC [39]) as well as commercial systems [58, 82, 127, 162]. Face detectors [127, 162] built into most of today's digital cameras to enhance auto-focus and into video conferencing systems to control pan-tilt heads are designed using this approach. Pedestrian detectors [58] used in automotive safety applications, e.g., detecting pedestrians and other cars from moving vehicles, are also motivated based on this approach. It is also popular in the general domain of biometrics; e.g., identity recognition in specialized images such as irises and fingerprints [82]. Given its high popularity and practical impact, in this work, we explore this approach in greater detail and study two important factors influencing the performance of the sliding window approach.

1.2 Challenges

Two important factors affecting the performance of the sliding window detector are *context* and *subcategories*.

Figure 1.2: Lack of Context: Can we predict what are the objects being represented, just given the information within the windows? (Answers in Figure 1.4)

Context

Consider the images in Figure 1.2. Can we guess what is the object being represented within each window? Given just the information within the window, there are many possibilities and it is difficult to predict the exact object identity. Nonetheless once the *context* around the window is revealed (see Figure 1.4), the task becomes quite easy. Clearly we humans use a wealth of information outside the window to reason about the objects in our visual world, while the sliding window detector has no access to this information, thus resulting in poor performance (false predictions).

From the above observation, it is evident that context should be made an integral part of the sliding window detector. There is indeed a broad agreement in the computer vision community about the valuable role that context plays in any image understanding task. Numerous psychophysics studies (see [111] for an overview) have shown the importance of context for human object recognition. Several recent computer vision approaches have demonstrated that the use of context improves recognition performance [20, 55, 72, 75, 97, 106, 129, 140, 150, 166]. Yet, in practice, when a high-performance recognition system is required (e.g., for commercial deployment or to enter a recognition competition), implementors almost always revert to the tried-and-true local sliding window approaches [26, 47].

We believe there are two reasons for such a disconnect. First, in all the previous work on context, there has been a lack of standardization in the experimental evaluation (choosing datasets, reporting results, etc.). Thus it becomes very difficult to compare the different approaches to each other and to the standard non-contextual baseline methods. Second, there is very little agreement in the literature about what constitutes "context", with poor differentiation between very simple types of context (e.g., using a slightly larger local window) and ones that are much more involved (e.g., scene type, geometry). As a result, it is unclear which, if any, of the contextual approaches might be worthwhile for any given task, and how much of an increase in performance they are likely to produce.

The focus of the first part of this thesis is to bring context into the mainstream of object detection research by providing an empirical study of the different types of contextual information on a standard, widely used test set. This provides a basis for assessing the inherent limitations of the existing paradigms and also the specific problems that remain unsolved.

Subcategories

The key ingredient for the success of the sliding window detection approach is the classifier that scores features extracted within the window. The standard procedure to train the classifier is using a supervised machine learning algorithm. The learning algorithm is typically formulated as a binary classification problem in which the positive examples are bounding boxes of a specific object or scene category and negative examples are background patches (See Figure. 1.3). However, due to large intra-class variation in object appearance, object pose, and camera viewpoint, it becomes difficult to learn a single linear classifier that can achieve good performance on a challenging test set.

To deal with this intra-category diversity problem, one common approach is to improve the feature representation, or to increase the number of features used and to use a more powerful classifier such as a non-linear SVM. Another alternative is to simplify the basic-level category recognition problem by reorganizing the data into smaller groups. The smaller groups have reduced appearance diversity and thus lead to simpler classification problems. The grouping is often based on extra semantic (ground-truth) annotations, such as bounding-box aspect ratio, object viewpoint or pose, taxonomy, etc. The problem with semantic subcategories is that there are an infinite number of ways to partition a basic-level category into subcategories. For example, meaningful car subcategories can be based either on object pose (e.g., left-facing, right-facing, frontal), or car manufacturer (e.g. Subaru, Ford, Toyota), or some other functional attribute (e.g., sports car, utility vehicle, limousine). It is thus unclear what single subcategory scheme should be used for devising a high-performance recognition system.

In the second part of this thesis, the problem of discovering subcategories from a basic-level category in an unsupervised fashion is studied. Instead of using semantic metadata to drive the creation of these subcategories, we let the feature representation and classifier type suggest its

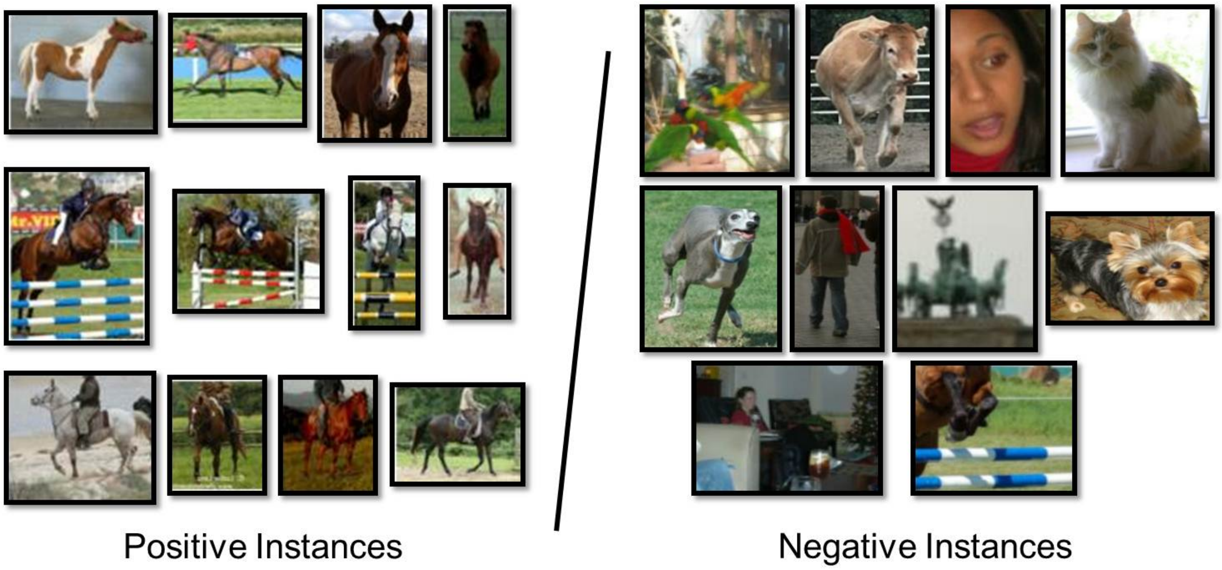


Figure 1.3: There is wide visual variability within a single semantic category (e.g. 'horse'), that prevents existing methods from learning a good discriminative object model. Notice the huge variation in the appearance, shape, pose and camera viewpoint of the different instances – there are left and right-facing horses, horses jumping over the fence in different directions, horses carrying people in different orientations, and close-up shots, etc.

own optimal clustering. Beyond improvements in detection performance, we will see that the use of subcategories offers other benefits hitherto unavailable in the case of a single monolithic classification procedure.

1.3 Thesis Overview

Part I of this thesis is dedicated to understanding the role of context. Chapter 2 begins with the introduction of a taxonomy of context, where various sources of contextual information and their uses in object detection are detailed. It then presents an approach for modeling the contexts, explaining the procedure for extracting contexts from multiple data sources, and training classifiers for rescoring local window detection results. Comprehensive analysis on a benchmark dataset is subsequently described. The chapter ends by describing a novel approach to include contextual information while training a local window detector to address a key problem with context-driven detection approaches. Chapter 3 is about an application of context to another important computer vision problem. It first introduces the notion of unsupervised patch-based context and describes an approach to extract useful contextual information from large collection of unlabeled web images. Results on two important image parsing tasks, namely surface layout estimation and semantic region classification, are presented.

Part II of this thesis focuses on understanding the role of subcategories. Chapter 4 introduces the notion of subcategories and compares the utility of unsupervised vs. supervised subcategories. It then presents an approach for learning subcategories in an unsupervised setting using a latent SVM approach. Experimental results analyzing the importance of subcategories for improving sliding window detection performance are then described. An analysis comparing the role of subcategories to deformable parts (another popular tool for addressing intra-class diversity) is subsequently presented. An important benefit of adapting the feature representation that is specifically offered by the use of subcategories is discussed. The chapter ends by describing an application of subcategories to scene classification (another important computer vision problem). Chapter 5 addresses the problem of data fragmentation transpired by the use of subcategories. It presents a novel approach for generating additional training data by reusing existing training samples in two different ways, by shrinking and enlarging ground-truth boxes.

Finally, Chapter 6 presents our conclusions, and describes a few potential areas of future investigation.

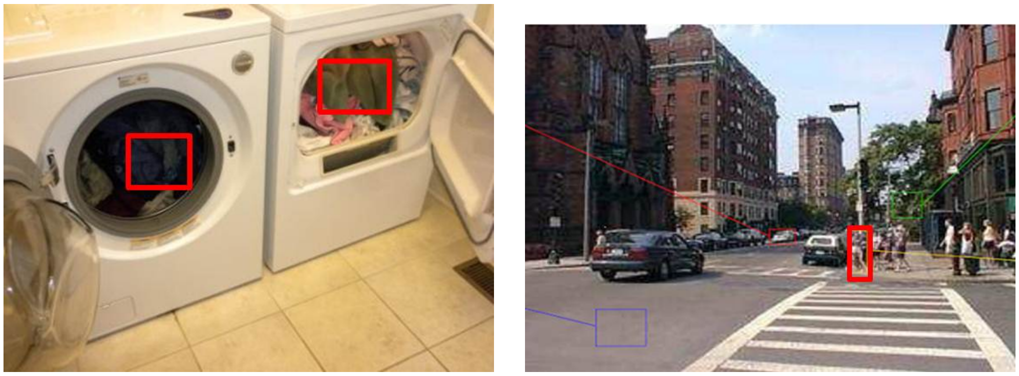


Figure 1.4: Importance of Context: Humans use a wealth of information outside the window to predict the object within the window.

Part I: Thinking Outside the Window

> “Nothing limits achievement like small thinking; nothing expands possibilities like unleashed imagination.”
>
> William Arthur Ward

Chapter 2

Role of Context

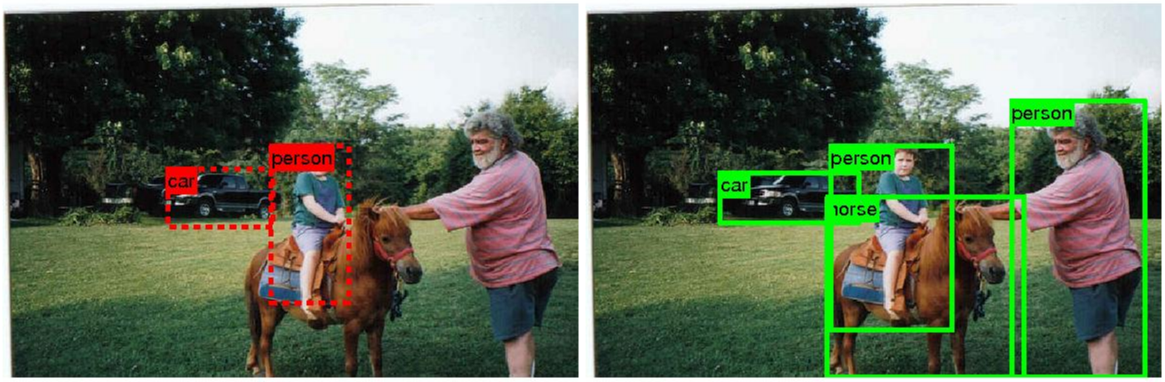


Figure 2.1: On the challenging PASCAL VOC dataset, even the best local-window detectors [47] often have problems with false positives, poor localization, and missed detections (left). In this work, we enhance these detectors using contextual information (right). Only detections above 0.5 precision are shown. (Red Dotted: Detector, Green Solid: Detector+Context)

There is a broad agreement in the community about the valuable role that context plays in any image understanding task. Numerous psychophysics studies (see [111] for an overview) have shown the importance of context for human object recognition. Several recent computer vision approaches have demonstrated that the use of context improves recognition performance [20, 55, 72, 75, 97, 106, 129, 140, 150, 166]. Yet, in practice, when a high-performance recognition system is required (e.g., for commercial deployment or to enter a recognition competition), people almost always revert to the tried-and-true local sliding window approaches [26, 47].

[1] Parts of this work have been described in Divvala et al. [35].

Why such a disconnect? We believe there are two reasons. First, in all the previous work on context, every approach reported results only on its own, home-grown dataset. Because of this lack of standardization, it becomes very difficult to compare the different approaches to each other, and to the standard non-contextual baseline methods. Second, there is very little agreement in the literature about what constitutes "context," with poor differentiation between very simple types of context (e.g., using a slightly larger local window) and ones that are much more involved (e.g., scene type, geometry). As a result, it is unclear which, if any, of the contextual approaches might be effective for any given task, and how much of an increase in performance are they likely to produce.

The goal of this work is to provide an empirical study of the different types of contextual information on a standard test set. This provides a basis for assessing the inherent limitations of the existing approaches and also the specific problems that remain unsolved. The main contributions are as follows: *1) Objective evaluation of context in a standardized setting.* We chose to conduct our experimental evaluation in the context of the PASCAL VOC Detection Challenge [39] – by far the most difficult, of all object detection datasets. As the baseline local detector, we choose from amongst the top-performing detectors in this challenge. Experimental results demonstrate that carefully used contextual cues can not only make a very good local detector perform even better, but also change the typical error patterns of the local detector to more meaningful and reasonable errors. *2) Evaluation of different types of context.* In this study, we look at several sources of contextual information, as well as different ways of using this information to improve detection performance. *3) Novel algorithms.* While we employ several contextual cues that have been used before, we also propose a few new approaches, including the use of geographic context and a new approach for using object spatial support.

2.1 Taxonomy of Context

2.1.1 Context Sources

While the term "context" is frequently used in computer vision, it lacks a clear definition. It is informally understood as "any and all information that may influence the way a scene and the objects within it are perceived" [145]. Many different sources of context have been discussed in the literature [14, 111, 145] and others are proposed here (see Table 2.1 for summary). The most common is what we broadly term *local pixel context*, which captures the basic notion that

Local Pixel Context	window surround, image neighborhoods, object boundary/shape
2D Scene Gist Context	global image statistics
3D Geometric Context	3D scene layout, support surface, surface orientations, occlusions, contact points, etc.
Semantic Context	event/activity depicted, scene category, objects present in the scene and their spatial extents, keywords
Photogrammetric Context	camera height, orientation, focal length, lens distortion, radiometric response function
Illumination Context	sun direction, sky color, cloud cover, shadow contrast, etc
Weather Context	current/recent precipitation, wind speed/direction, temperature, season, etc.
Geographic Context	GPS location, terrain type, land use category, elevation, population density, etc.
Temporal Context	nearby frames (if video), temporally proximal images, videos of similar scenes, time of capture
Cultural Context	photographer bias, dataset selection bias, visual clichés, etc

Table 2.1: Taxonomy of sources of contextual information.

image pixels/patches around the region of interest carry useful information. The classic trick of increasing the size of a scanning-window detector to include surrounding pixels [26, 166] is one simple implementation of local pixel context, as are more involved MRF/CRF-based methods, such as [20, 84, 140]. Image segmentation, object boundary extraction, and various object shape/contour models are also examples of local pixel context, as they use the object's surroundings to define its shape/boundary [123]. *2D scene gist* uses global statistics of an image to capture the "gist" of the visual experience [109, 129]. *Geometric context* aims to capture the coarse 3D geometric structure of a scene, or the "surface layout" [74], which can be used to reason about supporting surfaces [75], occlusions [73], contact points, etc. *Semantic context* might indicate the kind of event, activity, or other scene category being depicted [13, 92, 109]. It also may indicate the presence and location (spatial context) of other objects and materials [54, 55, 65, 143]. *Photogrammetric context* describes various aspects of the image capturing process, such as intrinsic camera parameters, i.e., focal length, lens distortion, radiometric response [97], as well as extrinsic, i.e., camera height and orientation [75]. *Illumination context* captures various parameters of scene illumination, such as sun direction [86], cloud cover, shadow contrast, whereas *weather context* would describe meteorological conditions such as current/recent precipitation, wind speed/direction, temperature, season as well as conditions of fog and haze [108]. *Geographic context* might indicate the actual location of the image (e.g. GPS), or a more generic terrain type (e.g., tundra, dessert, ocean), land use category (e.g. urban, agricultural), elevation, population density, etc. [70]. *Temporal context* would contain temporally proximal information, such as time of capture [53], nearby frames of a video (optical flow), images captured right before/after the given image, or video data from similar scenes [95]. Finally, there is what we broadly term the *cultural context*, a largely neglected aspect of context modeling. Its role is to utilize the multitude of biases embedded in how we take pictures (framing [141], focus, subject matter), how we select datasets [120], how we gravitate towards visual clichés [134], and even how we name our children [52]!

2.1.2 Context Uses

While in the previous section we cataloged the many possible sources of context that could be available to a vision system, what we are primarily interested is how context can be used for the task of object detection. Let us now consider the different aspects of an object detection architecture to see how contextual information could be used.

Object Presence. Many objects occur in typical environments, such as toasters in kitchens or moose in woodlands. The appearance of the scene (gist context), its layout (geometric context), scene or event category/the presence of other objects (semantic context), previous scenes (temporal context) can all help in predicting the presence of an object. Moreover, some objects tend to appear in certain parts of the world (geographic context), and some objects are more likely to be photographed than others (cultural context). Object presence is roughly equivalent to the *probability* constraint proposed by Biederman [14].

Object Appearance. The color, brightness, and shading of an object will depend on scene illumination and weather. For example, the measured color of a green apple during sunset (when the ambient illumination is red) would be somewhat different from its true color. However, once the color of the illumination source is provided, the true color can be inferred correctly [147]. Camera parameters such as exposure and focal length (photogrammetric context) can help explain intensity and perspective effects.

Object Location. 3D physical constraints, such as objects requiring a ground plane or some other support surface, help to determine likely locations of objects in the scene (geometric context). Moreover, some objects are likely to appear near others, such as people near other people, or in particular relations to objects or materials, such as cars on the road, squirrels in trees, grass below sky, etc (semantic context). Presence of an object at a particular location in nearby scenes can help predict its location in a future scene (temporal context). Photographer biases (cultural context) often provide useful information, such as an object being centered in the image due to photographer framing and its bottom position to be towards the bottom of the image due to roughly level imaging. Object location is roughly equivalent to Biederman's *support* and *location* constraints [14].

Object Size. Given object presence and location, its size in the image can be estimated. This requires knowing either camera orientation and height above the supporting surface (photogrammetric context), or relative sizes of other known objects in the scene (semantic context) and their geometric relationships (geometric context). Object size is roughly equivalent to Biederman's *size* constraint [14].

Object Spatial Support. Spatial support refers to the *local pixel* context (Section 2.1.1) around an object, including its segmentation and boundary information. Object segment or boundary is a type of context as boundaries do not exist in isolation. They are defined both by the object as well as its surround. Moreover, many objects do not "own" their boundaries. For example, the boundary of a grass region is not defined by the grass but by the objects that surround it. Given object presence, location and size in the image, its spatial support can be estimated in order to: 1) better localize a bounding box; 2) perform more accurate non-max suppression and multiple object separation (by using segment overlap instead of bounding box overlap); 3) estimate a more precise object shape and appearance model. Estimating the spatial support of an object can be assisted by a number of contextual cues. Local image evidence, such as contours/edges, areas of similar color or texture (local pixel context), occlusion boundaries and surface orientation discontinuities (geometric context), as well as class-specific shape prior (semantic context) can all provide valuable information. This use of context is roughly equivalent to Biederman's *interposition* constraint [14].

2.2 Modeling of Context

In the previous section, we generated a full wish list of contextual cues and their uses that can potentially benefit object detection. In designing our approach, we picked the context cues which could not only be reliably learned given the available data, but also fit the "plug-and-play" philosophy of taking an off-the-shelf local detector and adding contextual information to it. Therefore, in this work, we have used local pixel context, 2D scene gist, 3D geometric, semantic, geographic, photogrammetric and, to a limited extent, cultural context cues, while finding that we did not have good training data for the others. Based on these available context sources, we have implemented object presence, location, size, and spatial support uses of context.

To fairly evaluate the role of context, we need to start with a good local detector. In this work, we use the UoCTTI [47] detector, which has been the top-performing PASCAL VOC challenge [39] detector. Qualitatively, we have observed that the detector achieves substantially better results than that suggested by the raw performance numbers. This is because, although the detector does a fair job in detecting the presence of an object correctly, it often makes mistakes in localizing it, partially due to the fixed aspect ratio of the bounding box and multiple firings on the same object. Thus, some false positives are due to mistakes in the appearance model but others are due to poor localization. We attempt to overcome these problems by augmenting the detector with contextual information.

In this work, we use the detector trained on the VOC'07 trainval set, and use the VOC'08 trainval set for learning the context classifiers (described below). This ensures that the baseline detector and context are trained on different datasets to avoid overfitting. To help ensure that few true detections are missed by the detector, we reduce the threshold for detection such that there are at least 1000 detections per image per object .

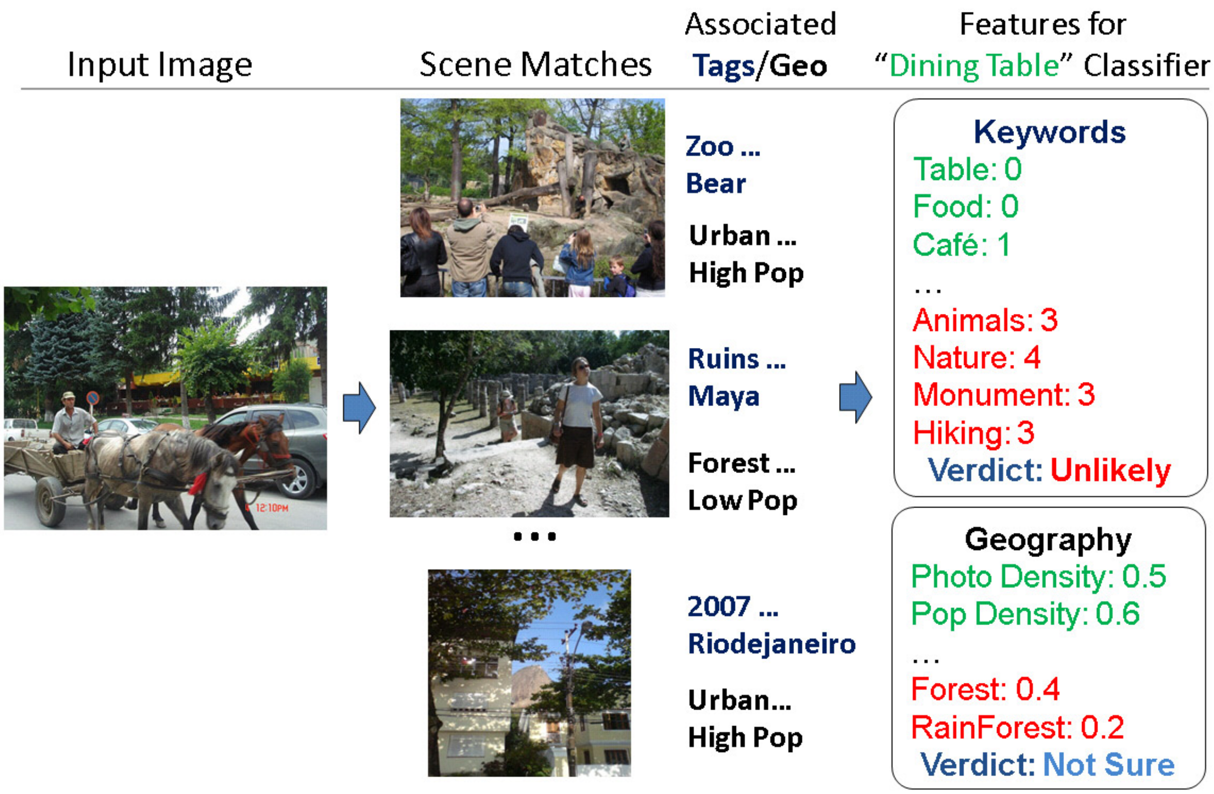


Figure 2.2: Geographic and Semantic (keyword) context: Geographic properties and keywords associated with the scene can help predict object presence in an image. The base detector finds a dining table in this input image (see Figure 2.6), while the context indicates that a dining table is unlikely.

2.2.1 Object Presence

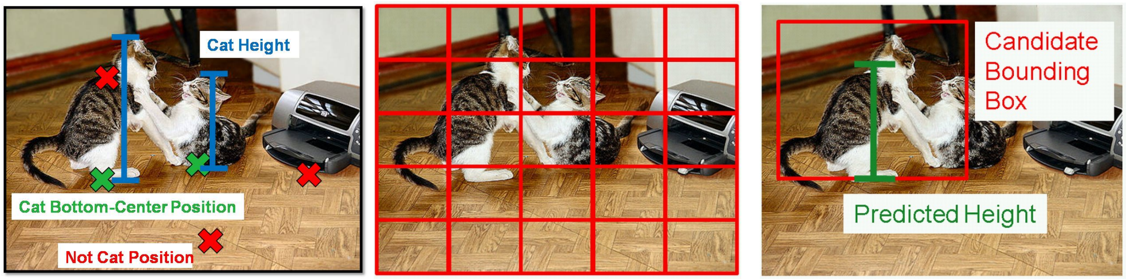
To predict the likelihood of observing an object o given the image I i.e., $P(o|I)$, we use the 2D scene gist, 3D geometric, semantic and geographic contexts. The 2D scene gist of an image is computed in the standard way as described in [109]. The geometric context for an image is computed as a set of seven geometric class (ground, left, right, center, sky, solid, porous) confidence maps as described in [74]. These confidence maps are re-sized to 12×12 grids and vectorized to serve as a coarse "geometric gist" descriptor. We use logistic regression [81] to train two separate object presence classifiers based on each descriptor. The use of these descriptors for scene classification has become fairly standard in literature and has shown good results. However, our use of geographic and semantic information is a novel contribution.

For the geographic context, we follow the approach of [70], estimating geographic properties for a novel image by finding matching scenes within a database of approximately 6 million geotagged Flickr photographs (excluding images that overlap with the VOC dataset). We compute 15 geographic properties such as land cover probability (e.g., ‘forest’, ‘cropland’, ‘barren’, or ‘savanna’), vegetation density, light pollution, and elevation gradient magnitude. We train a logistic regression classifier based on these geographic properties. Object class occurrence is correlated with geography (e.g., ‘boat’ is frequently found in water scenes, ‘person’ is more likely in high population density scenes) but the relationship is often weak. For instance, the ten indoor object classes in the VOC dataset cannot be well distinguished by geography.

For semantic context, we use the keywords associated with matching scenes in the im2gps dataset [70] to predict object occurrence. The 500 most popular words appearing in Flickr tags and titles were manually divided into categories corresponding to the 20 VOC classes and 30 additional semantic categories. For instance, ‘bottle’, ‘beer’, and ‘wine’ all fall into one category, while ‘church’, ‘cathedral’, and ‘temple’ fall into another category. For a novel image we build a histogram of the keyword categories that appear among the 80 nearest neighbor scenes (Figure 2.2). We use logistic regression to predict object class based on this histogram. Keywords from Internet images are very noisy and sparse (the im2gps database averages just one relevant keyword per image), but they are quite discriminative when they do occur. All the above classifiers are trained on the VOC’08 trainset.

2.2.2 Object Location

The goal is to predict *where* an object is likely to appear in an image given that there is at least one object occurring in the image i.e., $P(x|o, I)$. To train this location predictor, we divide the image into $n \times n$ grid ($n = 5$) and train for each cell in the grid, two separate logistic regression classifiers [81], one that trains on the whole image scene gist descriptor and the other that trains on the whole image 3D geometric context descriptor. The classifiers are trained using the VOC’08 trainset. A grid is labeled as a positive example if the bottom mid-point $(\frac{x_{left}+x_{right}}{2}, y_{bottom})$ of a bounding box falls within it (Figure 2.3). We then combine the predictions of the above two classifiers using another logistic regression classifier trained on the VOC’08 validation set. For some classes, a few grid cells end up having no (or very few) positive examples (e.g., dining tables never occur in the (1,1) grid). No classifiers were trained for such grid cells, and the confidence of finding an object in this location was set to a minimum value while testing.



Location

Size

Figure 2.3: Object properties such as bottom-center position and height are used for modeling object location (Section 2.2.2) and object size (Section 2.2.3) respectively.

2.2.3 Object Size

The idea here is to predict the size (as log pixel height) of an object, given its location in the image i.e., $P(h|x, o, I)$ as illustrated in Figure 2.3. This is learned using three types of contextual cues: 1) photogrammetric context modeled in terms of viewpoint estimates [75] (relative y-value) and the object depth [73] (value at the bottom mid-point of an object bounding box); 2) 2D scene gist; and 3) 3D geometric contexts (the latter two modeled as whole image descriptors). We train a separate logistic regression classifier on the VOC'08 trainset for each of the above feature descriptors. This regression task is reformulated as a series of classification tasks [106], where we first cluster object sizes (using K-means) into five clusters s_1, s_2, s_3, s_4, s_5 and then train a separate classifier for each size (i.e., size $< s_2$, size $< s_3$, size $< s_4$, size $< s_5$). The object sizes for training classifiers are calculated using the ground-truth annotations provided in the VOC'08 dataset. The predictions from individual classifiers are combined using another logistic regression classifier trained on the VOC'08 validation set. At testing, we calculate $P(size = k)$ as $P(size < k + 1) * (1 - P(size < k))$, with $\sum_k P(size = k) = 1$ and compute the expected object size as $\sum_k P(size = k) * center(k)$.

2.2.4 Object Spatial Support

The task here is to compute the object's spatial support given an (often poorly localized) candidate detection and its confidence. This is a much easier problem than the general segmentation problem because the type of object and its rough location in the image is known. We implement a simple segmentation approach based on graph cuts.

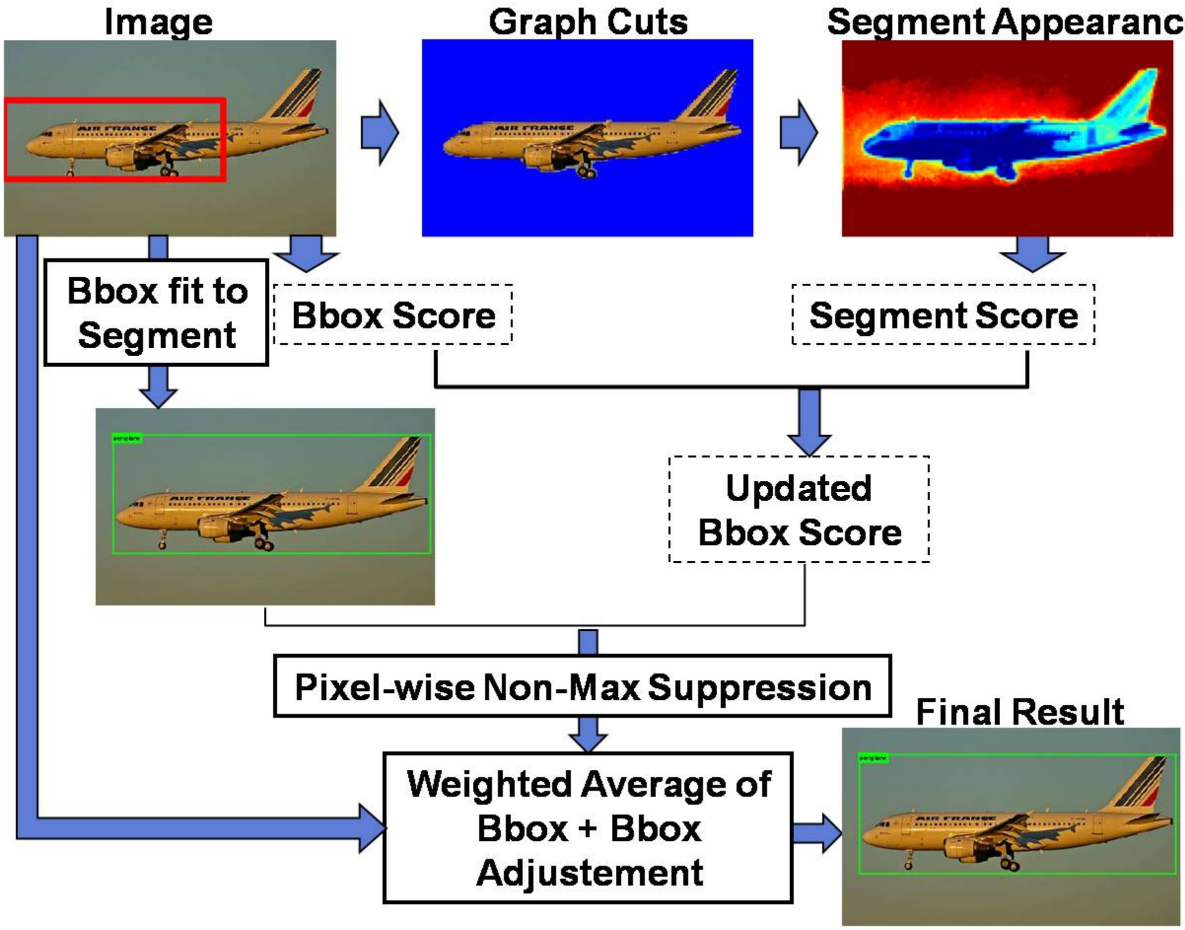


Figure 2.4: Modeling Object Support: We apply graph cuts segmentation to each bounding box for (i) improving the box localization, (ii) rescoring the box using region-based features. Further we perform more accurate non-max suppression and multiple object separation by using segment overlap instead of bounding box overlap.

Unary and Pairwise Features: Our unary features model the object class appearance, a position/shape prior, and the object instance appearance. For class appearance, we compute K-means clustered $L*a*b$ color (K=128) and texton [155] (K=256) histograms, geometric context confidences [74] and the probability of background confidences (trained using [74] on LabelMe [130] examples), quantized to ten values. The features are the class-conditional log-likelihood ratios i.e., $\log \frac{P(feature|object)}{P(feature|background)}$, given the quantized value, as estimated on the segmentation groundtruth in the VOC'08 trainset. The position/shape prior is computed as the log-likelihood ratio for each pixel given its location with respect to the location and scale of the bounding box. The object instance appearance is modeled by taking the log ratio of the histograms computed within and outside the bounding box. Altogether, this gives us thirteen features (class appearance: color, texture, seven geometric classes, probability of background; location/shape prior; instance appearance: color, texture), plus a prior.

For the pairwise features, we use the probability of boundary (Pb) [98] and probability of occlusion [73] confidences.

Learning: The potentials for the unary and the pairwise term are defined using a function of the form $w \cdot f$, where f indicates the feature, and w indicates the weight vector learned discriminatively from the training data. (We use L_1 regularized logistic regression to learn the weights.) Separate weights are learned for horizontal, vertical, and diagonal neighbors (eight-connected neighborhood) in case of the pairwise potential.

Inference: Each candidate detection is segmented using graph cuts [18], after resizing the image so that the object length is 100 pixels. (The resizing is important to achieve good segmentations for objects of different sizes). For computational reasons, only post-context detections that are above a threshold (corresponding to 0.025 precision in validation) are processed. See Figure 2.4 for an illustration.

After segmenting an object, we represent the segment appearance with four features: histograms of K-means quantized color f_c (K=128), texture [155] f_t (K=256), and HOG features [26] f_h (K=1000), and a measure of segmentation quality f_s. We define f_s as $\frac{E_{gc}-E_{bgrnd}}{N_o}$, where E_{gc} is the energy of the graph cut solution (from the inference step described above), E_{bgrnd} is the energy when all pixels are labeled as background, and N_o is the number of object pixels. The K-means cluster centers for computing the histograms are estimated using features extracted on the VOC'08 trainset. Altogether, this gives us a 1385-dimensional (128+256+1000+1) feature representation for every segment. A classifier on these segment-based features is trained using a linear SVM [79] for each object class. When testing, we reclassify the object based on the features computed within the segment and assign the final detection score as a linear combination of the original score and this segment-based score. This is similar to the segmentation-based verification strategy of Ramanan [123], who instead uses the pixels of the segmentation mask as features.

Beyond rescoring, we also use the computed spatial support to improve non-maximum suppression and localization. If two candidate detections yield segmentations with pixel overlap (intersection over union) of at least 0.5, the candidate with the lower score is removed. A new bounding box is estimated by taking a weighted average of the original bounding box and a tight fitting box around the segment. The box is then adjusted by a fixed percentage of width or height to account for bias (e.g., consistently undersegmenting the legs of chairs). Parameters are learned on the validation set. For few classes (sofa, bicycles), the spatial support cannot be reliably estimated, resulting in a decrease in performance. To avoid this, a per-class parameter is learned on the validation set to decide if the rescoring/improved localization step is applied during the testing phase.

2.2.5 Combining Contexts

The task here is to combine the object detection results with the various context uses, so as to rescore those detection hypotheses that do not agree with the object presence, location and size context predictions to a lower value. Detections that occur at unusual poses should have significantly high score from the base detector for them to be selected in this scheme [106]. First we retrieve the top 100 detections (after non-max suppression) per image for all the training images. For each detection, we retrieve: 1) object presence estimates in terms of the scene gist, geometric context, geographic and semantic context classifier confidences; 2) object location estimates in terms of the confidence of the grid in which the bottom center of the bounding box occurs and also the max confidence in its neighborhood; 3) object size estimates in terms of the predicted height and the negative absolute difference between the bounding box height and the predicted height. We train a logistic regression [81] classifier using the above features on the VOC'08 validation set. We consider a detection hypothesis to be positive if there is at least 50% overlap with a true detection. If any of the above context features are assigned a negative weight during the training process, we retrain the classifier again after setting those features to zero. While testing, we retrieve the top 500 detections for every image (obtained using [47]) and rescore these detections using the above classifier. These rescored detections are used by the object spatial support context described in Section 2.2.4.

In all cases, we evaluate different classifiers for modeling the various contexts and also for combining them - kNN, SVM (linear and RBF) [79], logistic regression (L1 and L2). We found L1-regularized logistic regression to perform the best.

Objects	UoCTTI 2007	+Context		UoCTTI 2008	+Context	
		+Scene	+Scene +Support		+Scene	+Scene +Support
plane	18.8	21.3	**34.5**	28.7	26.8	**32.7**
bike	**33.5**	31.7	32.7	**44.6**	42.9	42.9
bird	9.3	9.9	**12.3**	0.5	**5.0**	5.0
boat	10.4	10.6	**11.0**	12.6	**13.1**	13.1
bottle	22.9	**23.2**	22.4	**28.8**	27.8	27.8
bus	**19.2**	17.7	18.5	22.7	**23.9**	23.9
car	25.1	26.0	**27.8**	**31.9**	31.6	31.6
cat	6.7	15.8	**21.6**	14.4	18.1	**19.8**
chair	13.3	**14.1**	8.8	15.9	**17.4**	17.4
cow	**16.6**	14.7	14.1	**14.4**	12.3	12.3
dtable	15.0	**18.4**	15.2	12.0	**21.4**	21.4
dog	6.3	7.9	**17.8**	**11.4**	7.7	9.4
horse	24.6	26.6	**27.4**	34.3	**35.7**	35.7
mbike	32.7	34.0	**40.9**	**37.7**	37.1	37.1
person	26.4	28.7	**37.4**	36.6	**39.5**	39.5
pplant	**11.2**	10.8	11.2	8.6	**12.6**	12.6
sheep	10.9	**12.0**	7.0	12.1	**13.5**	13.2
sofa	11.6	**13.7**	13.5	15.0	**15.8**	15.8
train	16.0	17.6	**28.2**	30.1	31.4	**32.2**
tv	32.9	33.3	**38.5**	34.7	**35.2**	35.2
Mean	18.2	19.4	**22.0**	22.4	23.4	**23.9**

Table 2.2: Detection Results on PASCAL VOC 2008 testset. The first column is the average precision (A.P.) obtained using the base detector. The second and third column show the A.P. obtained upon the addition of the scene context (object presence, location and size) and the spatial support context. Context aids in improving the detection results for many object classes.

2.3 Experimental Results

The PASCAL 2008 dataset [39] consists of roughly 10,000 images (50% test, 25% train, 25% validation) containing more than 20,000 annotated objects from 20 classes. The images span the full range of consumer photographs, including indoor and outdoor scenes, close-ups and landscapes, and strange viewpoints. The dataset is extremely challenging due to the wide variety of object appearances and poses and the high frequency of major occlusions.

Per-class detection results. Table 2.2 displays the detection results obtained on the VOC'08 test set with and without using context. The results are reported using the average precision (A.P.) metric, which is the standard mode of evaluation in the PASCAL VOC challenge. Our experiments show the importance of reasoning about an object within the context of the scene, as we are able to boost the average precision of the original UoCTTI'07 detector from 18.2 to 22.0. The table also includes a comparison with 2008 version of the UoCTTI detector to demonstrate the generalizability of the results across detectors. We also display the relative improvement obtained by the scene context (presence, location and size), and the spatial support context. We observe that both pieces of information contribute to the increase in performance (however, they cannot be compared on an absolute scale as the output of one process is the input to the other). Notice that for many classes there is a large improvement (e.g., airplane, cat, person, and train), while for some (e.g., bicycles, cows) there is a small drop in performance indicating that the benefit of context varies per class. It must be noted that our numbers cannot be directly compared to the official PASCAL VOC 2008 challenge rankings as our approach involves the usage of external datasets (VOC 2007 and Flickr images). Comparing the results obtained using the two different detectors reveals similar performance by our contextual information in either case. Therefore the rest of our analysis is conducted using the UoCTTI'07 detector on the VOC'08 validation set.

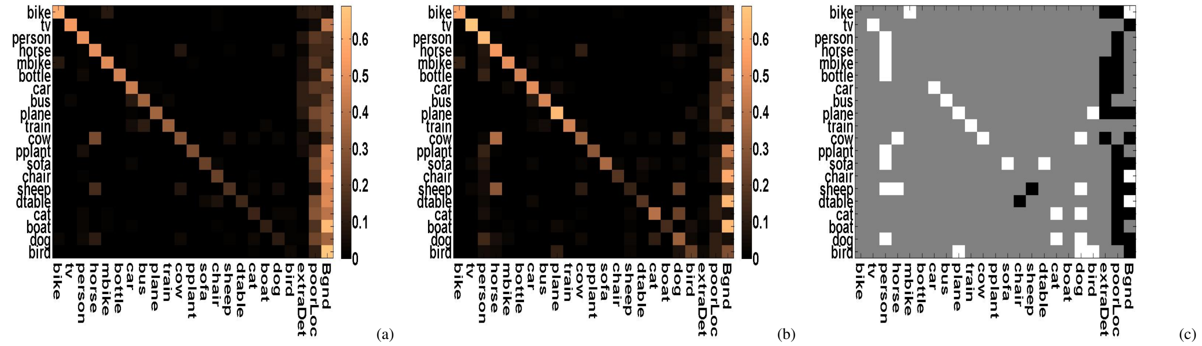


Figure 2.5: Confusion matrices (a) Without Context (b) With Context (c) Change in confusions i.e., (b-a) quantized into three values - white indicates positive change, black indicates negative change, and gray indicates negligible change (within +/- 0.05) . Observe that many fewer extra detections, localization errors, and background detections occur upon the addition of contextual information. Further, the remaining errors made are more reasonable – cows getting confused with horses, cats confused with dogs etc.

Change in confusion matrices. Figure 2.5 displays the change in the types of mistakes that are made after adding contextual cues. The confusion matrix is computed as usual, except that we include three new classes: 1) 'extraDet' addresses the scenario in which the overlap of a box is greater than 0.5 on an already detected object (extra detection); 2) 'poorLoc' includes scenarios in which overlap is between 0.25 and 0.5 (poor localization); and 3) 'Bgnd' denotes the case when the overlap is under 0.25 (fired on the background). Observe that there are much fewer extra detections (better non-max suppression), fewer localization errors, and fewer detections on background upon adding contextual information. Further, the remaining mistakes that occur after adding context are more reasonable where the confusions are between similar classes such as bicycles getting confused with motorbikes, buses with cars, cows with horses and sheep, etc.

Analysis of sources and uses of context. We measured the influence of each of the individual *sources of context* for the tasks of object presence, location and size estimation. For object presence ("Does the object appear in the image?"), the mean A.P. across 20 classes using individual cues was as follows: Semantic (25.6%), Gist (23.9%), Geometric (21.5%) and Geographic (15.1%), while using all the cues gave 31.2%. For object location ("In which of the 25 grids is the bottom of the object located?"), the mean A.P. across 20 classes was: Gist (3%), and Geometric (2.5%), while using both cues gave 6.5%. Finally for object size estimation, the average prediction error i.e., $\frac{\sum |log(trueHeight/predictedHeight)|}{\#instances}$ across 20 classes was: Photogrammetric (1.08), Gist (1.16) and Geometric (1.18) while using all the cues gave an error of 1.086. The baseline error of simply predicting the mean object height is 1.22.

To analyze the importance of the *uses of context* i.e., object presence, location and size, we run our detection experiments in a leave-one-out methodology. The mean A.P. across 20 classes for each of the case is as follows: 1) excluding object presence - 19.8%; 2) excluding object location - 20.2%, 3) excluding object size - 19.2%, 4) excluding all the three (i.e., simply running the base detector) - 18.5%, and 5) including all the three - 20.5%. Thus we observe that the object size context is the strongest, while object location is our weakest context use.

Change in accuracies with respect to size and occlusion. We also analyzed the change in accuracies as a function of two different object characteristics/types, namely occlusion and size (Table 2.3). The type 'occluded', 'non-occluded' and 'difficult' are as defined in the PASCAL annotations. The type 'small'/'large' refers to the object instances that were lesser/greater than the median object area in the image. All detections output by the local detector are considered valid for computing the accuracy. (Another alternative is to ignore detections on valid ground-truth that are excluded.) The accuracy is computed by pooling detection results from all the 20 VOC classes together i.e., computing the A.P. of a single precision-recall curve. (This is different

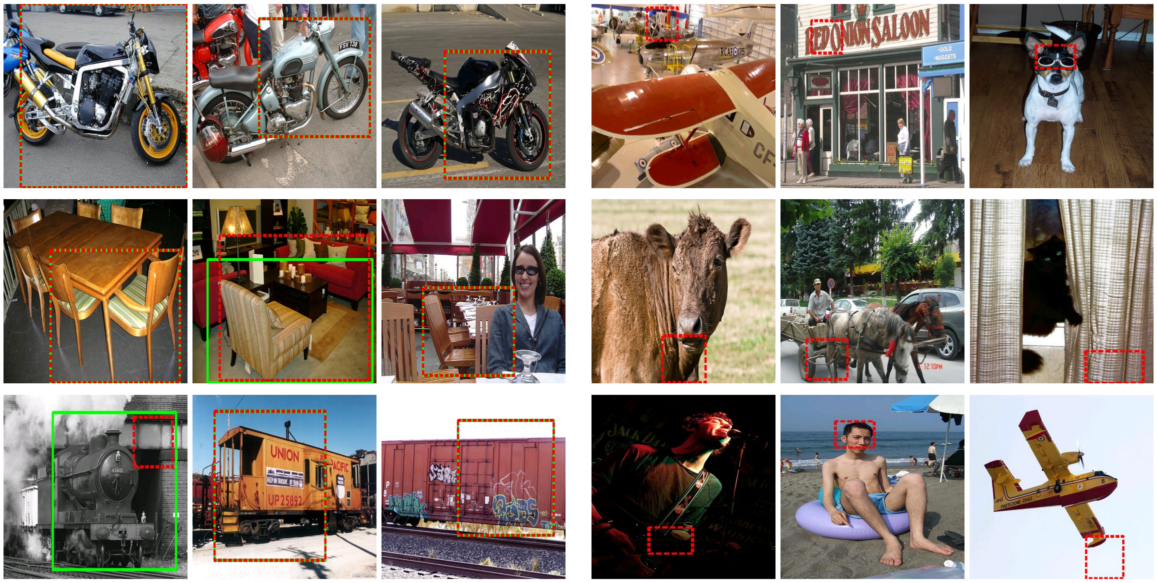
Type	Mean A.P.		Most Improved	Least Improved
	w/o	w/		
Small	6.7	12.0	planes (5.4 to 24.8)	pplant (10.3 to 5.9)
Large	9.3	9.7	dtable (4.5 to 9.3)	sheep (5.4 to 0.7)
Occluded	4.8	7.5	cat (3.1 to 13.8)	mbike (18.7 to 16.5)
Non-Occluded	10.4	11.5	dog (2.5 to 7.4)	chair (12.5 to 5.1)
Difficult	0.2	0.3	dtable (0.3 to 2.9)	chair (2.2 to 0.1)

Table 2.3: Average Precision w.r.t. two object types, Size and Occlusion. For each type, we display the mean A.P. across all object instances without ('w/o') and with ('w/') context along with most/least improved classes. Context particularly helps when objects have impoverished appearance.

from the numbers reported earlier where we compute the A.P. per class and then compute the mean A.P.) Since the classifier scores per object class may not be necessarily calibrated, the accuracy reported here will differ from (be lower than) the accuracy reported earlier. Context is particularly helpful when the objects have impoverished appearance, i.e., when they are small and occluded in the image.

We also analyzed the results by segregating objects into man-made vs. natural object categories. In this case, we observed that for natural objects (i.e. bird, cat, cow, dog, horse, person, sheep) the improvement in A.P. is 2.1 (from 14.4 to 16.5), while for man-made objects (i.e. aeroplane, bicycle, boat, bottle, bus, car, chair, diningtable, motorbike, pottedplant, sofa, train, tvmonitor), it is 0.8 (from 20.2 to 21.0).

Qualitative analysis. Figure 2.6 displays some of the qualitative results showing the largest increases and decreases in detection confidences after adding contextual information. Although context almost always helps in improving the detector performance, there are certain scenarios where it hurts. Figure 2.7 displays some cases where the addition of context leads to some of the original highly confident detections being discarded. Finally, in Figure 2.8, we display the mistakes/errors that still occur despite augmenting a top-performing detector with several contextual cues. Most errors are amongst classes that share similar contexts, e.g., cats confused with dogs, airplanes confused with birds, etc. Such confusions are subtle and present a challenge to the existing detection algorithms. In the next section, we will present an approach that can potentially alleviate such confusions.



Largest increase in confidence

Largest decrease in confidence

Figure 2.6: Images for the bike, diningtable, and train classes for which the best detections had the largest increase and decrease in confidence with the addition of context. In these cases the local appearance and global context disagree most strongly. When the addition of context increases confidence (left) it is because a detection is in a reasonable setting for the object class, even if the local appearance does not match well (motorbikes on top row share context with bicycles). When the addition of context decreases confidence (right) it is typically pruning away spurious detections that had high confidence scores from the local detector. (Red Dotted: Detector, Green Solid: Detector+context)

2.4 Towards Tying Objects and Context in a Loop

In the analysis presented in Section 2.3, we observed that the main remaining source of confusion affecting detection performance occurred amongst similar looking objects that shared similar contexts (Figure 2.8). For example bicycles and motorbikes have subtle differences in their appearance and incidentally share similar contexts; so is the case with cats and dogs, cows and horses, airplanes and birds, etc. In such scenarios, post processing candidate detections using context shows little benefit, and it is left to the local window detector to perform a good classification job. As detection datasets scale to contain more (thousands of) classes, the effect of such confusions would become more pronounced and would hinder higher recognition accuracies from being achieved.

Can such confusions ever be resolved? Indeed, they can be resolved as there have been detec-

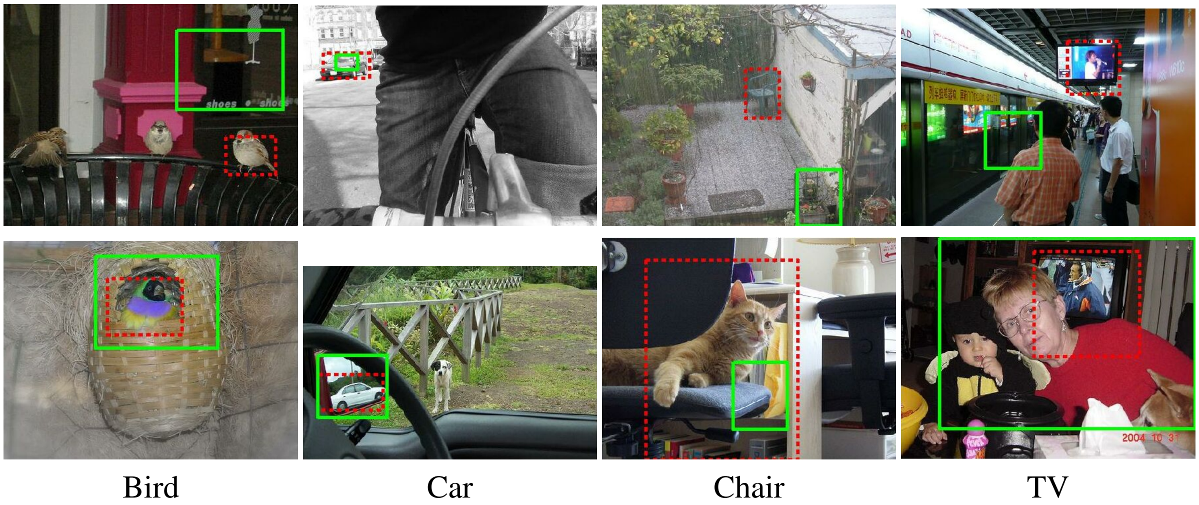


Figure 2.7: Examples where addition of context leads to some of the original highly confident detections being discarded. In each image, the red dotted bounding box is the highest scoring detection window (amongst the pool of all possible windows) before applying context, while the green solid box is the highest scoring window after applying context. Performance is hurt mostly in cases when the objects occur outside their typical context. Since objects are not in their typical context, the detection windows correctly localizing them are downgraded by the application of context.

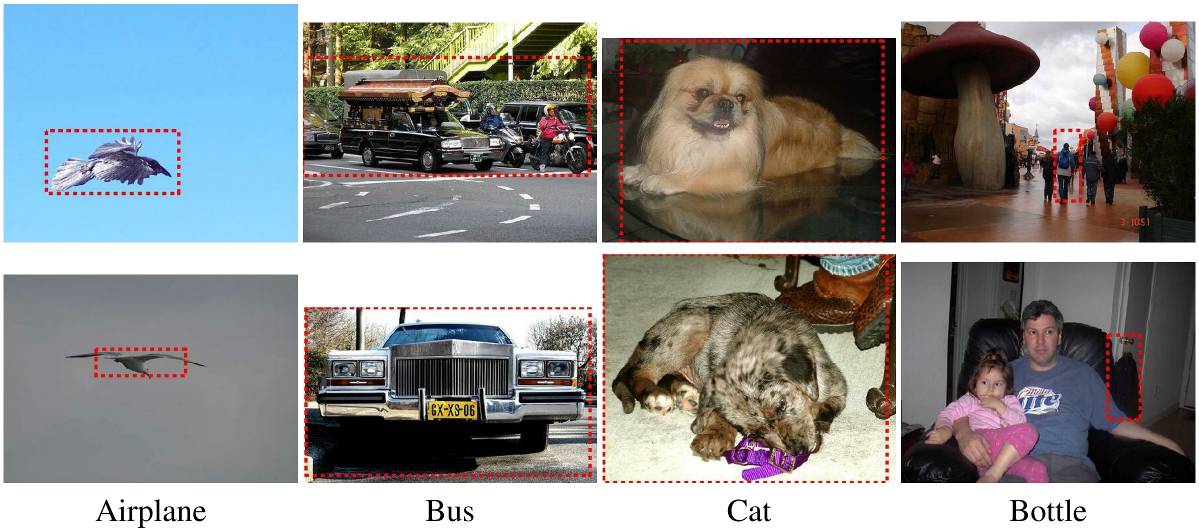


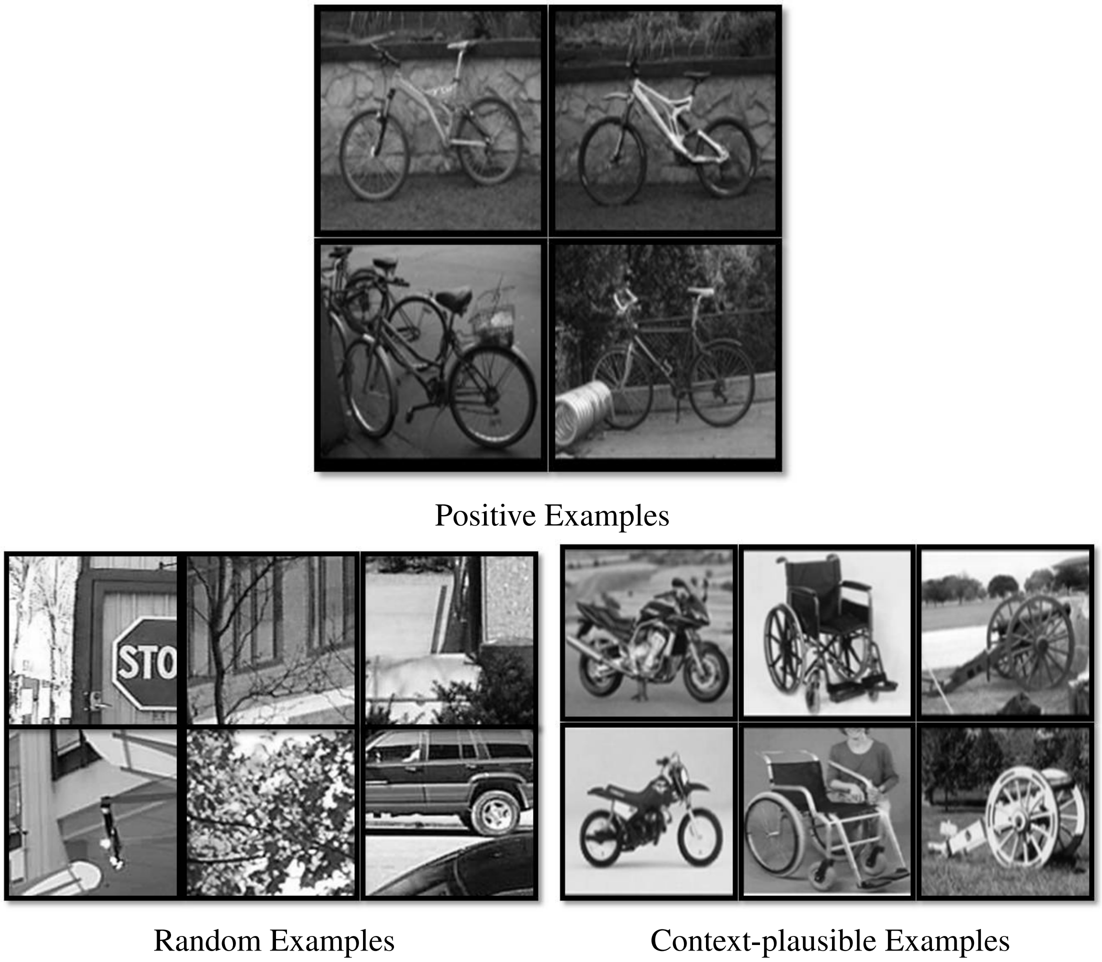
Figure 2.8: Mistakes/Errors made despite augmenting a top-performing object detector with several contextual cues. Such scenarios present a challenge to existing detection algorithms.

tion algorithms that have achieved high accuracies when specifically trained to classify similar objects e.g., bicycles against motorbikes [102], cows against horses [112], etc. This indicates that it can be possible to equip the detection models with discriminative ability for classifying similar classes when trained and tested in a restricted setting, but this ability is diminished in case of the general setting.

How can context be used to alleviate such confusions? One possibility is to explicitly supply similar and context-sharing object exemplars as negative instances while training the detector, e.g., instances of sheep, horse, goat, etc., as negative examples for a cow detector. However such a solution is neither appealing nor scalable as it requires hand-picking of negative exemplar classes for every object class. This may not even be possible for certain object classes, e.g., pedestrians are most often confused with parking meters, lampposts, mailboxes, etc., that may not even be annotated in an image database. Is it possible to instead automatically gather such negative instances for a given object class? Recall that during the training process, the detector is provided with the pool of all negative examples (taken across all possible locations and scales). Thus, it already has access to all the relevant i.e., context-plausible negative examples. Therefore, the problem is not due to the absence of such examples, but due to the presence of a gigantic amount of extraneous examples in its pool that prevents it from learning a robust discriminative classifier against confusing exemplars.

We propose to circumvent this problem by tying the detectors and the context "in a loop." Our proposal is as follows: Since context primes the focus of the object detector towards context-plausible image regions at test time, it is in fact sufficient to train the detector to discriminate the positive object instances from only the negative examples occurring in the context-plausible regions, rather than using negative examples taken from across all possible image regions, as done in the conventional settings. Incidentally, the negative examples in the context-plausible regions would encompass the set of all similar looking objects that share similar context with the given object class, e.g., a bicycle detector would find instances of motorbikes, carts, wheel chairs etc. in the bicycle's context-plausible image regions (see Figure 2.9). This provides an elegant framework where the detector is implicitly trained on the data that is drawn from the same distribution as testing, instead of wasting effort on irrelevant data that is out of context and will never be useful. Moreover this makes the training process computationally and memory efficient as there exist an order of magnitude fewer context-plausible image patches to be discriminated against, compared to the set of all possible image patches.

We refer to our proposed approach as context-aware detection, one where detectors are made *aware* of context at training time. This contrasts to previous section (which we refer as context-driven detection), where context was used to post-process detection results at testing time.



Positive Examples

Random Examples

Context-plausible Examples

Figure 2.9: Rather than training a detector in conventional sliding-window paradigm where negative instances are drawn randomly from training images, we only use the context-plausible examples, which helps achieve an efficient and improved detector.

Approach. For investigating the use of context at training time, we have chosen the task of pedestrian detection in natural outdoor scenes. More specifically, we take the pedestrian detector of Dalal and Triggs [26], along with the photogrammetric context of Hoiem et al. [75] (i.e., 3D camera viewpoint and horizon relations) and combine them to build a context-aware pedestrian detector. The detector trains on examples drawn from context-plausible regions, which is the same distribution that it *sees* at testing. Although our proposed hypothesis holds for any generic object category and relevant source of contextual cues, we believe reasoning about 3D relationships in case of pedestrians offers a strong source of context, providing us with a convenient framework to conduct our analysis.

Let y_i be the 3D height of an object with a 2D height of h_i and a vertical position of v_i (measured from the bottom of the image). The camera pose is denoted by y_c and v_0, which correspond to the camera height and the horizon position. Following [75], the inference is based on the relationship

$$y_i = h_i \frac{y_c}{v_0 - v_i}, \tag{2.1}$$

assuming that objects stand on the ground and that ground is not tilted from side to side, and it is roughly orthogonal to the image plane. Thus, given the position and height of an object category in the image, the camera pose can be estimated. Conversely, if the prior distribution over camera poses is available, it can be used to obtain a good estimate of object heights. In case of pedestrians, the height of people is normally distributed with a mean of 1.7 meters and a standard deviation of 0.103. The horizon is assumed to be at the middle of the image and the camera height is defaulted to 1.6 meters (roughly eye level).

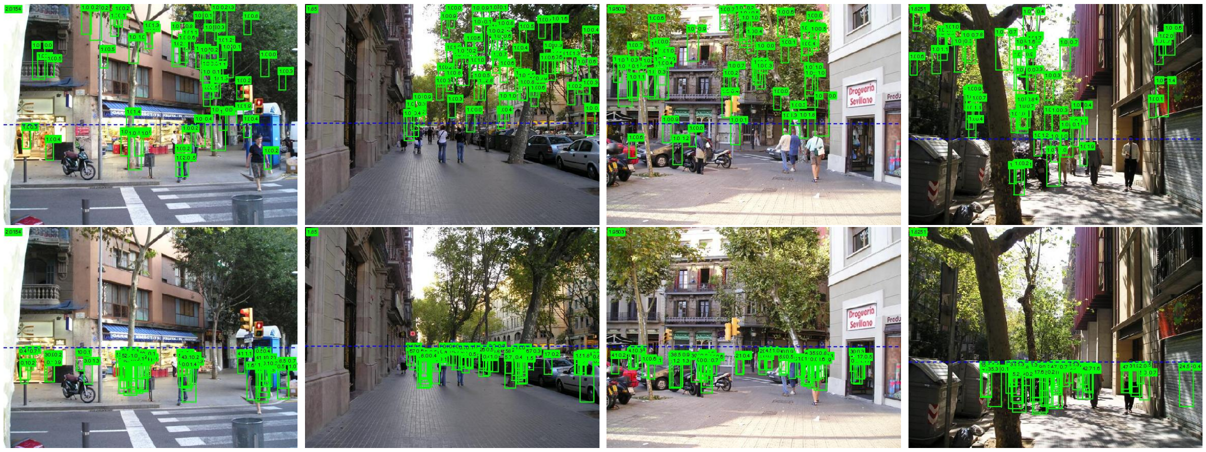


Figure 2.10: Top 50 hard negatives mined by the conventional Dalal-Triggs detector (top) and the context-aware detector (bottom). Notice that many of the false positives occur in non context-plausible regions for former while the latter has more meaningful ones. The blue dotted line indicates the horizon in the image and number at the top left indicates the camera height.

Let a false positive, i.e., detection found by the detector in a negative image (image that does not contain any instance of the object), have bounding box coordinates $bbox_i = \{u_i, v_i, w_i, h_i\}$ (lower-left coordinate, width, and height, respectively) with a score s_i (the SVM classification score is converted to a probability using the method of [118]). As an object's height depends on its position when given the viewpoint, we can compute a context-plausibility score for the given bounding box as follows: if y_i is Gaussian distributed, with parameters $\{\mu_i, \sigma_i\}$, then h_i is also

Gaussian distributed, with parameters $\frac{\mu_i(v_o - v_i)}{y_c}$ and $\frac{\sigma_i(v_o - v_i)}{y_c}$ [75]. This gives a context score c_i for the given bounding box. We define the updated bounding box score as a product of the context score c_i and the original score s_i. We then prune away the boxes with very low confidences (below a certain threshold) to get rid of the detections/false positives in non context-plausible image regions. The resultant set of bounding boxes can be used for subsequent stages of processing, i.e., further iterations in the case of training or final evaluation in case of testing. In Figure. 2.10, the false positives retained by this method are displayed. Notice that the hard negatives mined in this case are those patches that the detector could potentially get confused with at test time.

Experimental Analysis. We conduct our analysis on a subset of LabelMe images [130] that was used in the work of [85]. We chose this dataset because the 3D scene geometry for the images have already been estimated (using the object annotations within the image).

Method	A.P.
Base Train + Base Test	0.25
Base Train + Context Test	0.40
Context Train + Base Test	0.22
Context Train + Context Test	0.41

Table 2.4: Pedestrian detection results on the LabelMe dataset: using context at train, as well as test time produces improved results than only using context at test time.

In Table 2.4, the results obtained using our context-aware detector (‘Context Train’) on the test set are displayed. We compare our results to the detector trained using all possible false positives in the conventional setting (‘Base Train’). Using context at train time, as well as test time produces improved results over only using context at test time. Apart from the performance gain, another important benefit of using context at train time is that it makes the training process memory and computation friendly. The ‘Context Train’ scenario has an order of magnitude fewer negatives (15,000 context-plausible patches) compared to the ‘Base Train’ scenario (150,000 all possible negative image patches). We found the ‘Context Train’ to take about 63 secs for classifier training compared to the 264 secs taken by the ‘Base Train’ scenario (on a 1.1Ghz 8 core AMD Opteron 2354 with CentOS 5.5 running Matlab R2010a). Figure. 2.11 displays a few qualitative results. We observe that the ‘Context Train’ detector avoids many false positives and has fewer missed detections than that occur in case of ‘Base Train’ scenario.

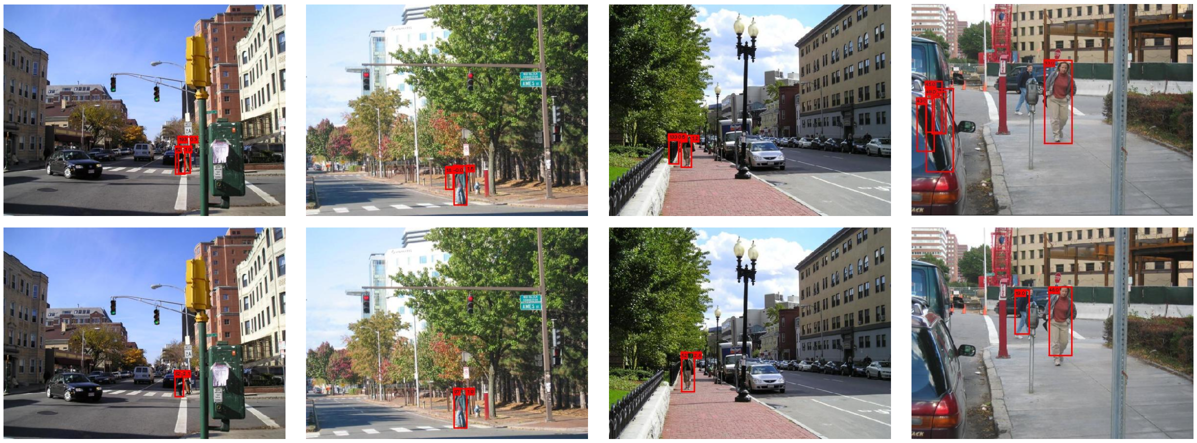


Figure 2.11: Qualitative results – top row:'Base Train' + 'Context Test' detector, bottom row:'Context Train' + 'Context Test' detector. 'Context Train' detector has fewer false positives and fewer missed detections than that of 'Base Train' detector.

Future Extensions. Several issues remain to be explored in this direction. We list a couple of interesting future works: (i) In our experiments, we have assumed the ground-truth context estimates to be available at training time for gathering the context-plausible negative patches. Although this assumption is not very strong [35], it may not always hold. In general, contextual reasoning is often more accurate if the interactions between the scene elements are modeled together, rather than in isolation, as highlighted in [75]. Nonetheless, such a joint inference scheme can be difficult to perform at train time as the object candidates involved are false positives, which could lead to incorrect estimates. To circumvent this issue, an iterative framework can be adopted where one first starts with a detector trained in the conventional setting to obtain good 3D scene geometry estimates [75]. These scene estimates can be used to train a context-aware detector for another object in the scene. This new detector can be used to compute improved scene geometry estimates, which in turn can be used to update the original detector. (ii) In our experiments, we make a hard decision of rejecting false positives below a threshold. Instead, one could consider re-ranking the false positives based on their context-plausibility score.

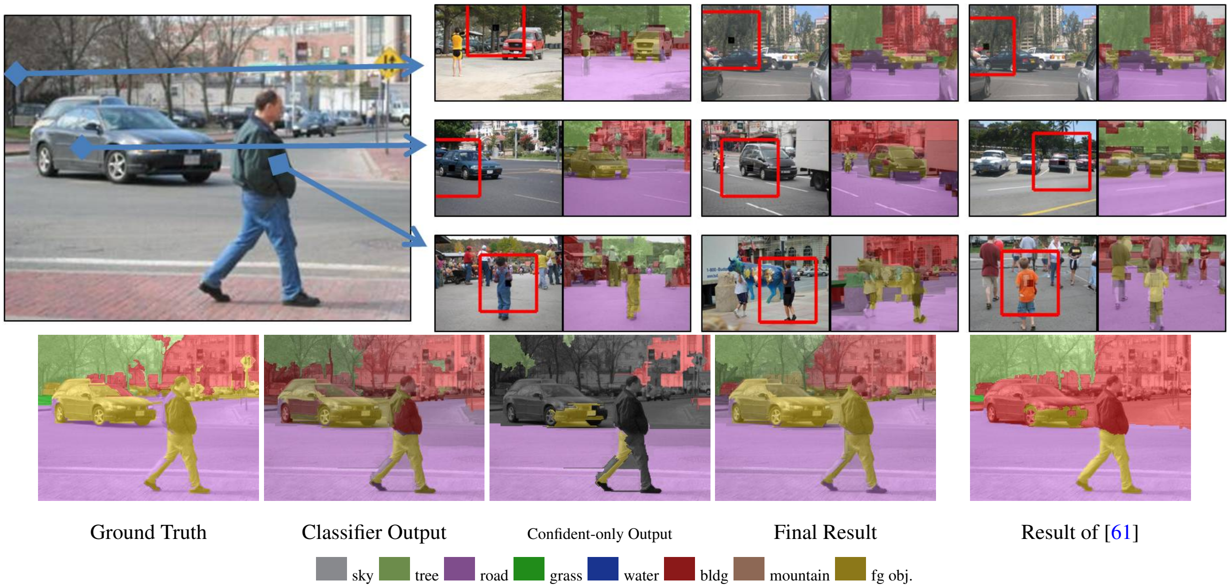
2.5 Conclusion

In this chapter, we have presented an empirical analysis of the role of context for the task of object detection. By achieving substantial gains on the challenging PASCAL VOC dataset, we have shown that contextual reasoning is a critical piece of the object recognition puzzle. Context not only reduces the overall detection errors, but, more importantly, the remaining errors made by the detector are more reasonable. Many sources of context provide a large benefit for recognizing a small subset of objects, yielding a modest average improvement. This highlights the importance of evaluation on many object types as well as the need to include many types of contexts if good performance is desired for a wide range of objects. We also described an approach for tying the detectors and context “in a loop” so that detection can benefit from contextual information even while training.

In this work, we have considered general sources of context that mostly relate to how pictures are taken by humans in an unrestricted setting e.g., images in PASCAL VOC dataset [39]. If additional domain knowledge is available about how the data has been collected, e.g., pictures taken by a mobile robot in an indoor environment or an autonomous car driving on streets, then it is possible to incorporate this knowledge as additional contextual constraints [33, 36].

Chapter 3

Beyond Object Detection: Role of Context in Image Parsing


Ground Truth

Classifier Output

Confident-only Output

Final Result

Result of [61]

Figure 3.1: Given a query image, we retrieve matches for each individual (local) patch by searching a database of 6 million images. The matches are used to compute contextual priors that are used in updating a supervised classifier (trained from a small set of labeled images) to improve its performance.

While we have seen that context can certainly aid object detection performance, are there other visual tasks that can benefit from the use of context? If so, what type of context can

[1] Parts of this work have been described in Divvala et al. [30, 34].

they leverage upon? In this chapter, we will study the application of context to another very important vision problem. We will show that it is possible to estimate useful contextual cues from completely unlabeled web images to aid the task of image parsing.

3.1 What context can unlabeled data offer?

Image parsing deals with the problem of describing an image of a 3D scene in terms of its constituent regions. This problem is often posed as a multi-class classification task, i.e., associating a label to every pixel within an image. Many approaches have been proposed recently to address this problem, e.g., [61, 74, 93, 140, 171]. Almost all of the approaches use a small set of labeled images for modeling the region classes. However, for most real-world problems (with many classes and high variability within the classes) there is simply not enough labeled data available to learn rich discriminative models for classification. Although with the rise of Amazon Mechanical Turk and other online collaborative annotation efforts, the process of gathering more labeled data has been greatly eased for several tasks, pixel-wise image labeling remains difficult as it is much more involved. Thanks to the popularity of social-networking and photo-sharing websites (Facebook, Flickr), today there exist several billion unlabeled images available on the web. Devising algorithms that can automatically benefit from this wealth of information can help alleviate the limited labeled data problem.

The effective use of unlabeled images in learning better models for classification is not straightforward. A lot of research exists in the area of semi-supervised machine learning to specifically deal with this problem, e.g., transductive support vector machines, graph-based semi-supervised learning, and co-training (see [172] for a literature survey). The key idea of these methods is to exploit labeled samples as well as a large number of unlabeled samples for obtaining an accurate decision boundary. This intuition is summarized by the so-called "cluster assumption," i.e., assuming different classes come in clearly separated clusters, unlabeled data can help to delineate the boundaries of the clusters [142]. For the problem of image parsing though, the "cluster assumption" fails to hold due to high appearance ambiguity, i.e., regions that look very similar in terms of appearance can have different labels. This ambiguity has prevented the conventional semi-supervised methods from succeeding at this task [105]. One possible solution to overcome the ambiguity in features is to increase the size of the regions, i.e., the elementary patch units such as superpixels [124] used in the clustering process. However, this is not plausible due to the scarcity of labeled data. That is, we would end up with well separated clusters but with no labeled data samples to label them.

To address this conundrum, we propose to decouple the size of a region (also referred as patch) and the type of the training data (i.e., labeled vs. unlabeled) involved. As there is only limited amount of labeled data, we use a small patch size, which provides us with enough data to learn a rich local classifier. While for the unlabeled data, which is available in large quantities, we use a larger patch which allows the "cluster assumption" to hold and encodes longer range connections not accessible to the local classifier. The idea of using a larger neighborhood to guide local classifiers has been studied in many recent works [61, 71, 74, 93, 96, 104, 122, 129, 151, 171]. The basic insight shared amongst them is to use information from the neighborhood around a local patch to derive a prior probability over different classes. However, almost all of the approaches have considered deriving such priors in completely supervised settings, i.e., using label information from annotated images, which as highlighted earlier comes in scarce quantity and thus prevents from learning anything at a large scale.

In this work, we present an approach for deriving unsupervised patch-based context from a large collection (millions) of unlabeled images for the tasks of region classification [61] and surface-layout estimation [74]. In Section 3.2, we explain our notion of unsupervised patch-based context and describe our approach for extracting the contextual prior. Section 3.3 presents our results and demonstrates that useful contextual priors can be extracted from unlabeled images.

3.2 Unsupervised Patch-based Context

Consider the image in Figure 3.1. A local region-based classifier does a good job at parsing this scene (see *Classifier Output*). However, since it makes local decisions without reasoning about the high-level context of the scene, it makes mistakes in some of the regions (particularly those with unusual or confusing local patch-based features, e.g., person shirt, car door, etc.). Now let us assume that we have access to the set of nearest neighbors to the query patch that all have the same underlying semantic (label) configuration as the query, but have different local patch-based features. Amongst the retrieved matches, let us assume that the region-based classifier produces better results on some of them (compared to the result on the query). By marginalizing the outputs of the classifier on the retrieved matches, we can compute a useful prior probability over the region-classes for the local patch (referred to as *contextual prior*).

Two challenges arise:

- How can we retrieve the set of good nearest neighbor matches to a query patch (given the high *intra*-class variance and low *inter*-class variance in local region features)?

- How can we ensure that the retrieved matches all share similar semantic characteristics as the query (but are not corrupted by the same mistakes made by the classifier on the query patch)?

To address the first challenge, we consider using features not only from the query patch but also those extracted in a neighborhood around it while performing the matching step. The features arising in the neighborhood provide the much-needed *context* that helps in constraining the search and resolving ambiguities. The size of the neighborhood plays a crucial role. Many recent works have considered image-based context, i.e., using the entire image as the neighborhood [34, 69, 96, 129]. Although global matching works well for some scenes (e.g., alley and shores) where there is enough data, it is not possible to find good matches for all other types (e.g., city and outdoor neighborhood) due to insufficient data for such scene types. To circumvent this issue, in this work, we consider using sub-image neighborhoods for matching [38, 128, 149].

In order to retrieve matches that share the same underlying semantics as the query, we use the outputs of a supervised classifier (trained on a small set of labeled images) as semantic features for performing the matching step. As the classifier is trained to perform a specific task, using its outputs on the local patch as well as in its neighborhoods helps in further constraining the search to the underlying task being solved. One potential problem with this method is that the supervised classifier would make similar mistakes on similar image regions, and thereby relying on those outputs as our features would retrieve matches that are corrupted by the same errors. This would result in computing non-informative priors as marginalizing over the corrupted predictions would reinforce the mistakes and thus lead to no new information. To circumvent this problem, we rely only on the 'confident' outputs/predictions of the classifier while performing the matching step. In most general scenarios, the *easy* regions within an image are often confidently labeled by a supervised classifier. For a classifier exhibiting a low recall-high precision characteristic, its highly confident predictions are mostly correct and thus can be treated as a weak form of ground-truth labels. Therefore, by avoiding the non-confident regions and relying only on the confident predictions to guide the search process, we avoid retrieving matches that share similar mistake patterns.

Our overall approach is as follows: Given a set of (few) labeled and (many) unlabeled images, we first train a supervised classifier (Section 3.2.1) using a subset of labeled images as training data. We then run this classifier over the entire set of images (both labeled and unlabeled) to compute the *semantic* features over them. The semantic features are used (along with appearance features) to search the unlabeled images for retrieving nearest neighbor matches to every image patch in the labeled dataset (Section 3.2.2). The retrieved matches are then used to compute the contextual prior, which is subsequently used to update the supervised classifier to improve its performance (Section 3.2.3).

3.2.1 Supervised Classifier

We use a multiple segmentation approach [74] to train our supervised classifier. Given an image and its corresponding superpixel map, simple features based on color, texture and location are extracted from the superpixel regions and are used to train a superpixel similarity classifier. This classifier is used to group similar superpixels together to form larger segments. The larger segments offer better spatial support for extracting more complex region-based cues such as vanishing lines (geometry), shape, and boundary characteristics. These high-level features (in combination with the low-level cues) are used to train a region classifier so as to learn the mapping between the regions and their corresponding classes. Multiple segmentations are generated and the predictions are marginalized across the segments to assign final label confidences to each superpixel (i.e., the probability p of a super-pixel belonging to one of the classes).

The multiple segmentation approach is simple and fast yet powerful and has shown good performance at various tasks [74, 99, 131]. In our experiments, we found that it achieves a good level of performance given a limited amount of training data and is on a par with other state-of-the-art methods evaluated on the two selected datasets (see Section 3.3).

The multiple segmentation process is based on the hypothesis that some of the generated segments (in the soup of segments) would offer good spatial support that is crucial for classification. Thus, the process encourages homogeneous segments, i.e., *local* regions belonging to a single class, and does not encode higher-order contextual interactions/relations across classes (in fact, such non-homogeneous segments are discouraged in this framework). This is exactly the knowledge we want to augment the supervised (local) classifier with using the contextual prior.

3.2.2 Sub-Image Contextual Matching

Given an image (either labeled or unlabeled), we first divide it into non-overlapping 20×20 pixel patches yielding an $m \times n$ resolution grid (typically 15×20 for a 300×400 pixel image). We extract two types of cues for the image at this resolution. For appearance, we use the gist feature descriptor [110], which has been shown to perform well at grouping similar scenes [69].

We create this descriptor for each image at the $m \times n$ spatial resolution where each cell contains that image patch's average response to steerable filters at 8 orientations and 4 scales. For semantic context, we run the supervised classifier on the original image and discretize its output confidences to the resolution of the grid (hereby referred to as the *semantic* feature). We mask out the semantic features in the *non-confident* cells and only use the features from the *confident* cells. A feature is *confident* if its max prediction value is above a precomputed threshold for the predicted class. The thresholds are decided based on the classifier's performance on a validation dataset.

The feature descriptor for a given patch in the image is built by concatenating the gist and the semantic features from a $k \times k$ square neighborhood around it. (In Section 3.3, we provide details about the specific neighborhoods chosen). Using this feature descriptor, we compute distances for every patch in the query image to all the patches in the database images. We use the L1 metric for computing the distances separately for the gist and the semantic features. (As indicated above, we ignore the cells that have low confidence in computing the distance for the semantic features. This avoids relying on matching noisy estimates to other noisy estimates.). We scale the distances so that their standard deviations are roughly the same to ensure that their influence in ordering the matches is equal. After sorting the aggregate feature distances, we pick the top K-nearest neighbors amongst them. Rather than searching all the patches within each of the 6 million unlabeled images for every query image patch, to circumvent the huge computational cost, we first retrieve the top 10000 global scene matches to the query using the approach of [69] and perform the costlier sub-image search within that subset. This entire process roughly takes 2 hours per query image (using unoptimized Matlab code) on a contemporary Intel Xeon processor. (The computational cost could be reduced by using several recent systems-level optimizations, e.g., branch-and-bound [87], hashing [77, 144], etc.). Some sample matches retrieved for query patches are shown in Figure 3.1 and 3.2.

3.2.3 Estimating the Contextual Prior

Given the top K-nearest neighbors $N_{1:K}$ retrieved for a query patch q in an image, its contextual prior P_q is computed by marginalizing the outputs of the supervised classifier $p(\cdot)$ on the retrieved unlabeled matches, i.e., $P_q = \sum_{i=1}^{K} p(N_i)$. The matching process implicitly enforced the constraint to retrieve neighbors possessing similar underlying label characteristics as that of the query. Thus by marginalizing their outputs, a good prior over the image classes is derived. This idea of encouraging similar scenes to have similar semantic labels, can be viewed as a weak

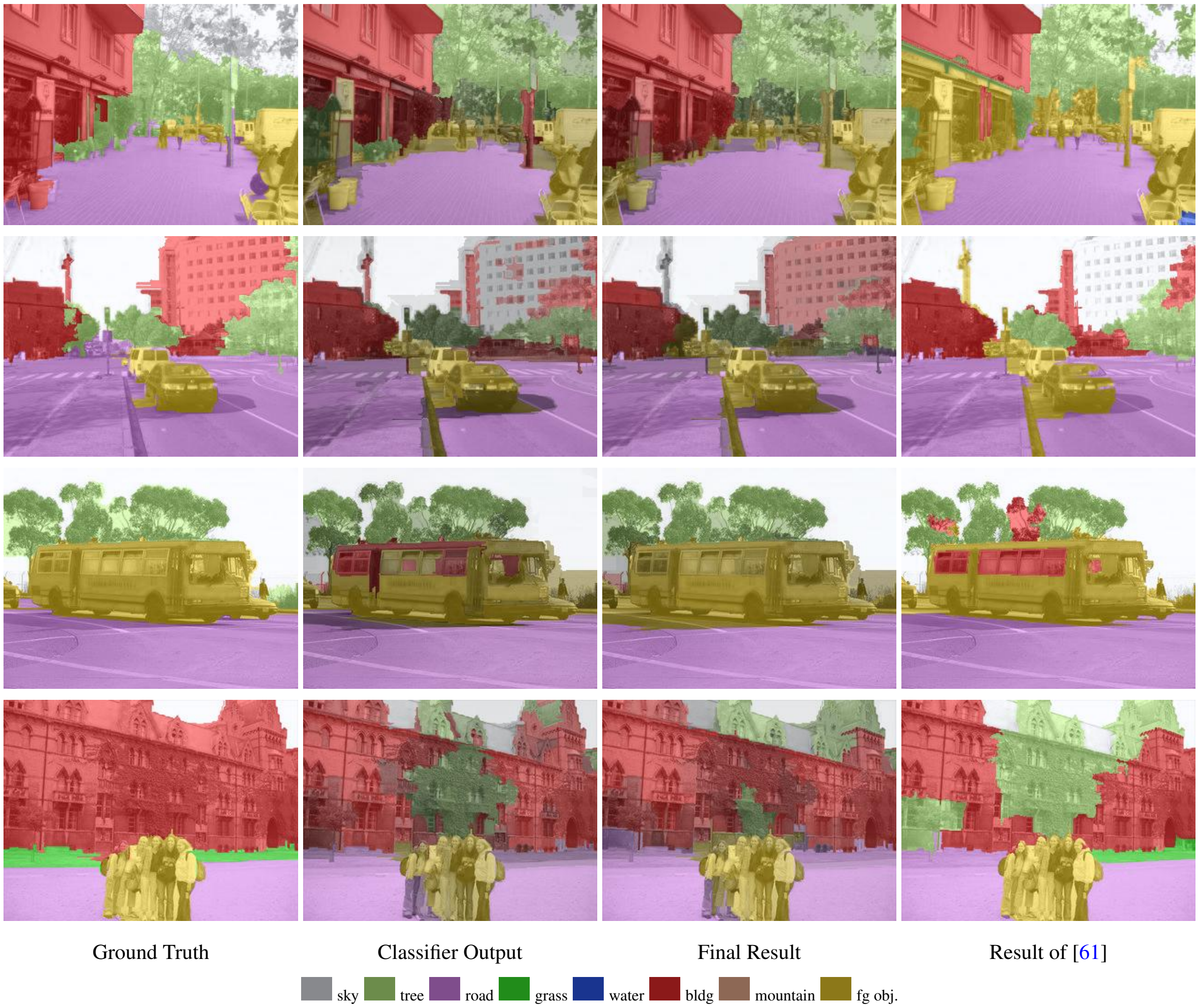


Figure 3.2: Results on the region classification task. First and second row: part of the 'building' is misclassified as 'foreground'. Third row: parts of 'foreground' (bus) are labeled as 'building'. Last row: parts of the 'building' are misclassified as 'tree.' By retraining the classifier using the contextual prior, the mistakes have been rectified (Final Result). Results from [61] are also included for comparison.

form of manifold regularization [12].

In our experiments, we considered two methods for performing the marginalization step: i) direct marginalization of the classifier's outputs across the retrieved matches; ii) marginalizing the outputs over the matches only when they are "confident" (i.e., using N_i only when $p(N_i)$ is above a threshold). We empirically found that using around 15 matches in the first scheme and about 50 matches in the second scheme to perform equally well. We used the first scheme in our experiments.

The estimated contextual prior acts as a useful cue for classification of the input image. In our experiments, we use it as an additional feature alongside the original set of features to retrain the supervised classifier as suggested in [34]. In order to eliminate the variance in the results due to the randomness in the multiple segmentation process, we maintain the same segmentations from the original training process. However, it is possible to use these features in the multiple segmentation process too (i.e., to retrain the superpixel-similarity classifier), which would further help to generate better segmentations.

3.3 Experimental Results

We analyze the performance of our approach on the tasks of region classification [61] and surface-layout estimation [74]. The unlabeled images used in our work are a collection of 6.5 million Flickr images [70].

3.3.1 Semantic Region Classification

The region classification task is to classify the different regions within an image into one of the eight categories: *sky, grass, road, water, mountain, tree, building* and *foreground*. In [61], a unified region-based model that combined appearance and scene geometry to automatically decompose a scene into semantically meaningful regions was used. To compare the results obtained by our supervised learner to the one used in [61], we repeat the experiment following a similar set-up as used in [61], but using the supervised learner as described in Section 3.2.1. Quantitatively, we achieve a pixel-wise accuracy of 76.43%, which is similar to the result reported in [61] (76.4%). This confirms our baseline learner's performance as being on a par with the existing state-of-the-art on this dataset.

For our experiments with unlabeled data, we divide the dataset of 715 images into 4 random splits - 100 images for training the superpixel-similarity classifier, 350 images for training the region classifier, 65 images for validation and 200 images for testing. The validation image set is used to compute the confidence thresholds (The thresholds are set so as to achieve a minimum precision value – 0.15 for mountains and 0.9 for remaining classes). We train the supervised classifier using the 350+100 images and test it on the 200 test image set. The pixel-based accuracy on the test set using this classifier is 75.6%. To obtain the *semantic* features on the training

	sky	tree	road	grass	water	bldg	mntn	fgob
sky	92.6	2.1	0.1	0.0	0.2	4.3	0.0	0.6
tree	4.9	61.4	1.2	0.9	0.1	26.6	0.1	4.9
road	0.2	1.4	88.0	0.5	1.0	4.0	0.0	4.9
grass	1.4	10.4	6.2	74.6	0.8	2.5	0.6	3.5
water	7.7	0.8	25.5	4.6	50.9	5.1	1.9	3.5
bldg	1.9	5.1	2.5	0.6	0.1	83.2	0.0	6.6
mntn	25.6	10.0	9.0	0.8	6.1	42.0	0.8	5.7
fgob	2.2	4.3	16.2	1.5	1.1	24.5	0.0	50.2

Baseline (Supervised) classifier

	sky	tree	road	grass	water	bldg	mntn	fgob
sky	93.2	2.9	0.0	0.0	0.4	2.9	0.0	0.6
tree	4.6	66.5	1.2	2.1	0.0	20.0	0.4	5.1
road	0.1	0.5	89.1	0.4	0.9	2.9	0.0	6.2
grass	0.5	6.0	2.7	84.0	1.8	0.6	0.1	4.3
water	7.1	0.3	24.5	5.6	50.3	2.9	1.3	8.0
bldg	1.7	6.6	2.3	1.2	0.1	81.3	0.1	6.8
mntn	26.2	20.1	8.4	2.0	6.1	12.0	3.0	22.3
fgob	2.8	4.3	14.0	2.0	1.0	17.5	0.1	58.3

After re-training with our Contextual Prior

Table 3.1: Confusion matrix (row-normalized) of the supervised classifier before and after incorporating the contextual prior.

images (required for the matching step), we train/test a separate classifier on the training images using cross-validation.

Given a query image, we repeat the process described in Section 3.2.2 to retrieve the nearest neighbors and compute the contextual prior for retraining the classifiers. The updated classifier improves the result on the test set by 2.4% (from 75.6% to 78.0%). The confusion matrix is shown in Table 3.1. Although the increase in the result seems small (in absolute numbers), it must be emphasized that the improvements made are substantial. More specifically, our approach helps in correcting the mistakes made in classifying pedestrians/cars, tree branches or parts of buildings that typically occupy fewer percentage of pixels in the image (compared to sky and ground) but are crucial for successful image parsing. Qualitative results for some of the test images are displayed in Figure 3.2 and 3.3. Observe that in the regions where the supervised classifier is unconfident (or incorrect) in its result, the contextual prior from unlabeled images helps predict the correct result.

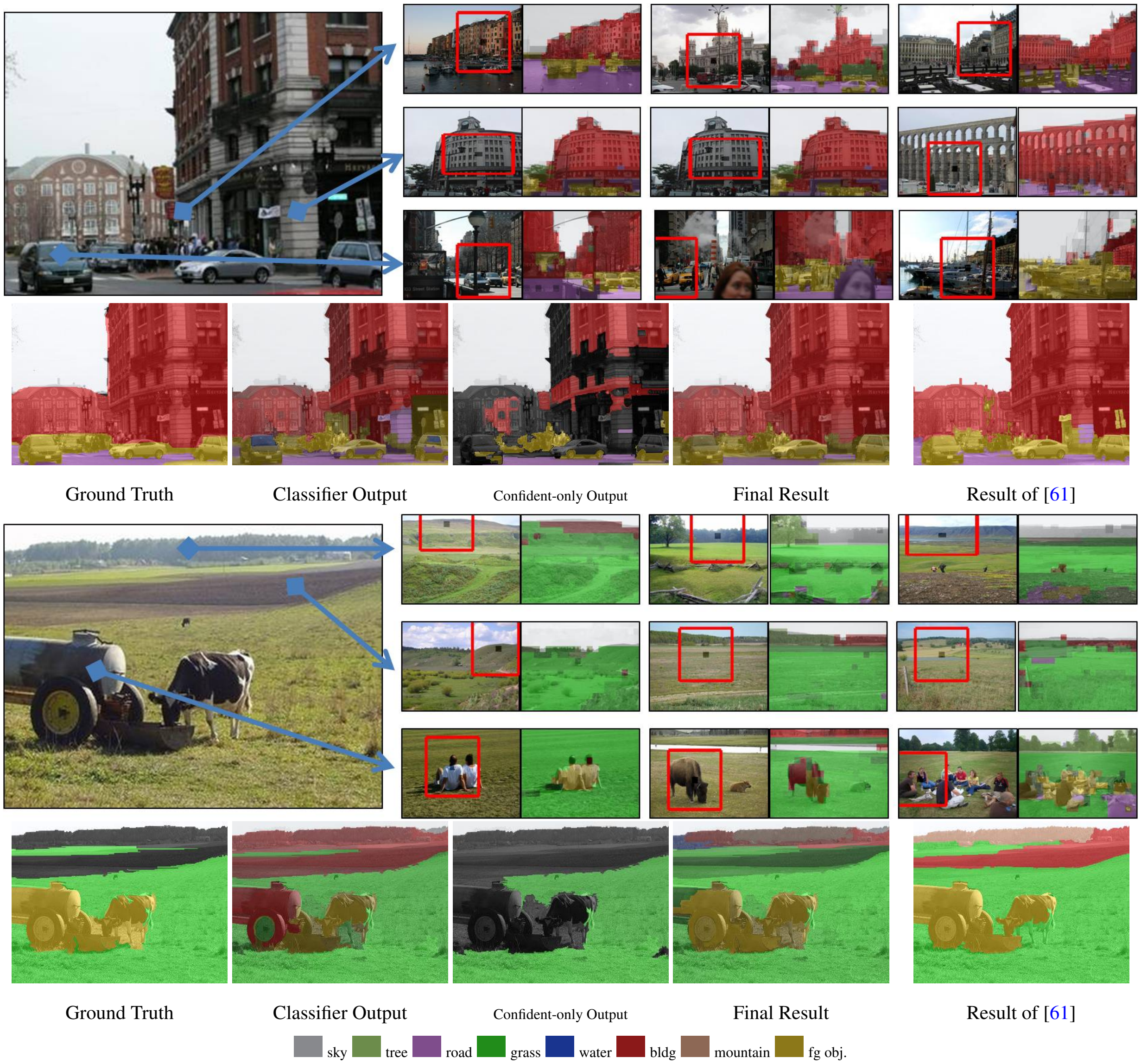


Figure 3.3: Results on the region classification task. Top row: parts of the car and the building are misclassified. However, the confidence of the correct prediction is increased by using the contextual prior from the retrieved matches. Bottom row: parts of the vehicle and grass are incorrectly classified, but corrected by the retrained classifier.

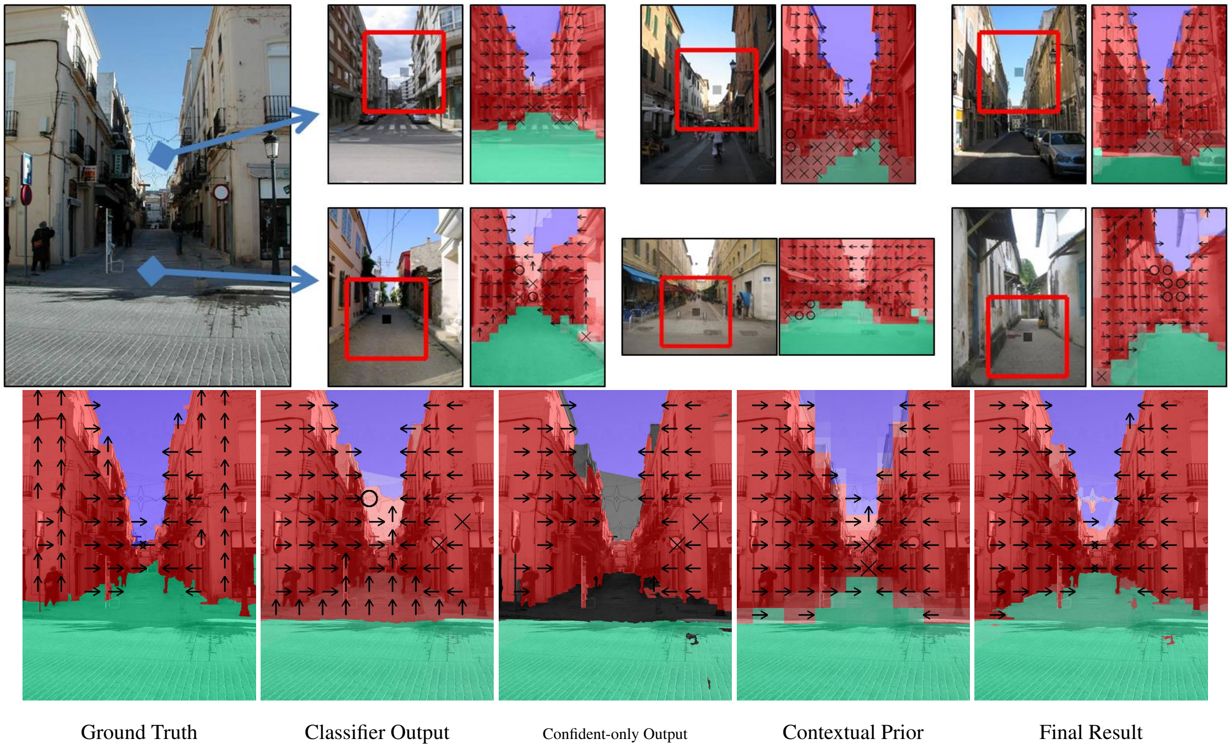


Figure 3.4: Result on Geometric context dataset: Notice that the incorrectly classified ground and sky regions are corrected after the incorporation of the prior.

3.3.2 Surface Layout Estimation

The goal of this task is to segment an image into meaningful geometric surfaces: *ground, planar-left, planar-frontal, planar-right, non-planar porous, non-planar solid,* and *sky*. In [74], an approach based on multiple segmentations was used on a dataset of 300 images. 50 images were used for training the superpixel-similarity classifier and the remaining 250 images were used for training/testing the region classifier in a 5-fold cross validation setup. We use the same splits and setup in our experiments. The accuracy obtained in our experiments was 87.1% for the main class and 59.3% for the sub-class. This is slightly different from (lower than) the accuracy reported in [74]. We attribute this difference to the randomness in the segment generation process (which affects the results by +/- 1% as reported in [74]). However, we use the same set of segmentations in the retraining process (with the contextual prior), to eliminate the variation in our subsequent results (due to the randomness).

Due to the lack of labeled data, we could not maintain separate train-val-test splits in this experiment. We used a classifier trained on the entire set of 250 images to run on all the unlabeled

	grnd	vert	sky
grnd	83.0	16.6	0.3
vert	9.3	88.6	2.1
sky	0.1	9.7	90.2

Main-class

	left	front	right	porous	solid
left	36.8	36.5	7.3	9.9	9.6
front	7.0	53.8	11.9	17.5	9.8
right	4.0	23.6	49.9	11.3	11.1
porous	2.3	8.5	3.0	80.1	6.1
solid	4.5	18.7	7.1	18.6	51.1

Sub-class

Baseline (Supervised) classifier

	grnd	vert	sky
grnd	83.5	16.0	0.5
vert	9.0	89.4	1.6
sky	0.6	6.4	93.0

Main-class

	left	front	right	porous	solid
left	41.7	25.8	7.8	10.7	14.0
front	5.4	56.8	10.6	14.3	12.9
right	2.8	25.8	47.5	11.9	12.1
porous	1.3	6.6	2.1	83.2	6.8
solid	4.4	17.7	5.1	18.5	54.4

Sub-class

After re-training with our Contextual Prior

Table 3.2: Confusion matrix (row-normalized) of the supervised classifier before and after incorporating the contextual prior.

images. The thresholds were set based on the results obtained in the 5-fold cross validation process (to have a minimum precision of 0.9 for sky and ground, and 0.7 for the rest of the classes). Given a query image, we repeat the process described in Section 3.2.2 to retrieve the nearest neighbors and to compute the contextual prior for retraining the classifiers. Quantitatively, for this task, we improve the results by 1.2% on the main-class (i.e., from 87.2% to 88.4%) and by 2.6% on sub-class (i.e., 59.3% to 61.9%). The confusion matrix is shown in Table 3.2. The qualitative results are shown in Figure 3.4,3.5.

What is the right neighborhood for matching? The size of the neighborhood used for extracting features around a patch in the sub-image matching approach plays a non-trivial role (Figure 3.6). Having too small a neighborhood would lead to potentially many matches (but with the good ones lost in the pool of retrieved matches), whereas using a global neighborhood (i.e., the entire image) would lead to too few (potentially zero) matches. Indeed, choosing the right size is data and task dependent. We experimented with various sizes of the neighborhood - 5×5, 9×9 and 13×13 for gist and 9×9 and 13×13 for the semantic features. We found the matches retrieved using a 9×9 neighborhood for gist and 13×13 neighborhood for semantic features

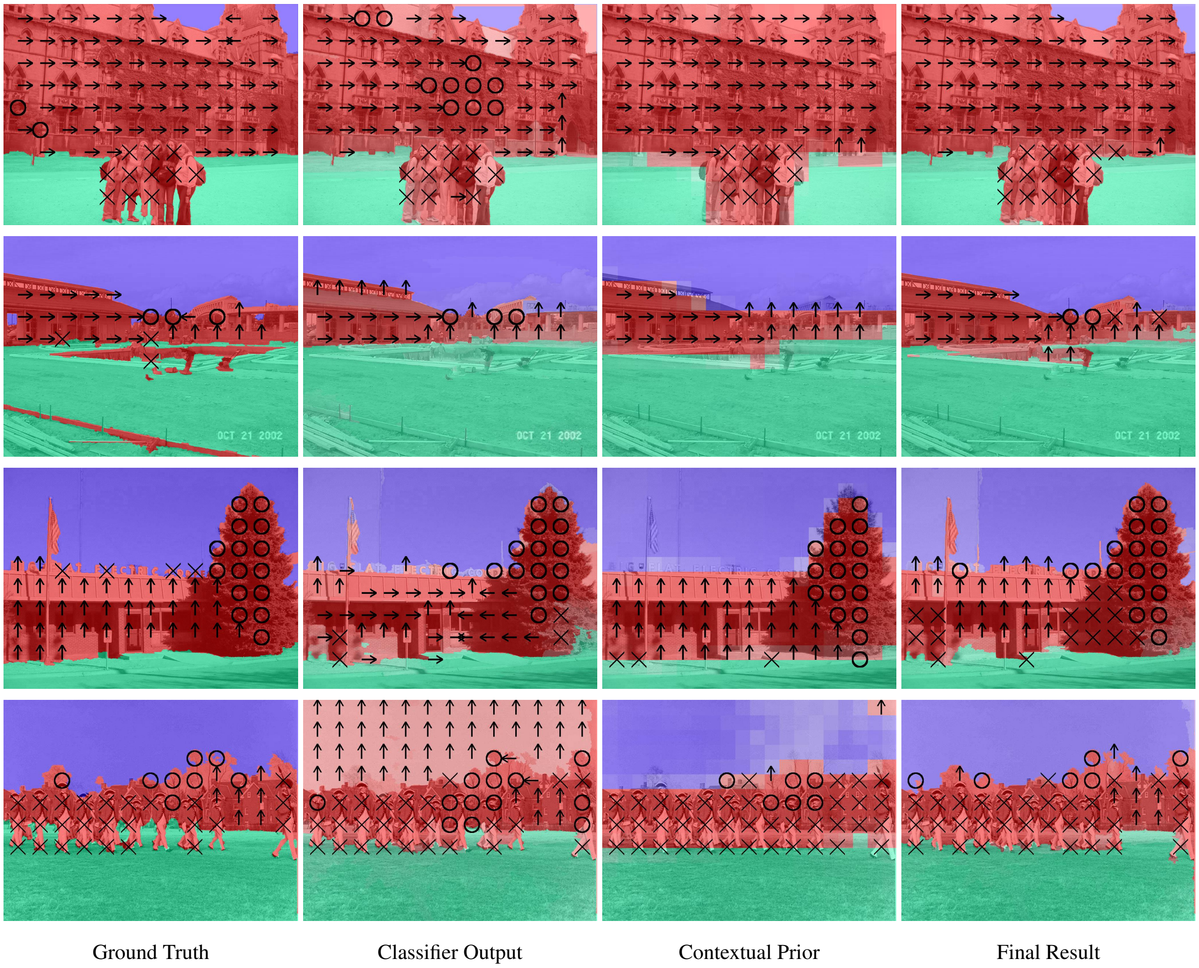


Figure 3.5: Results on the geometric context dataset. ('X' indicates the non-planar solid and 'O' indicates the non-planar porous class). Top row: part of the left-facing building is misclassified as porous due to confusing texture. Second row: left-facing roof of the building is misclassified as 'frontal'. Third row: frontal face of the building is confused with 'left' and 'right' classes. Last row: sky is misclassified as vertical (frontal) class. In all cases, the contextual prior computed from unlabeled images helps to improve the result.

to be good (based on the performance on a validation set). We used these settings in all our experiments.

Standard Semi-Supervised Learning comparison. We compared the performance of our approach to a standard semi-supervised learning (SSL) algorithm that directly takes labeled and unlabeled data together, with no intermediate labeling of the unlabeled data while using the same patch sizes for labeled and unlabeled data types. More specifically, we experimented with

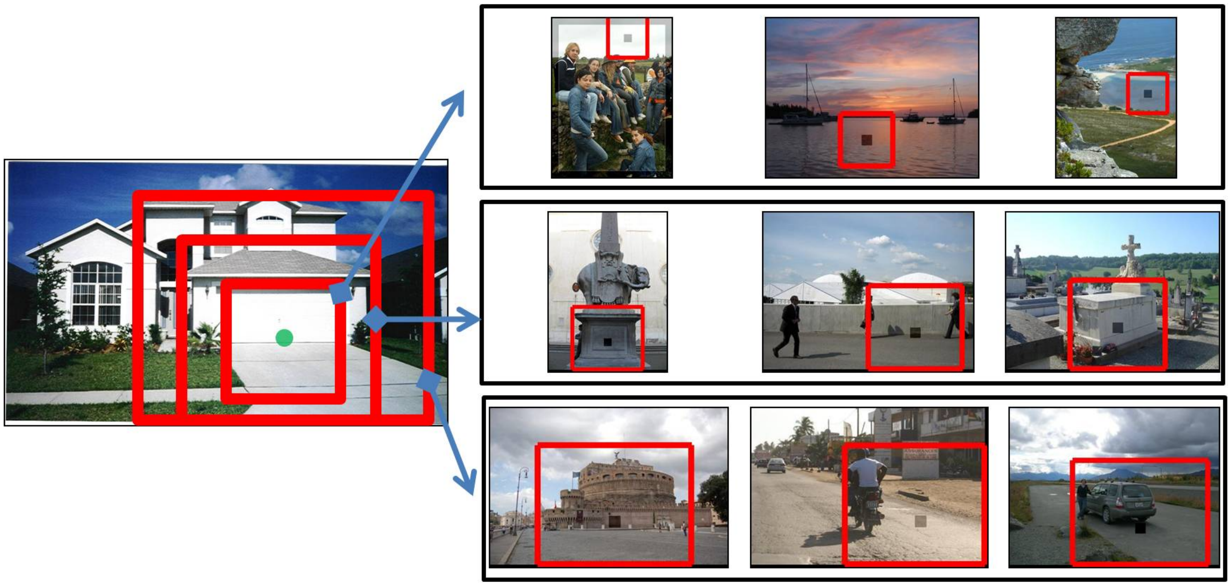


Figure 3.6: Role of the context neighborhood size: Top few matches retrieved for a selected query image (with 15×20 resolution grid) patch using the gist feature at $5 \times 5, 9 \times 9$ and 13×13 cell neighborhoods from a set of 60000 unlabeled images. Having too small a neighborhood around the selected patch (green circle) leads to the top few matches being poor, i.e., random matches on sky and sea regions as the 5×5 cell sub-image does not have enough spatial context (thus the ‘good’ matches are lost in the potentially infinite matches). A larger neighborhood helps to retrieve better matches.

the multi-view SSL algorithm described in [15]. We trained classifiers using the available labeled data for various splits of our feature set and then applied them to all the unlabeled images for bootstrapping the initial classifiers with informative patches mined from them (i.e., patches that are classified with high confidence by at least one view but not all). This method failed to achieve any performance gains in our experiments. Due to the high appearance ambiguity of local patches across multiple feature views (e.g., a local patch of blue in a scene close to the horizon could either be ‘sky’ or ‘water’ unless a neighborhood around it is revealed), this method failed to gather informative samples. As a result, no new information is leveraged from the unlabeled images leading to no improvements in accuracy.

Image matching comparison. We compared our proposed sub-image matching approach (Section 3.2.2) to two other matching approaches: (i) In order to support our hypothesis that sub-image matching helps retrieve improved matches over global methods, we repeated our experiments by using the prior computed from global matches. More specifically, for each query image,

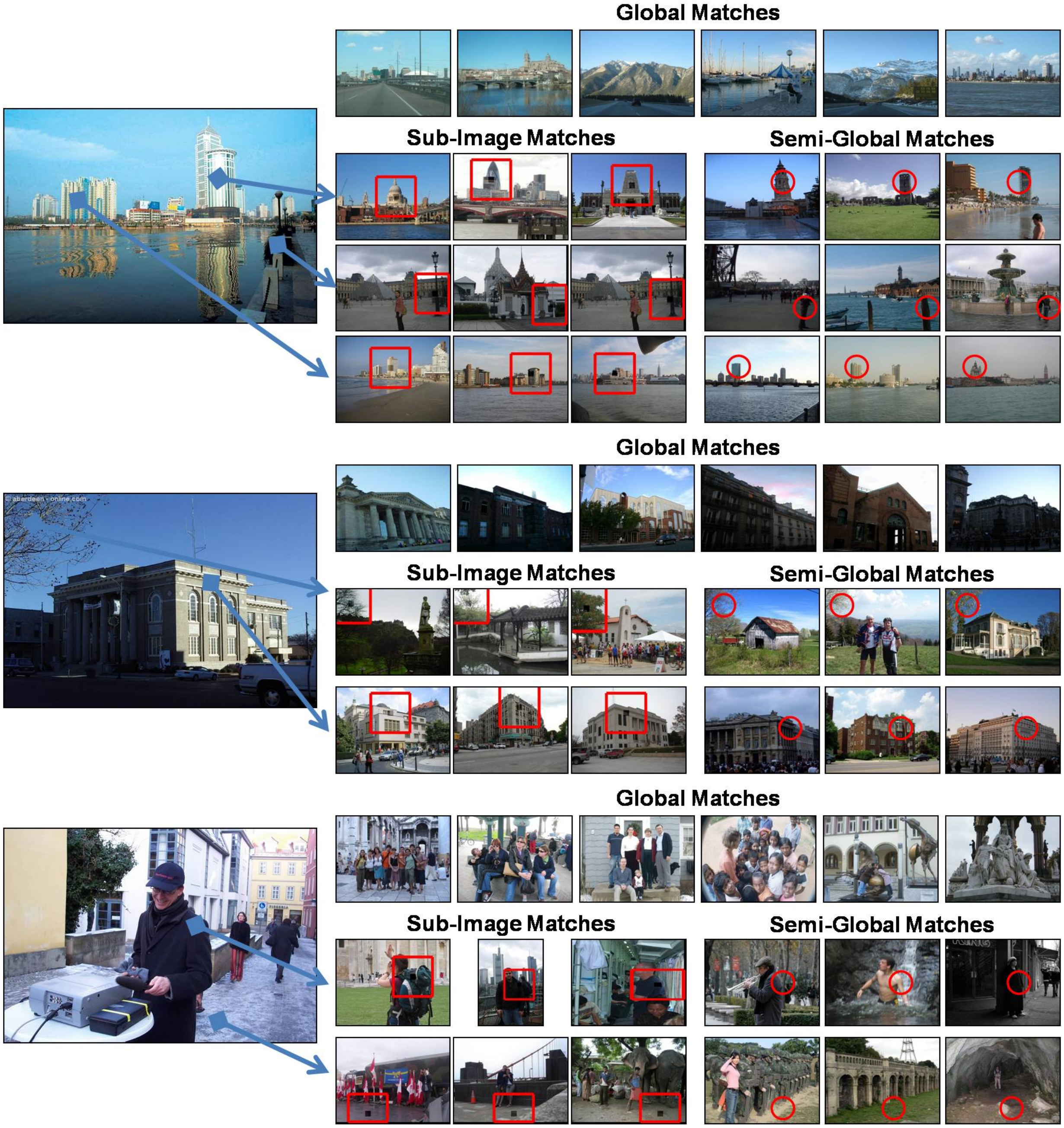


Figure 3.7: Global vs. Sub-image matching: The matches retrieved using features from the entire image do well in getting the overall gist of the scene but fail to match the individual regions within the image. By using the sub-image based approach, we retrieve better matches. The semi-global approach is also displayed for comparison.

we retrieve the top 50 global scene matches and compute the contextual prior by marginalizing the classifier outputs over the matches (on the entire image). For the region classification and the main-class surface layout estimation task, using this prior did not help in improving the result (the change in accuracies was less than 0.2%), while for the sub-class surface layout estimation task, the results improved by 2% (i.e., from 59.3% to 61.3%). Figure 3.7 compares the matches retrieved using both the global and sub-image matching schemes for a few query images. The global matches get the gist of the scene right but do not localize the regions and the boundaries specific to a query patch, whereas using the sub-image approach retrieves much better matches. (ii) We also studied a semi-global scheme to obtain matches. Instead of using a straightforward L1 distance function over the entire image features, we weight the distances using a Gaussian centered around the query patch so as to focus more on the distances in its immediate neighborhood while still matching weakly on the rest of the image. We found this method to qualitatively retrieve better matches (Figure 3.7). However the final classification result, obtained by using the contextual prior estimated from the retrieved matches, was the same as the result obtained with the sub-image matching scheme. As our focus was to improve the classification accuracy (rather than just matching), in our experiments, we used the sub-image matching scheme as it performed equally well and was faster to compute.

3.4 Conclusion

Image parsing is a hard problem as local evidence learned from a small set of labeled images is used for making scene-level decisions. In this chapter, we presented an approach to alleviate the limited labeled data problem by deriving contextual priors from unlabeled images for aiding supervised region labeling classifiers. The main components of our approach are sub-image based matching, and semantic feature based similarity, that together enable us to encode *higher-order* context for measuring similarity and retrieving good matches. Beyond the region labeling tasks explored in this work, the proposed method allows us to leverage the huge collection of untapped images from the Internet in multiple interesting ways. For example, once the nearest neighbor matches to the query patches are retrieved, one could transfer any weak labels associated with the matches (e.g., Flickr tags, captions or any other annotations) onto the query to arrive at a completely data-driven interpretation of the query image.

Part II: Thinking Inside the Window

> "I'll be more enthusiastic about encouraging thinking outside the box when there's evidence of any thinking going on inside it."
>
> Terry Pratchett

Chapter 4

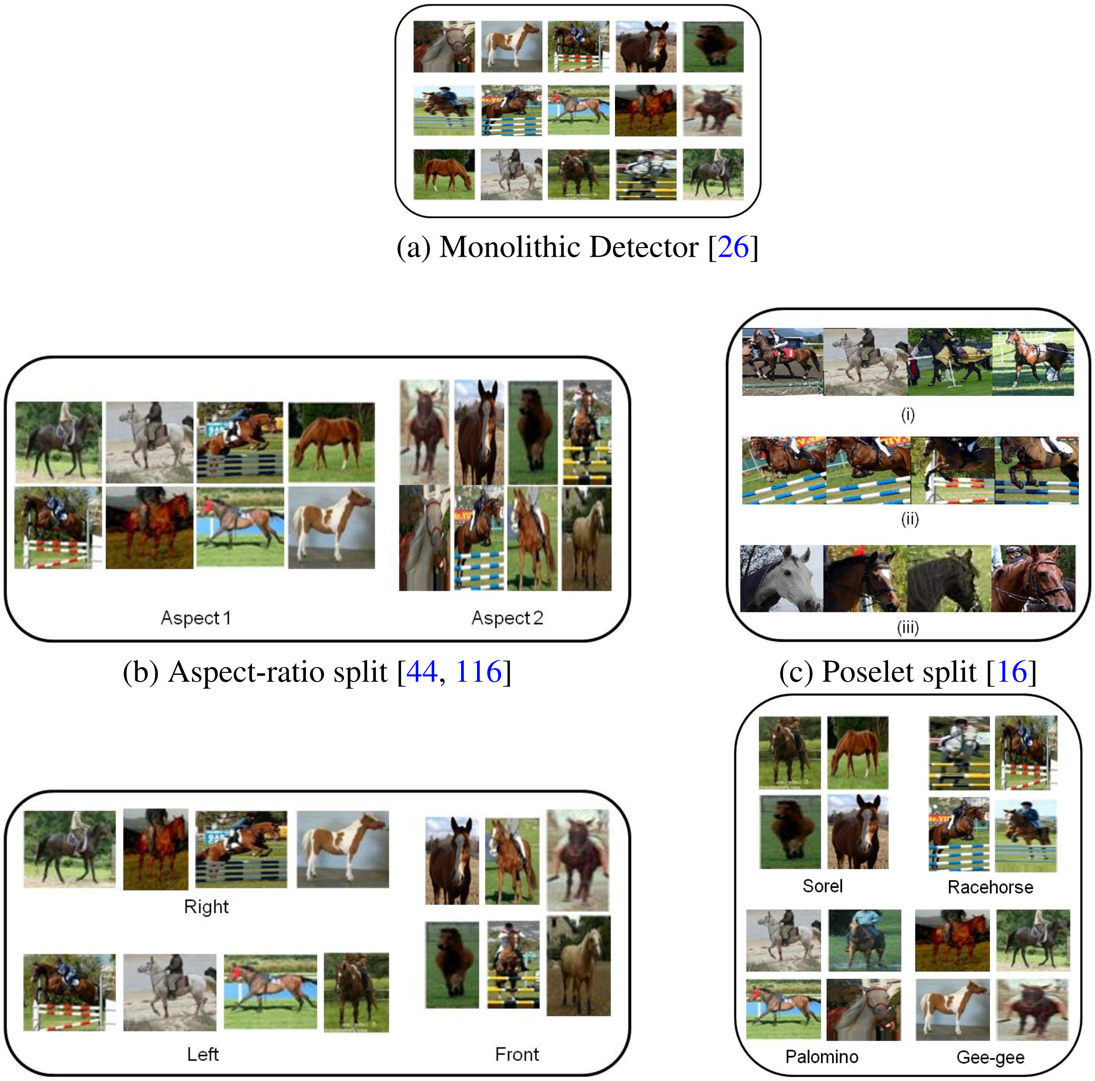
Role of Subcategories

Consider the images of category "horse" in Figure 4.1 from the PASCAL VOC dataset. Notice the huge variation in the appearance, shape, pose and camera viewpoint of the different instances – there are left and right-facing horses, horses jumping over the fence in different directions, horses carrying people in different orientations, and close-up shots, etc. The standard procedure to learn a sliding window classifier treats all instances together as belonging to a single class and trains a binary classifier to identify new "horse" instances in a test dataset. While such an approach [26] was effective for older datasets that had simple appearance variations (e.g., all instances in the INRIA horse and person dataset [2] have similar pose, scale and camera viewpoint), it cannot handle the rich diversity in modern datasets.

To cope with this challenge, recent works have introduced the idea of partitioning the data into smaller clusters, i.e., *subcategories* for training multiple classifiers. The smaller clusters have reduced diversity, thereby leading to simpler learning problems. However, there is an infinite number of ways to split a basic-level category into subcategories. For example, meaningful car subcategories can be based on object pose (e.g., left-facing, right-facing, frontal), car manufacturer (e.g., Subaru, Ford, Toyota), or some functional attribute (e.g., sports car, utility vehicle, limousine). Figure 4.1 illustrates a few popular subcategorization schemes. While acknowledging the performance gains demonstrated by the various schemes, we seek to understand the key insight shared behind their success.

What is it that the different partitioning schemes are trying to achieve? A closer look at the figures reveals that all the methods are trying to encode the homogeneity in appearance. It is the *visual homogeneity* of instances within each subcategory that simplifies the learning problem

[1] Parts of this work have been described in Divvala et al. [32].



(a) Monolithic Detector [26]

(b) Aspect-ratio split [44, 116]

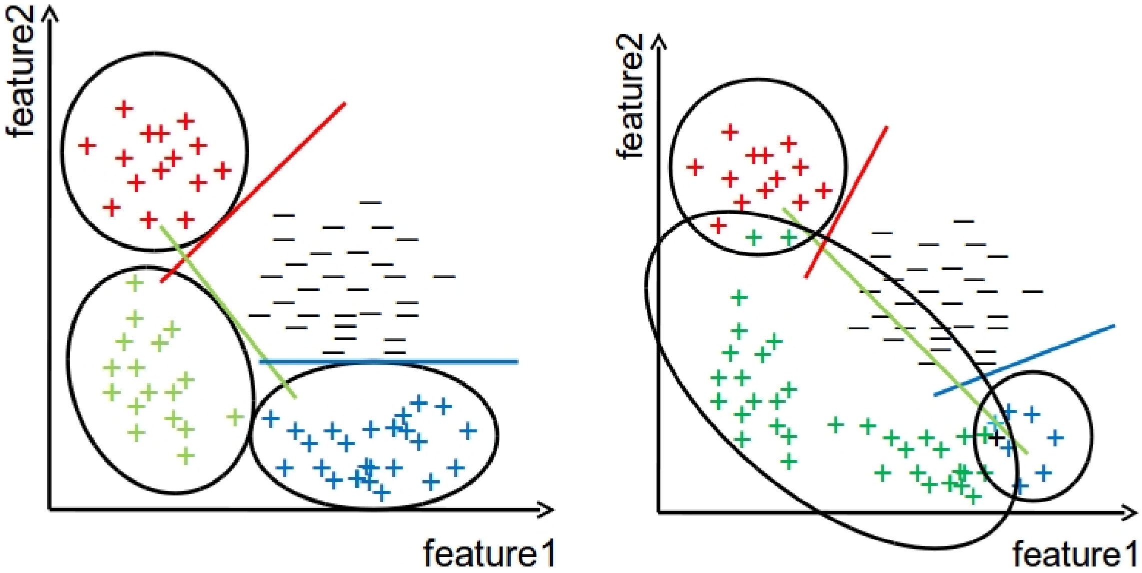
(c) Poselet split [16]

(d) Viewpoint split [23, 63]

(e) Subordinate-category split [27, 29]

Figure 4.1: There is a wide visual variability within a single semantic category, e.g. ‘horse’, that prevents existing methods from learning a good discriminative object model. (a) The standard monolithic approach is trained on all instances together. Recent methods have considered reorganizing data into subcategories based on (b) an aspect-ratio heuristic, (c) keypoint annotations (d) viewpoint annotations (e) subordinate category annotations. This thesis identifies the common insight shared between the different partition schemes (that is *visual homogeneity*) and formalizes it using the concept of visual subcategories.

leading to better-performing classifiers (Figure. 4.2). What this suggests is that, instead of using semantics or heuristics, one could directly use appearance features for building the subcategories.



Visual Subcategories

Semantic Subcategories

Figure 4.2: A single linear model cannot separate the data well into two classes. (Left) When similar instances (nearby in the feature space) are clustered into subcategories, good models can be learned per subcategory, which when combined together separate the two classes well. (Right) In contrast, a semantic clustering scheme also partitions the data but leads to subcategories that are not optimal for learning the category-level classifier.

In this work, we build upon this key insight and present a data-driven approach for discovering *visual subcategories* (Figure 4.3).

The proposed approach is attractive as it neither requires any ground-truth annotations nor involves heuristics in generating the subcategories. This is important as annotations such as keypoint configurations may not be available for many large datasets and even if available, could be sub-optimal (e.g., a "frontal" viewpoint horse-subcategory could still have considerable appearance variation with close-up shots of horses, horses jumping over the fence, etc). On the other hand, heuristics are often brittle and may fail to generalize to a large number of categories (for example, using the aspect ratio for horses does not separate the instances well). Since semantic (human-annotated) subcategorization has the problem of grouping different visual concepts together, this thesis advocates visual subcategorization (Figure. 4.2).

Beyond offering improved performance, visual subcategories are attractive for two reasons. First, as instances within each of our subcategories are better aligned, simple object representations and learning algorithms (such as a linear SVM) suffice to obtain well performing object models. For example, we show that it is possible to obtain a result only slightly inferior to the

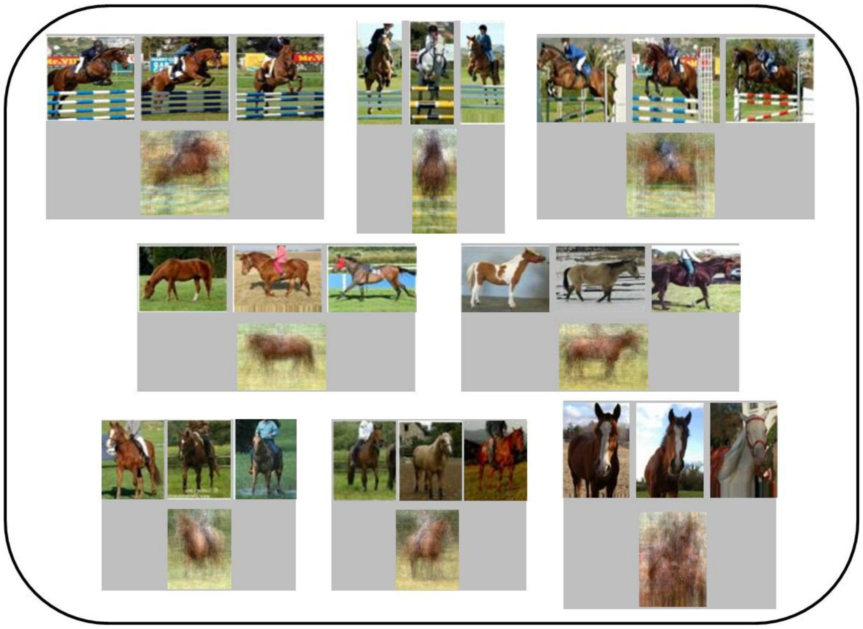


Figure 4.3: Top 8 of 15 components using the proposed unsupervised clustering approach. The models are much more interpretable and lead to higher detection performance. Figure shows a few examples (top), the mean image (left bottom), and the learned SVM weight vector (bottom right).

deformable part model of Felzenszwalb et al. [44] without their *deformable* parts. Furthermore, we show that the uniform griding of the feature space chosen as a convention in typical sliding window detectors may no longer be desirable. By taking advantage of the high intra-subcategory alignment, we present a simple way to prime the feature space to obtain a compact yet discriminative representation. Second, the discovered subcategories are interpretable and in many cases match the fine grained subordinate categories carved using human supervision. This is attractive as it alleviates the need for ground-truth annotations for gathering fine grained subcategories.

4.1 Background

Our approach is inspired in part by work in the machine learning literature [22, 50, 51, 76, 78] that considers solving a complex (nonlinear) classification problem by using locally linear classification techniques. The basic idea is to approximate a nonlinear decision boundary by linear decision surfaces, each of which is determined by a local linear classifier. In mixtures of local experts [76], different classifiers compete to control different regions of the input space, and a gate network is used to weight the output of the classifiers (i.e., models the mixing parameters of a mixture distribution). Instead of searching for regions of expertise as part of the classification process, in [50, 51, 78], the problem is separated into a clustering and a classification step. Each cluster is treated as a subclass that must be learned independently of the rest. This enables fitting models of varying complexity on different regions of the input space. In [22, 49], instead of treating each cluster independently of the others, the information given by the neighborhood clusters is used to enhance the specificity of a given cluster and to characterize its most typical behaviors. All of the above methods have demonstrated competitive generalization accuracy and higher training efficiency than other advanced approaches such as neural networks, generalized linear discriminative analysis, and nonlinear support vector machines.

The idea of explaining a complex class in terms of simpler, smaller classes has also received significant attention in cognitive psychology. In the seminal work of [125], the concept of prototypes was introduced. Prototypes are objects chosen to give a simple and understandable summarization of a class. The concept of prototype relies on the notion of typicality: its resemblance to the other members of the category and its differences to the members of other categories. Quite closely related is the concept of organizing instances of a basic-level category into semantic subordinate categories [9, 80]. Subordinate categories are at the bottom of object taxonomies and display a low degree of class inclusion and a low degree of generality. They provide identifiable and detailed gestalts with relatively detailed configurations of individuating properties.

Computer vision approaches, inspired from these findings, have considered different strategies for generating subcategories. In [23, 68, 136, 138], viewpoint annotations associated with instances were used to segregate them into separate left, right, frontal clusters. In [116], the size (height) of detection windows was used to cluster them into near and far-scale sub-classes. In [44], the aspect ratio of the bounding boxes was used to separate them into different clusters. In [169], co-watch features are used to group videos of a specific category into simpler classes. In [16], instances are clustered into *poselets* using keypoint annotations in the configuration space. In [27], subordinate categories of a basic-level category are constructed using human annotations.

However as argued earlier, semantic (human-annotated) subcategorization may still lead to grouping dissimilar visual concepts together and hence may not be optimal for learning improved classifiers. In passing, it must be noted that our approach uses subordinate categories to increase the performance of basic-level categorization methods. This is different compared to some of the recent works in computer vision that attempt to tackle the problem of fine-grained subordinate categorization [19, 168, 170].

Our work shares some similarities with the recent work of [63], which extends the approach of [44] to object viewpoint classification. By clustering the instances based on HOG features, they train discriminative classifiers, each implicitly corresponding to a different viewpoint of the object. However their focus was limited to viewpoint classification and thus evaluated only on the VOC car dataset. In this thesis, we demonstrate that the idea of unsupervised data-driven clustering is far more general and much more powerful and we use it to address other significant forms of object variation corresponding to appearance, shape and pose. We show our analysis on all 20 classes of VOC dataset as well as the SUN397 scene dataset. Also the issue of model calibration was not addressed in their work as they restricted their analysis to a small number (4 viewpoints) of equal-sized clusters.

Finally, our work is also closely related to the recently popular exemplar-based methods [23, 100]. While in a pure global strategy, a single classifier is trained using all instances belonging to a class as positives, in the case of exemplar-based methods, a separate classifier is learned for each individual instance. Although promising results have been demonstrated, exemplar methods are prone to overfitting as too much emphasis is often placed on local irregularities in the data [160]. The global and local learning strategies sit at two extremes of a large spectrum of possible compromises that exploit information from labeled examples. This work explores intermediate points of this spectrum.

4.2 Learning Unsupervised Subcategories

Our goal is to learn a set of subcategory classifiers to separate the positive instances (e.g., bounding boxes of horses) from the negative instances (background), wherein each individual classifier is trained on different subsets of the training data. Since we want the instances within each subcategory to be tightly aligned in the feature space, we propose to use a data-driven clustering step followed by iterative refinement for training the classifiers. We cast the iterative refinement problem as a latent variable model estimation problem [5] where the cluster assignments of instances are modeled as latent variables. Given the cluster assignments, classifiers are trained for each subcategory and are subsequently used to update the cluster memberships.

4.2.1 Latent SVM with Calibration

Consider a classification problem where we observe a dataset of n labeled examples $D = (< x_1, y_1 >, \ldots, < x_n, y_n >)$, with $y_i \in \{-1, 1\}$. For standard binary classification problem, a commonly used approach is to minimize the trade-off between the l_2 regularization term and the hinge loss on the training data:

$$\arg\min_{w} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\epsilon_i, \tag{4.1}$$

$$y_i \cdot s_i > 1 - \epsilon_i,\ \epsilon_i > 0, \tag{4.2}$$

$$s_i = w \cdot \phi(x_i) + b. \tag{4.3}$$

The parameter C controls the relative weight of the hinge-loss term, $w \in R^D$ is the vector of model parameters and $\phi(.) \in R^D$ denotes the feature representation for sample x_i.

Now consider learning a classification problem where examples are clustered into K separate sub-classes (subcategories), and a separate classifier is trained per subcategory. The assignment of instances to subcategories is modeled as a latent variable z. This binary classification task is formulated as the following (latent SVM) optimization problem that minimizes the trade-off between the l_2 regularization term and the hinge loss on the training data [44]:

$$\arg\min_{w} \frac{1}{2}\sum_{k=1}^{K}||w_k||^2 + C\sum_{i=1}^{n}\epsilon_i, \tag{4.4}$$

$$y_i \cdot s_{i,z_i} \geqslant 1 - \epsilon_i,\ \epsilon_i \geqslant 0, \tag{4.5}$$

$$z_i = \arg\max_{k} s_{i,k}, \tag{4.6}$$

$$s_{i,k} = w_k \cdot \phi_k(x_i) + b_k. \tag{4.7}$$

w_k denotes the separating hyperplane for the kth subclass, and $\phi_k(.)$ indicates the corresponding feature representation. Since the minimization is semi-convex, the model parameters w_k and the latent variable z are learned using an iterative approach [44]. Note that given the latent assignment of examples to the subcategories z_i, the optimization problem in equation (4.4) boils down to solving K separate one vs. all (binary) classification problems (corresponding to each subcategory). The latent relabeling step in equation (4.6) acts as a simple yet important *discriminative* re-clustering step that deals with noisy assignments and fragmentation of the data (which occurs because the "optimal" number of clusters is not known in advance).

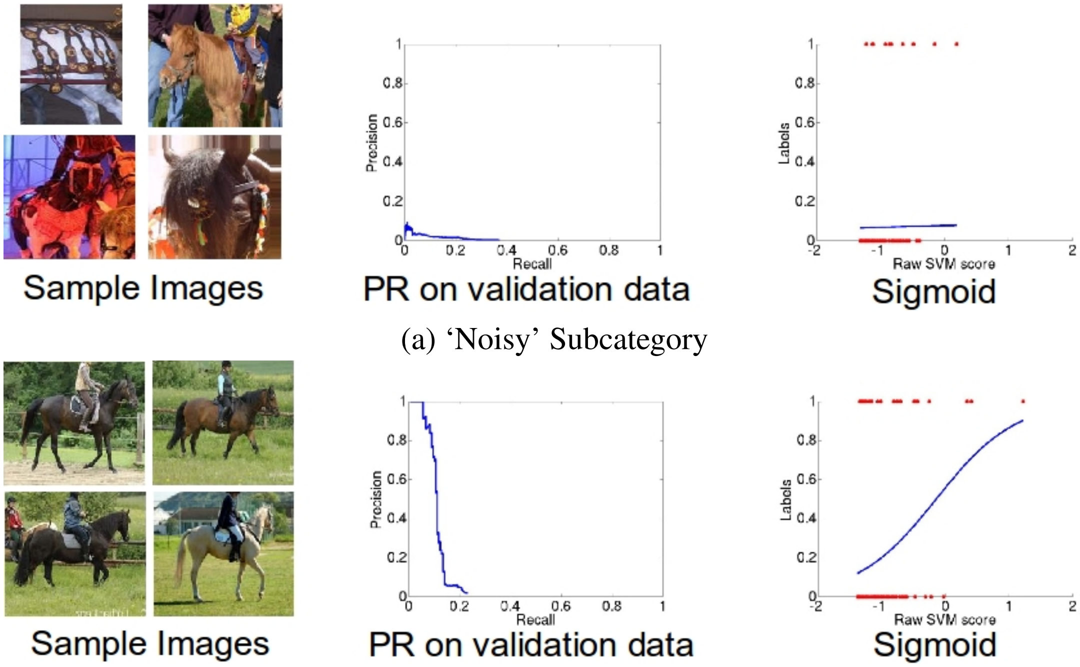
An important parameter to be set is the number of subcategories K. This is unknown apriori and depends on the category. For example, a large rigid object would need fewer subcategories, while a deformable object is expected to require a larger number. To deal with this problem, we initialize the number of subcategories to a large number and subsequently prune the noisy subcategories. The pruning is done via the following calibration step.

Calibration

Unlike previous works that have used a few equal-sized subcategories (3-5 in [78], 3 in [51], 2 in [44]), we explore the use of a large number of unequal-sized subcategories in this work (15-25 visual subcategories for detection experiments and 50 subcategories for scene classification experiments). As a consequence of using a large number of subcategories with different data distributions, we end up with classifiers having varying performance profiles, some of them being quite noisy. Note that, although the subcategory classifiers are coupled in the latent SVM formulation (4.4), a careful observation reveals that the classifiers are actually being learned independently. The coupling of classifiers only happens via the latent step (4.6) i.e., the assignment of positive and negative instances to the different subcategories. Subsequently the SVM learning per subcategory is independent [44, 63]. Therefore it is important to ensure that the scores produced by individual classifiers are calibrated appropriately (so as to suppress the influence of noisy ones) during the reclustering and testing step. Several approaches for calibrating classifiers exist in literature [67, 135, 164, 169]. In this work, we address this problem by transforming the output of each SVM classifier by a sigmoid to yield comparable score distributions [118] (Figure 4.4).

Each detector is applied to all the images in the training set, and the overlap scores with the ground-truth instances are computed. The parameters of the sigmoid are fit by using the detection scores and their corresponding overlap scores (as labels). Since there are several order of magnitude more negatives than positive detections, only the top 500 detections are considered in the fitting process (to avoid any bias in the learned parameters). A *good* subcategory will have high overlap score for its high scoring detections, while a *noisy* subcategory will have low overlap score for its high scoring detections. Given a thresholded output score $s_{i,k}$ for instance i in subcategory k, its calibrated score $g_{i,k}$ is defined as

$$g_{i,k} = \frac{1}{1 + exp(A_k.s_{i,k} + B_k)}, \tag{4.8}$$



(a) 'Noisy' Subcategory

(b) 'Good' Subcategory

Figure 4.4: The classifier trained on a noisy subcategory (horses with extreme occlusion and confusing texture) performs poorly on the validation dataset. As a result, its influence is suppressed by the sigmoid. While a good subcategory (horses with homogeneous appearance) classifier leads to good performance on the validation data and hence its influence is boosted by the calibration step.

where A_k, B_k are the learned parameters of the following logistic loss function L_k:

$$\underset{A_k,B_k}{\arg\min} -\sum_{i=1}^{n} t_i \log g_{i,k} + (1-t_i)\log(1-g_{i,k}), \tag{4.9}$$

$$t_i = Or(W_{i,k}, W_i). \tag{4.10}$$

$Or(w_1, w_2) = \frac{|w_1 \cap w_2|}{|w_1 \cup w_2|} \in [0, 1]$ indicates the overlap score between two bounding boxes [4], W_i is the ground-truth bounding box for the ith training sample, and $W_{i,k}$ indicates the predicted bounding box by the kth subcategory. (All boxes having no overlap with ground-truth i.e., *negative* boxes will have their t_i=0.) Instead of the overlap score $Or(.,.)$, we also tried using a +1/-1 label for fitting the logistic. We treated all detections with overlap score greater than 0.5 as positives and overlap lower than 0.2 are treated as negatives. We found using the $Or(.,.)$ to perform better than the +1/-1 label.

The calibration objective (4.9) is incorporated into the LSVM formulation (4.4) as:

$$\arg\min_{w} \frac{1}{2}\sum_{k=1}^{K}||w_k||^2 + C\sum_{i=1}^{n}\epsilon_i \tag{4.11}$$

$$y_i \cdot s_{i,z_i} \geqslant 1-\epsilon_i,\ \epsilon_i \geqslant 0, \tag{4.12}$$

$$z_i = \arg\max_{k} g_{i,k}, \tag{4.13}$$

$$g_{i,k} = \frac{1}{1+exp(A_k.s_{i,k}+B_k)}, \tag{4.14}$$

$$s_{i,k} = w_k \cdot \phi_k(x_i) + b_k, \tag{4.15}$$

$$(A_k, B_k) = \arg\min_{A_k,B_k} -\sum_{i=1}^{n} t_i \log g_{i,k} + (1-t_i)\log(1-g_{i,k}), \quad t_i = Or(W_{i,k}, W_i). \tag{4.16}$$

An alternating minimization approach is used for solving this non-convex objective function. Given the latent assignment of object instances to subcategories z_i, detectors w_k are first trained for each subcategory k. Fixing the detectors w_k, and the latent assignments z_i, the sigmoid parameters A_k, B_k are learned. Finally the detector w_k and the sigmoid A_k, B_k parameters are fixed to update the latent assignments z_i.

4.2.2 Initialization

A key step for the success of latent subcategory approach is to generate a good initialization of the subcategories. As highlighted in Figure 4.2, it is important to partition the data such that instances that are visually similar are clustered together. Our initialization method is to warp all the positive instances to a common feature space $\phi(.)$, and to perform unsupervised clustering in that space[1]. In our experiments, we found the Kmeans clustering algorithm using Euclidean distance function to provide a good initialization. (Another alternative for initializing appearance-based subcategories using a normalized cuts [139] based algorithm has been discussed in [63].) It must be emphasized that the same feature representation should be used for performing the clustering step as used in training the classifiers. This is critical to ensure that the resultant clusters could be leveraged to train the classifiers. For example, clustering in color space would not be very useful for training classifiers that use HOG features.

[1] It is also possible to use aspect-ratio (or shape masks when available) as an additional feature for the clustering process. Interestingly, we noticed that our appearance-based method automatically separates the instances of different aspect-ratios into separate clusters.

After the initial clusters are generated using the above procedure, we compute the mode aspect ratio of instances within each cluster k and resize all instances within that cluster to this bounding box dimension B_{mode} [44]. We also set the size of the template w_k to B_{mode} for that cluster. This is an important step since it allows us to adapt the feature representation based on the instances within a cluster (see "Template size adaptation" in Section 4.3).

Pooling detections from multiple subcategories

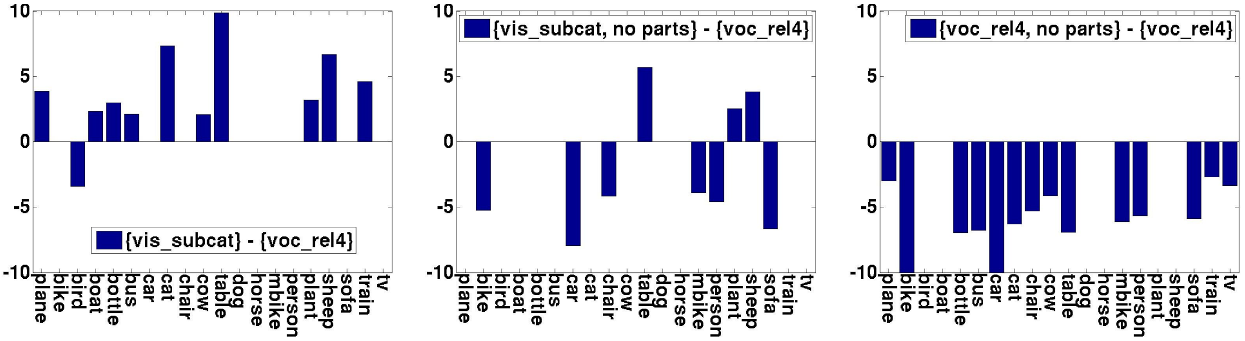
To localize objects in a test image, we scan each individual subcategory detector using the standard multi-scale window scanning framework and retrieve the top detections. Each detection is defined by a bounding box and a score. Multiple overlapping detections per subcategory are eliminated by using a greedy *non-maximum suppression (NMS)* procedure. Given the per subcategory non-max suppressed detections, we calibrate their scores using the learned sigmoid parameters A_k, B_k. To aggregate the detections from multiple subcategories, we use a simple rescoring scheme to leverage the fact that it is possible for multiple subcategories to produce high scoring detections on the same instance (due to the sharing of instances across multiple subcategories during training). Our approach is to rescore the most confident detection by setting its score to the sum of the scores from its overlapping detections before discarding them. We found this simple scheme in combination with the calibration step to improve performance (mean A.P. increase of 0.5%).

4.3 Experimental Results

We performed our analysis on the PASCAL VOC 2007 comp3 challenge dataset [39]. We used the standard PASCAL VOC comp3 test protocol, which measures detection performance by average precision (AP) over different recall levels.

As our baseline system, we use the concurrent release of the Deformable Parts Model (DPM) detector [46] (without the bounding-box prediction and context-rescoring steps). The DPM detector has emerged as a useful and popular tool for tackling the intra-category diversity challenge. Its success in the PASCAL VOC competition has drawn attention from the entire vision community towards this tool, and subsequently it has become an integral component of many classification, segmentation, person layout and action recognition tasks.

Figure 4.5 compares the results obtained using the different methods with respect to the base-



(a) Subcategories, with part deformations

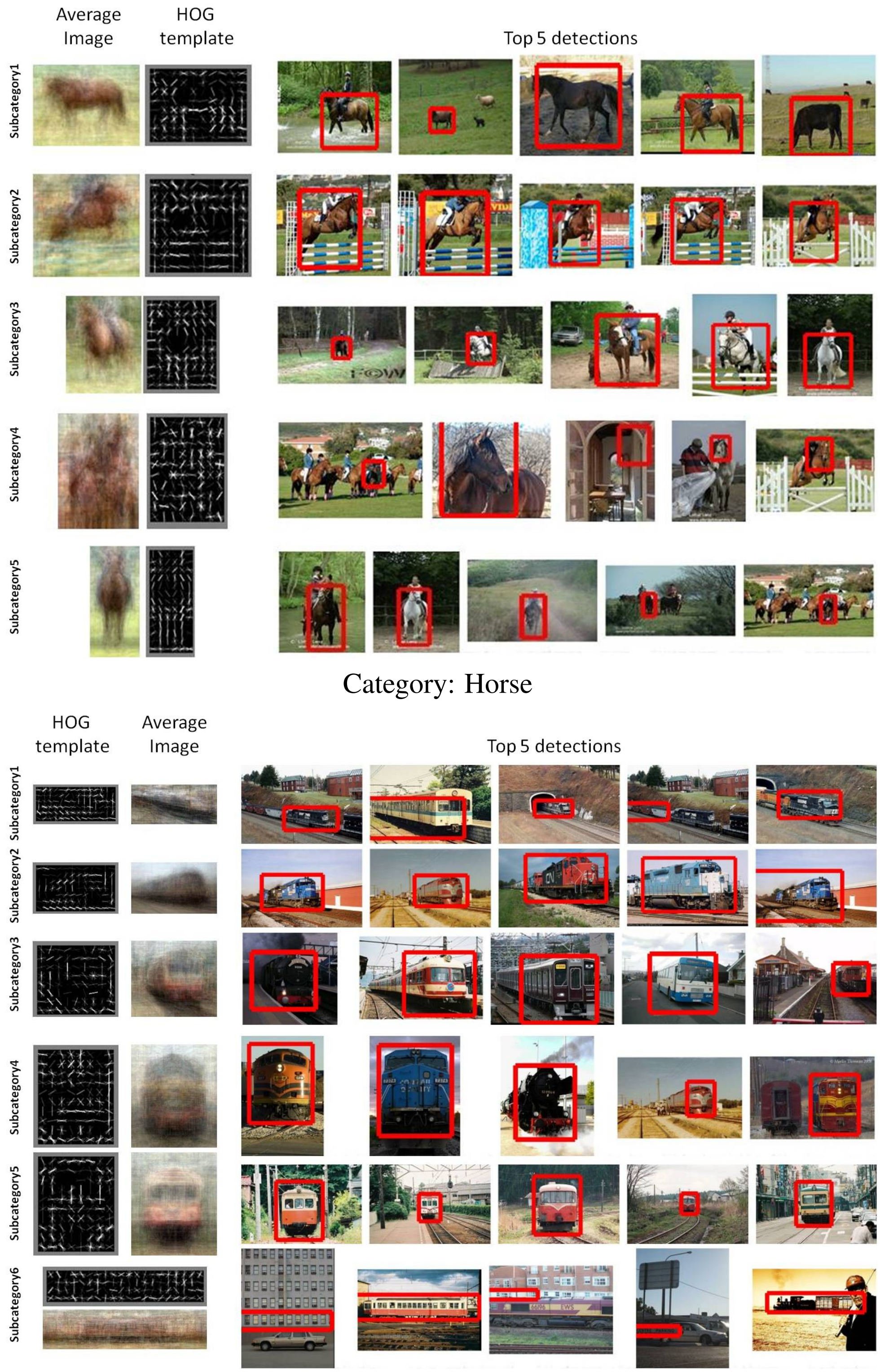
(b) Subcategories, no part deformations

(c) [46], no part deformations

Figure 4.5: Performance difference with respect to the baseline [46] (x-axis: 20 VOC classes, y-axis: difference in A.P.).

line for the 20 PASCAL object categories. The first sub-figure shows the improvements offered by using visual subcategories (with K=15) in the DPM detector. The mean relative improvement (over the baseline) across 20 classes is 9.4% (the mean A.P. improves from 0.32 to 0.35). Figure 4.6 shows the top detections obtained per subcategory for the horse and train categories. The individual detectors do a good job at localizing instances of their respective subcategories. In Figure 4.7, the discovered subcategories for symmetric (pottedplant) and deformable (cat) classes are displayed.

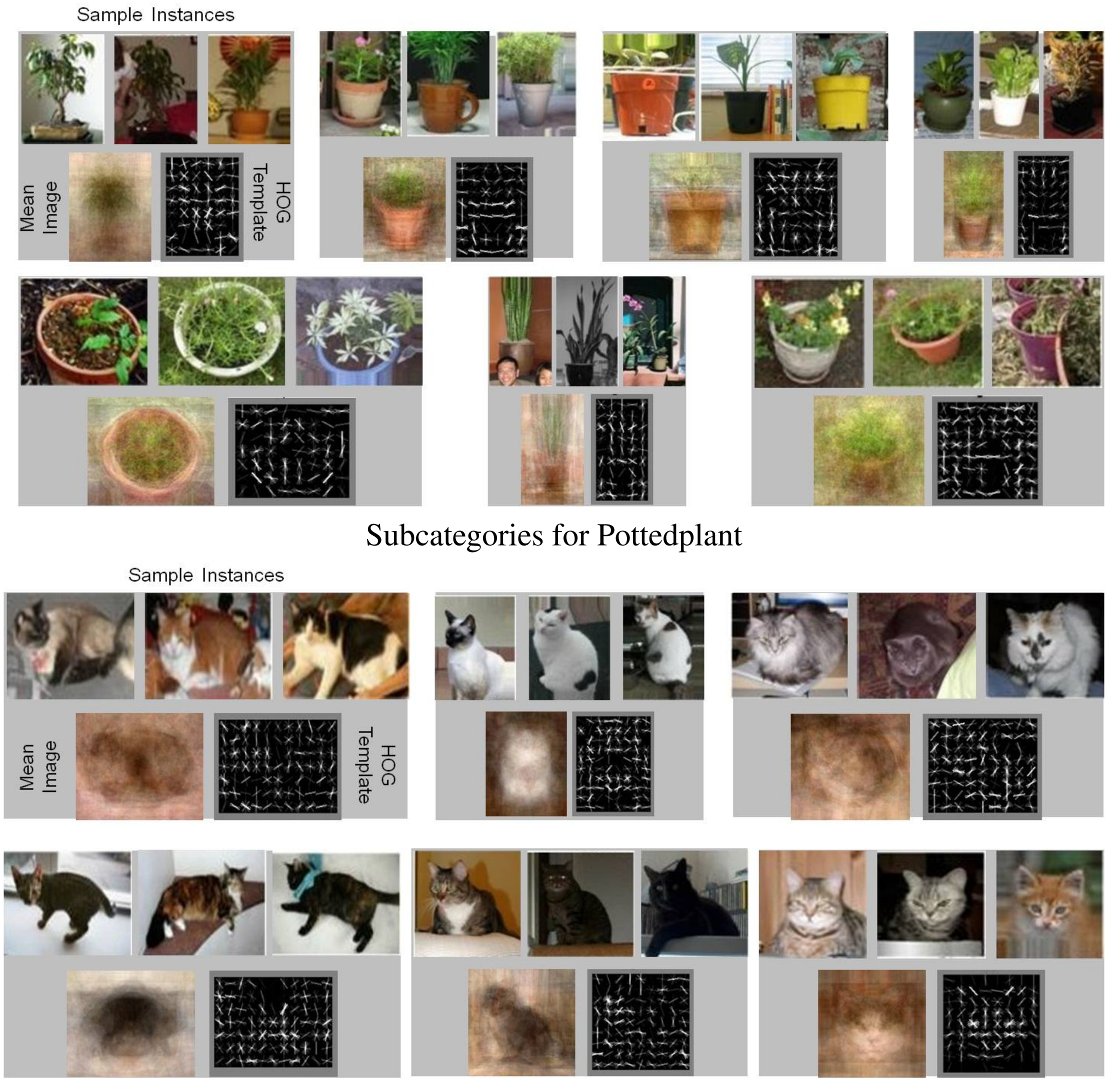
Number of subcategories. One important parameter of our system is the number of subcategories K (we use $K = 15$ in our experiments). There is an interesting trade off between the number of subcategories and the size of the training data. If K is too large, then the subcategories are compact and homogeneous, but at the cost of having very little data per subcategory. On the other hand, using a small K results in more data per subcategory, however they would be less homogeneous. We analyze the influence of K by running our setup with different values ($K = [1, 2, 6, 15, 25, N]$), where N is the number of exemplars per category. (For computational reasons, as rerunning our setup for all classes for every K value is computationally intensive, we use a simple root-template based model with the *deformable parts* turned off for running these experiments.) We plot the variation of the performance over the different number of subcategories in Figure 4.8. The performance gradually increases with increasing K, but stabilizes around $K = 15$. (Although it is possible to choose the best value for K per category by cross-validation, we use a fixed value in our experiments for computational reasons.)



Category: Horse

Category: Train

Figure 4.6: As the intra-class variance within subcategories is low, the learned detectors perform quite well at localizing instances of their respective subcategories. Notice that for the same aspect-ratio and viewpoint, there are two different subcategories (rows 4,5) discovered for the train category.



Subcategories for Pottedplant

Subcategories for Cat

Figure 4.7: The visual subcategories discovered for pottedplants correspond to different camera viewpoints, while cats are partitioned based on their pose. The baseline system [46] based on the aspect-ratio, left-right flipping heuristic cannot capture such distinctions (as many of the subcategories share the same aspect-ratio and are symmetric).

Unsupervised vs. Supervised Initialization. Proper initialization of subcategories is a key requirement for the success of latent variable models. While latent models have been used earlier [44], one of the key differences of this work is in the initialization step. Here we compare and evaluate the unsupervised features-space clustering scheme used in this work to few other popular supervised initialization schemes: viewpoint [23], taxonomy [168], and aspect-ratio [44].

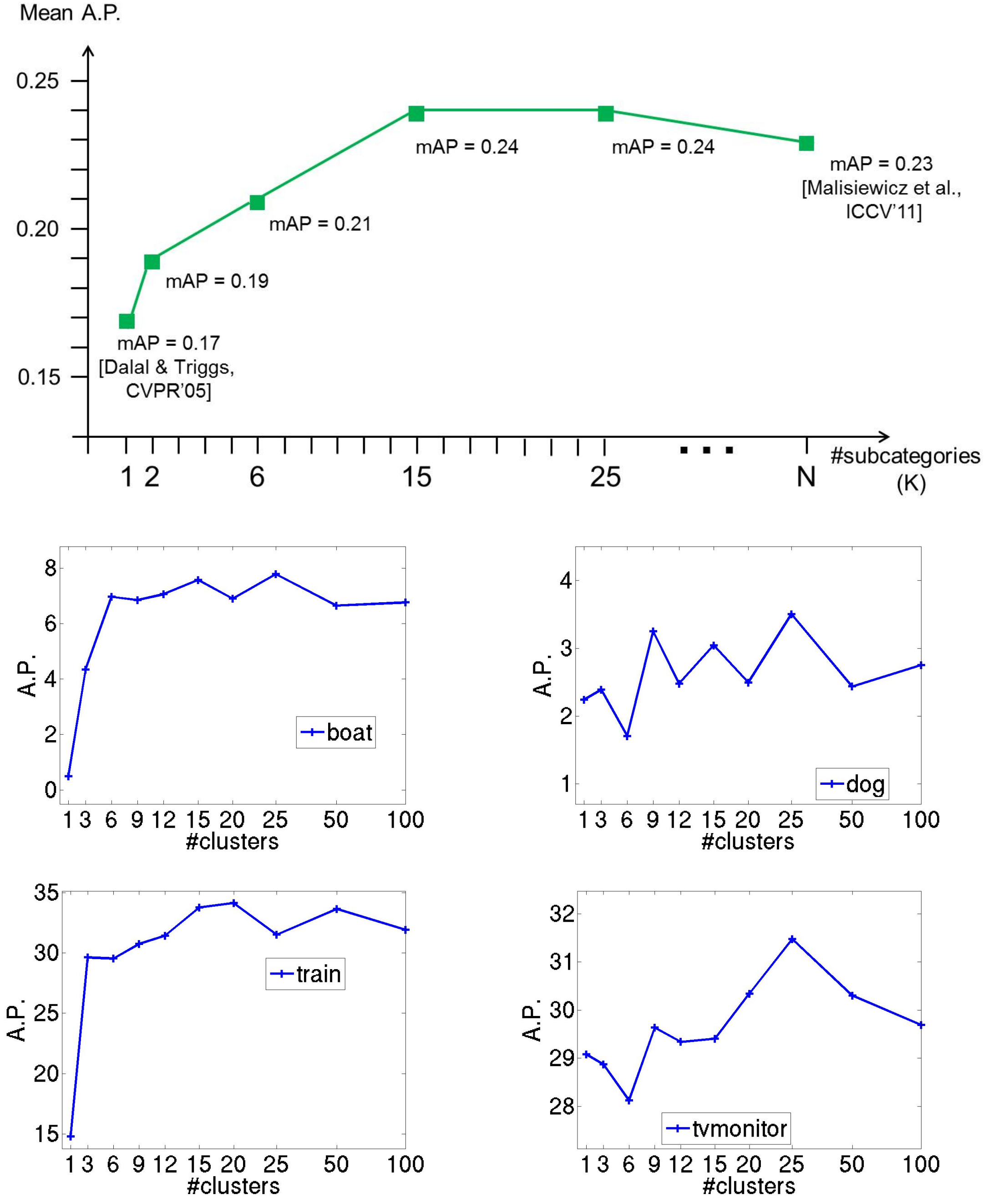


Figure 4.8: Variation in detection accuracy as a function of number of subcategories. (Top) Result obtained across 20 VOC classes, (bottom) result on four selected classes. The A.P. gradually increases with increasing number of subcategories and stabilizes beyond a point.

- Viewpoint: The viewpoint annotations associated with the ground-truth bounding boxes in the PASCAL VOC trainval dataset are used to initialize the subcategories within the mixture model for training the category detectors [23]. Viewpoint annotations, namely 'left', 'right', and 'frontal' resulting in K=3 subcategories, are provided for 6 out of 20 classes in the VOC 2007 dataset. The mean A.P. obtained across the 6 classes with this initialization was 0.37. In comparison, the result using unsupervised (appearance clustering) initialization for the 6 classes with K=3 was 0.38, which is on par with the supervised initialization scheme. (A similar observation has been made in [63], where they find unsupervised clustering to perform on par with viewpoint-based clustering with the same K on the PASCAL VOC 2007 "car" object category.) However increasing K to 15 in case of appearance initialization improved the mean A.P to 0.43.
- Taxonomy: In this case, the subcategories are initialized using fine-grained subcategory annotations provided by humans based on a semantic taxonomy. We conducted this experiment on the MIT SUN database [168] that has such annotations available. This initialization based scheme produced a mean A.P. (across the 15 scene categories) of 0.27, while the unsupervised clustering method got a mean A.P. of 0.27 (see Section 4.6.2 for more details). A similar observation has been reported in [29], where unsupervised initialization performed on par with taxonomy-based initialization on a subset of IMAGENET dataset [27].
- Aspect-Ratio: In this case, the subcategories are initialized using an aspect-ratio heuristic. The baseline system of Felzenszwalb et al. [46] uses such an initialization (aspect-ratio is used to split the training data into three subcategories followed by left-right bilateral clustering, resulting in a total of K=6 subcategories). This aspect-ratio initialization produced a mean A.P. of 0.19, while the unsupervised initialization (with K=15) produced a mean A.P. of 0.24. (We use the detector implementation with the deformable parts turned off for this experiment.) When $K = 6$, aspect-ratio and appearance initialization produced a similar result. Increasing the number of subcategories from K=6 to 15 in case of the aspect-ratio clustering dropped the mean A.P to 0.17. Figure 4.9 illustrates a few differences between the initialization schemes.

In case of the unsupervised clustering scheme, we noticed minimal variation in the final performance on multiple runs with different Kmeans initialization. We found the (latent) discriminative reclustering step helps in cleaning up any *mistakes* of the initialization step. Also we observed that most of the reclustering happens in the first latent update.

In summary, from our experimental analysis, we observe that unsupervised initialization based on feature-space clustering either does as well or better than other supervised initialization schemes. This indicates that human supervision may not be necessary for generating subcategories within the latent SVM framework. Future work could explore other schemes (based on random initialization or brute-force search over initial mixture assignments [10]) for generating the subcategories and study their effects.

Ensemble of subcategories. What happens when we combine the subcategory classifiers from different K settings and build a single ensemble-like [165, 167] classifier encompassing all of them? We investigated this question by running an experiment in which we combine subcategory classifiers from $K = 3, 6, 15$ by calibrating them together and merging them into a single (24-subcategory) model. We notice a significant improvement in performance (mean relative improvement of 5.1% over the baseline). As each sub-classifier is trained on different folds of the training data, combining them yields an improved classifier. Further, some training instances may be ambiguous and may well be classified into two or more subcategories, and thus forcing the decision between them may not be optimal. Using an ensemble-like model allows the formation of overlapping subcategories. In Chapter 5, we will see another approach for sharing training instances across subcategories.

Template size adaptation. To emphasize the importance of adapting the feature template to the individual subcategories, we conducted an experiment where the feature template for the subcategories are fixed to a standard size. More specifically, instead of setting the feature template size based on the mode aspect ratio of the ground-truth object bounding boxes within each subcategory, it is set as the mode aspect ratio of the instances across all subcategories. We noticed the mean A.P. to drop from 0.24 to 0.20, underlining the importance of the feature adaptation step.

Other datasets. In order to study the performance of our algorithm on other datasets and to avoid the risk of overfitting to VOC2007, we ran our experiments on the more recent VOC2010 dataset. We continue to see improvement in results. Mean A.P. improves by 3.5% over the baseline.

Benefit of additional labeled data. For many vision tasks, simply adding more labeled data to existing learning methods does not lead to performance improvement [105]. How does the

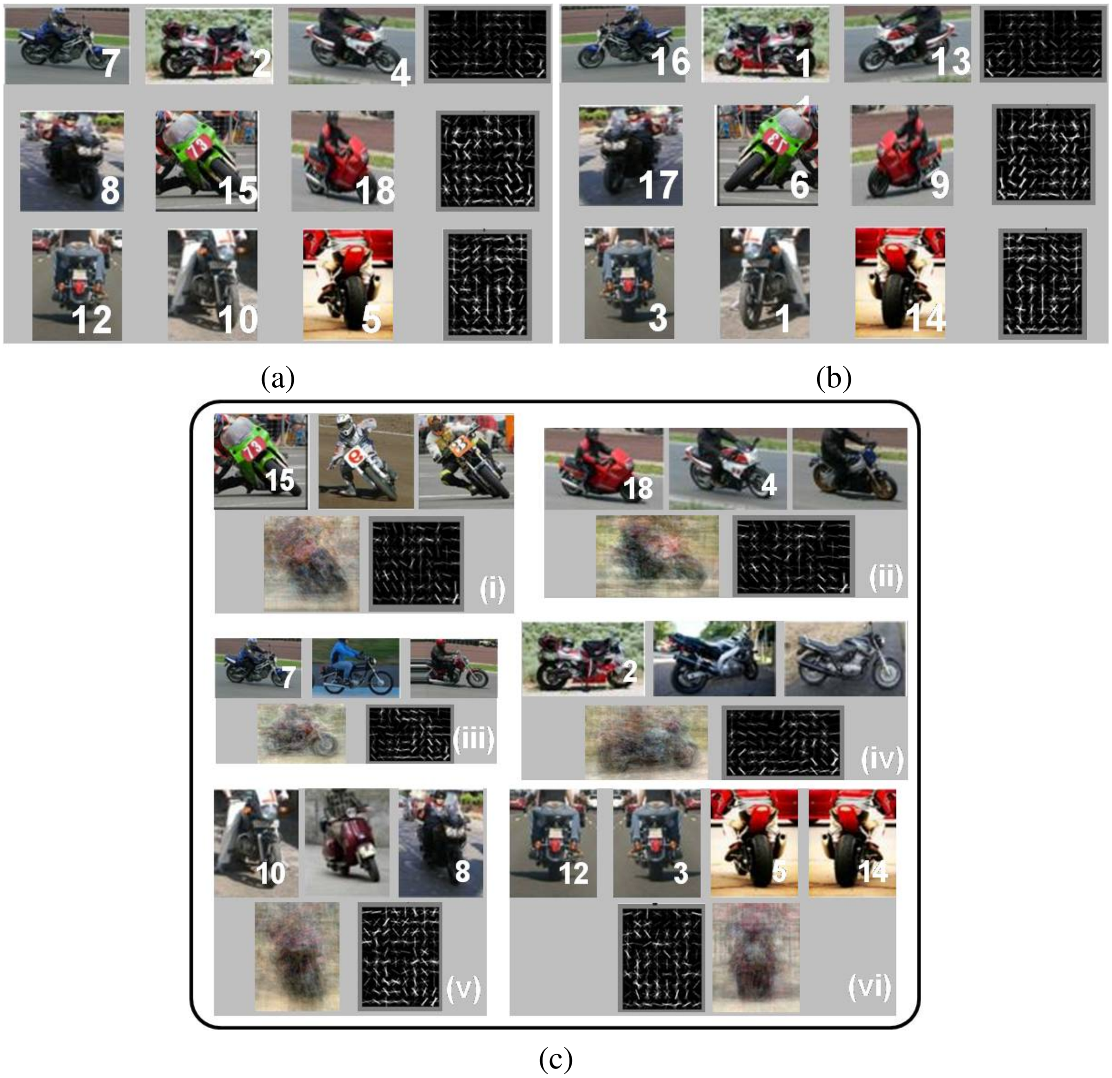


Figure 4.9: Unsupervised feature-space clustering approach produces better performing and more interpretable subcategories when compared to the aspect-ratio or left-right clustering approach [44, 46]. (a-b) Subcategories obtained using 3 aspect left-right clustering method and the corresponding models learned. There is significant amount of appearance variation within the generated clusters (e.g., instances 7, 2, 4 in top row of (a)) and many poor cluster assignments (e.g., instance 8 and 10 appear similar but end up in different clusters). Further, for symmetric clusters, left-right split leads to redundant clusters (e.g., instances 12, 5, 3, and 14 in bottom row of (a-b)). (c) Subcategories obtained by our proposed approach. (Clusters (i-v) have their corresponding left-facing versions that have not been displayed due to space constraints.)

subcategory approach fare in the case of additional labeled training data?

To test this point, we ran an experiment by adding extra training data to the VOC2007 dataset.

Care must be taken in choosing the additional data: simply adding extra data from any arbitrary external source might result in classifiers drifting away from the given train-test data distribution. To cope with this problem, we use additional ground-truth data from the VOC2010 dataset that has approximately the same distribution as the VOC2007. The VOC2010 trainval set offers about 3X more data compared to VOC2007 trainval for most classes. We repeat our experiments on the new augmented training set and compare the performance to the one trained just using the VOC2007 trainval. We also repeated the same experiment by adding more data to the baseline method [44].

We notice that adding more data does result in improved performance indicating that there is indeed some hope in benefiting from the proposed work of leveraging unlabeled data (mean relative improvement of 13.8%). Interestingly, the extra training data yields a higher performance increase in case of visual subcategories compared to the aspect-ratio based approach (which gets a mean relative improvement over baseline of 8.8%). This is in fact an important feature that is a direct consequence of using a large number of subcategories, instead of the previous limited clusters approach. The larger data set provides a better coverage of the within-class appearance variability, which our approach "has room to accommodate", while the baseline method saturates and cannot adapt well to the new subclasses emerging from the additional data. Similar observation has been made in [173], where the performance of a mixture model was analyzed in the context of introducing 10X more labeled data. Along with increase in the number of subcategories within the mixture model, [173] emphasizes the need for cross-validation of regularization parameters for benefiting from the use of additional data.

4.4 Analysis: Deformable Parts vs. Subcategories

Deformable Parts have recently emerged as another useful and very popular tool for tackling the intra-category diversity challenge. The idea here is to represent an object model using a lower-resolution "root" template, and a set of spatially flexible high-resolution "part" templates. Each part captures local appearance properties of an object, and the deformations are characterized by links connecting them. The success of the Deformable Parts Model (DPM) detector of Felzenszwalb et al. [44] has drawn attention from the entire vision community towards this tool.

In order to deal with significant appearance variations that cannot be tackled by the deformable parts, [44] introduced the notion of subcategories into their detector. The first version of their detector [47] only had a single subcategory. The next version [44] had two subcategories that were obtained by splitting the object instances based on an aspect ratio heuristic. In the latest version [46], this number was increased to three, with each subcategory comprising of two bilaterally asymmetric i.e., left-right flipped models (effectively resulting in 6 subcategories). The introduction of each additional subcategory has resulted in significant performance gains (see top row in Figure 4.10).

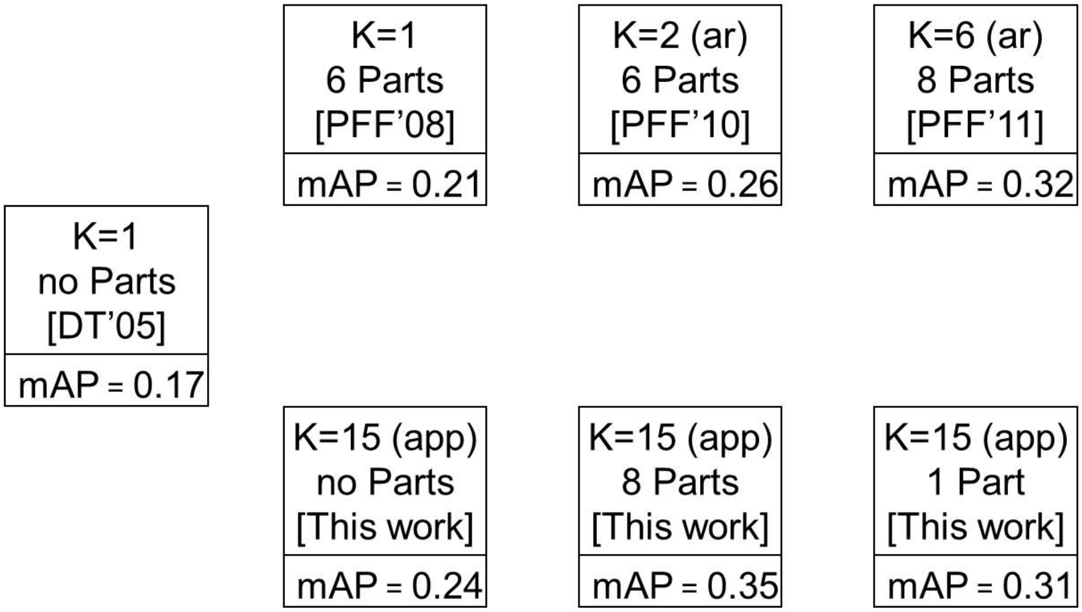


Figure 4.10: Subcategories vs. Deformable Parts analysis. This figure summarizes the results (mean A.P. across 20 VOC classes) obtained by using different detectors on the PASCAL VOC 2007 dataset. DT05 refers to the HOG+linear SVM detector used in [26]. (Top row) PFF08 is the first version of the DPM detector used in [47]. The next version is PFF10 [44], in which the number of subcategories (K) was increased from 1 to 2. In the latest version PFF11 [46], the number of subcategories was increased to 6. (Bottom row) The first block indicates the result obtained using a simple root-template based model (with the deformable parts turned off). The second block indicates the result obtained with the deformable parts turned on. The third block is the result obtained using the two-scale feature representation (indicated as "1 part"). "ar" indicates aspect-ratio based initialization used to generate subcategories, while "app" indicates unsupervised initialization based on feature-space clustering.

In this section, we will study the relationship between the role of deformable parts and subcategories (mixture model components) within this detector, and understand their relative importance. Recall that in Section 4.3, we found that by increasing the number of subcategories, and switching the initialization step from their aspect-ratio, left-right flipping heuristics to appearance-based clustering, considerable improvement in performance was obtained. Apart from the performance gain, an important benefit offered by this update was that the instances within each of the subcategory were highly aligned to each other.

Given the high-degree of alignment across the instances within each subcategory, it is interesting to now check the importance of modeling the deformations across the parts within each subcategory. Would a simpler model without deformations suffice for training the discriminative detectors? We tested this hypothesis with an experiment by turning off the deformable parts. More specifically, rather than sampling "parts" from the high-resolution HOG template (sampled at twice the spatial resolution relative to the features captured by the root template) and modeling the deformation amongst them, we directly use all the features from the high-resolution template. This update to the DPM detector results in a simple multi-scale (two-level pyramid) representation with the finer resolution catering towards improved feature localization.

Figure 4.5(b) displays the results obtained. We observe that for 11 of the 20 classes (e.g., pottedplants, tvmonitor, trains) there is no difference in performance. For 6 classes (e.g, person, sofa), turning off deformations hurt the performance, while for 3 classes (e.g., diningtable, sheep) performance actually improves. On average, using this two-level pyramid representation[2] for the visual subcategories yields a mean A.P. of 0.31 that is almost on par as the full deformable parts baseline (0.32). (To test the generalizability of this observation across detectors and to avoid overfitting of any parameters specific to [46], we repeated this experiment on the release3 version of the DPM detector [45] and observed the same behavior.) These observations suggest that, in practice, the relatively simpler concept of visual subcategories is indeed an equally important contribution in the DPM detector. They can potentially replace the need for part deformations for many object categories.

Computational Issues. In terms of computational complexity, the two-scale visual subcategory detector (K=15) involves one coarse (root) and one fine resolution template per subcategory, totaling a sum of 30 HOG templates. Whereas the DPM detector has K=6 subcategories each with one root and eight part templates, totaling 54 HOG templates, which need to be convolved at test time. In terms of average run time, it takes about 7 seconds for the two-scale subcategory detector, and about 9 seconds for the DPM detector on a 1.1Ghz 8 core AMD Opteron 2354 with CentOS 5.5 running Matlab R2010a. In terms of model learning, the DPM detector has

[2] We found the multi-scale representation to be important. The setup that just used the higher resolution features without the coarser lower resolution features (from the root-template) produced an inferior result, emphasizing the need for multi-scale representation. A similar observation was reported in [47].

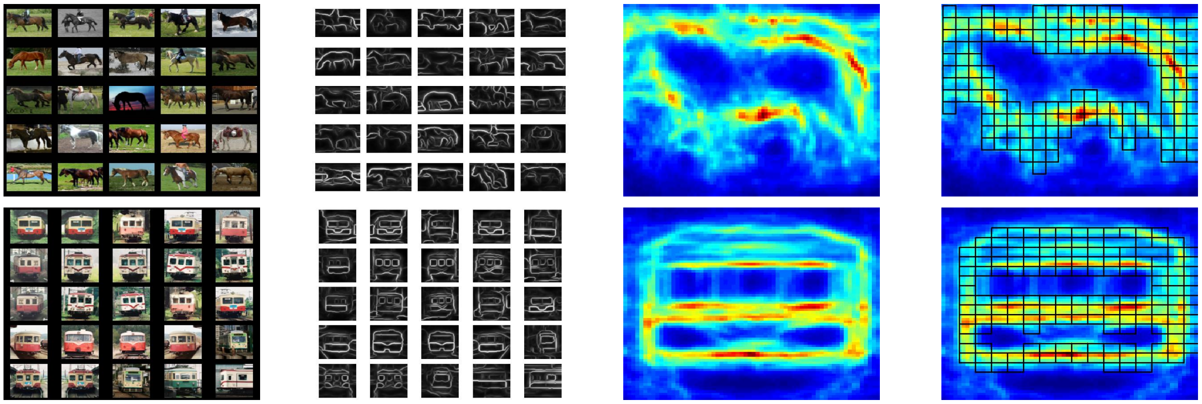
the subcategory, as well as the part deformation (six) parameters as latent variables for each of the 24 parts (total of 145 latent variables), while the visual subcategory detector only has the subcategory label as latent. Therefore it not only requires fewer rounds of latent training than required by the DPM detector (leading to faster convergence), but also is less susceptible to getting stuck in a bad local minima. In terms of the average training time, it takes about 8.3 hours for the two-scale subcategory model, and about 10.1 hours for the DPM detector. As emphasized in [44], simpler models are preferable, as they can perform better in practice than rich models, which often suffer from difficulties in training.

Given that deformable parts can potentially model exponentially large number of object deformations [44], it is expected that their use would lead to much better results and greater generalizability than the use (of a fixed number) of subcategories. However our empirical analysis has surprisingly pointed out that there is only a minimal performance gap between the use of part deformations and subcategories in the DPM detector. Further, the fact that a simple method (more subcategories, no parts) does almost as well as the relatively more complex method (fewer subcategories, with parts) is informative as the former is conceptually easy to understand and implement, computationally efficient, and generates easily interpretable detection models.

4.5 Adapting Feature Representation

An important benefit of the subcategory-based approach is that it offers the flexibility of choosing different feature representations and models (classifiers) for different subsets of data. (Note that the interplay between the subcategories happens only via the maximization step that only requires a calibrated score.) This freedom enables us to design and use improved representations and models that best suit the instances within each subcategory. In this section, we will highlight one particular benefit of adapting the feature representation per subcategory detector in the context of the sliding-window approach.

A typical 32 dimensional HOG-based representation [44] with a 8×8 cell griding (each of 8×8 pixels) comprises 2048 dimensions. In case of the two-scale pyramid representation highlighted in Section 4.4, including the 16×16 grid (at twice the spatial resolution of the root) would increase the number of dimensions by 8192. Instead of directly using all the features from the finer resolution for representing the subcategory model, is it possible to use a simpler representation that is of lower dimension but adapted to the subcategory instances? We explore



(a) Subcategory instances (b) Low-level contours [6] (c) Averaged contour map (d) Multi-resolution griding

Figure 4.11: Multi-resolution feature adaptation (best viewed in color). Given the (a) subcategory bounding boxes, we extract (b) low-level contours using [6]. (c) Average of the individual low-level contour maps across all subcategory instances. We chose high-resolution (4×4, black) cells in regions of high gradient energy.

a simple approach for adapting the feature representation by (again) leveraging the high degree of alignment between images in each subcategory.

A typical solution to cope with high-dimensional feature spaces is to selectively adapt the feature space i.e., to pick a relevant set of features from an overcomplete set [107, 137]. This problem is often formulated as a joint (latent) estimation problem where the feature selection (or induction) and model learning steps are iteratively optimized [5, 66, 101]. A common downside of such methods is that they are prone to getting stuck in a bad local optimum [83]. Instead in this work, we seek to alleviate the burden of large dimensional spaces by leveraging the fact that the subcategories discovered by our approach contain instances that are all well-aligned in the feature space. We use this to our advantage to compute average statistics of bottom-up low level contour cues to prime the feature space in order to obtain a compact and discriminative representation (see Figure. 4.11).

Our selection strategy is to assign more cells to regions of high contour strength as they are characteristic of the object appearance. Given the instances of a subcategory, we first warp them to the mode aspect-ratio bounding box dimension. Low-level contours are then extracted using the approach of Arbelaez et al. [6]. We then compute the mean response by averaging the contour maps across all the instances (we use a robust mean estimate to reject outliers). The values of the mean magnitude response are thresholded to select cells of high energy. We use a threshold such that half of the cells at the finer resolution are selected. Figure 4.11 illustrates our approach.

The chosen cells at the finer resolution are initialized by interpolating the root filter to twice the spatial resolution. The spatial convolution is implemented in the standard manner except that we explicitly zero out the cells that were not selected in our selection scheme.

The proposed method is attractive as we directly incorporate bottom-up cues into the standard highly-efficient sliding window based HOG detection pipeline. In terms of performance, the adapted feature representation achieves a mean A.P. of 0.31, the same as the result using the full feature representation. We also compared our proposed scheme to a baseline that randomly selects half of the cells from the finer resolution on a few classes (airplane, bikes, tables and sofa) and found it to perform poorer (mean relative drop of 9.6%), indicating the merit in our bottom-up contour based selection strategy.

4.6 Beyond Object Detection: Role of Subcategories in Scene Classification

Another scenario where the problem of high intra-class variability is observed is scene classification. Scene categories exhibit a large range of visual diversity due to significant variation in camera viewpoint and scene structure. For example, when we refer to the scene category ‘coast’ (from [168]), it could contain images of rocky shores, sunsets, cloudy beaches, or calm waters. From our analysis of visual subcategories on the object detection dataset, we could expect that their use could also aid in simplifying the learning task for scene classification. In this section, we present the results of applying our approach on two scene classification tasks - the PASCAL VOC object-image classification and the SUN397 scene-image classification.

The goal here is to train a classifier that can identify images belonging to a specific semantic category such as highway, or sunset, etc. Implementation-wise, image classification is a simpler problem compared to object detection as it does not involve scanning windows at multiple scales and locations and thereby involves a much simpler training process (with no computationally expensive data-mining steps). We compare the performance of our approach to two baseline methods: a single global linear SVM, and a single global RBF-kernel SVM. The classifiers are all trained in a ‘1-vs-all’ fashion, where instances belonging to a specific category are considered ‘positive’ examples and the rest of the instances (belong to all the other categories except the chosen one) serve as ‘negative’ examples in the training process. While training subcategory classifiers, instances belonging to a particular subcategory (within a category) are treated as 'positives' and the rest of the instances belong to that category are ignored (treated as 'don't care' examples).

We use the GIST feature representation (using the implementation of [109]), that has been well-studied in the literature for scene classification (e.g., [29]). We create this descriptor for each image at a $m \times n$ (10×10) grid resolution where each bin contains that image patch's average response to steerable filters at 8 orientations and 4 scales. We acknowledge that a classification system based on this representation is unlikely to beat the prevailing state-of-the-art. Multiple previous methods have shown that classification performance is significantly improved by the use of BOW-models with densely sampled feature points along with multiple sets of feature descriptors, and the use of spatial pyramids. We chose the simple GIST-based representation for our analysis as our focus was not to argue in favor of a new classification method, but to show the benefits of the visual subcategory concept using a generic and simple framework.

Equation 4.6 is implemented by simply applying each subcategory classifier w_k, b_k to every ground-truth instance in the training dataset (irrespective of its initial cluster assignment). The best scoring subcategory for an instance is selected and its cluster assignment is updated accordingly. Initialization of the clusters is performed using a Kmeans-clustering algorithm with a Euclidean distance function. The calibration of models is performed as described in Section 4.2.1.

4.6.1 PASCAL VOC Object-Image Classification

The goal in the PASCAL VOC object-image classification task is to predict the presence or absence of a specific object class such as airplane, cow, etc in the test image. Each image may contain zero or more examples of a specific object class or different object classes. The dataset contains 20 object classes and the performance is judged using precision-recall curve computed per object class (comp1 challenge).

In table 4.1, we report the results using the different methods. The baseline systems of a single linear SVM and non-linear SVM are trained using a 10×10 GIST feature representation. Our approach is initialized using $K = 15$ subcategories. The results obtained using visual subcategories significantly surpasses the results obtained using the baseline linear SVM. In Figure 4.12, we show the top images retrieved per subcategory for the a few object classes. The individual subcategory classifiers do a good job at retrieving instances of their respective clusters.

We also ran experiments in which we combine sub-classifiers from $K = 3, 5, 10, 15$ by calibrating them together and merging them into a single 33-cluster model (as motivated in Section 4.3). We continue to notice a performance gain on average across 20 classes when compared

	linearSVM	non-linearSVM	Ours	Ours (hierarchy)	Our (nonlinearSVM)
aeroplane	42.7	58.0	55.3	55.7	56.2
bicycle	14.7	27.4	24.6	25.8	29.5
bird	14.5	21.7	18.4	18.9	20.6
boat	30.5	40.9	35.5	36.1	42.7
bottle	6.8	10.0	7.9	8.9	13.4
bus	16.5	25.6	20.9	24.4	29.5
car	42.1	53.7	48.4	50.0	55.4
cat	18.8	26.6	21.3	23.1	27.2
chair	23.3	32.8	28.4	29.1	35.4
cow	6.3	10.5	9.0	9.2	14.8
diningtable	13.5	27.2	15.4	17.5	28.7
dog	15.1	21.3	18.4	19.7	24.0
horse	30.1	63.2	52.6	54.7	60.2
motorbike	14.5	33.2	27.6	27.2	37.2
person	59.7	66.9	57.4	59.1	65.7
pottedplant	7.1	10.0	8.6	8.0	9.4
sheep	7.6	13.3	6.7	6.9	11.8
sofa	11.9	21.7	16.1	16.5	25.6
train	32.6	48.8	39.4	41.1	47.3
tvmonitor	17.7	25.2	20.7	21.3	28.1
MEAN	21.3	31.9	26.6	27.6	33.1

Table 4.1: Results on VOC2007 object-image classification task. The first two columns display the baseline results obtained using a single global linear and nonlinear SVM. The third column is the result obtained using our visual subcategory classifiers. The fourth column is our approach with a pseudo-hierarchical scheme of using the visual subcategories. The last column is our approach while using nonlinear SVM for each individual subcategory.

to just using K = 15 model even in this scenario.

Comparison to non-linear SVM. Although the performance of our visual subcategory approach does not excel that of the non-linear SVM, it offers a tremendous computational benefit [11, 50, 51]. Since linear classifiers are faster to train than nonlinear ones, training several independent linear SVMs is thus far more efficient and easier to parallelize than building a monolithic non-linear SVM. For example, we found the RBF-SVM in our implementation to be approximately 100X slower than a single linear SVM (1.5 hours vs. 1 minute on the SUNS dataset using libsvm [21]). Although using K=50 makes our approach 50X slower than a single

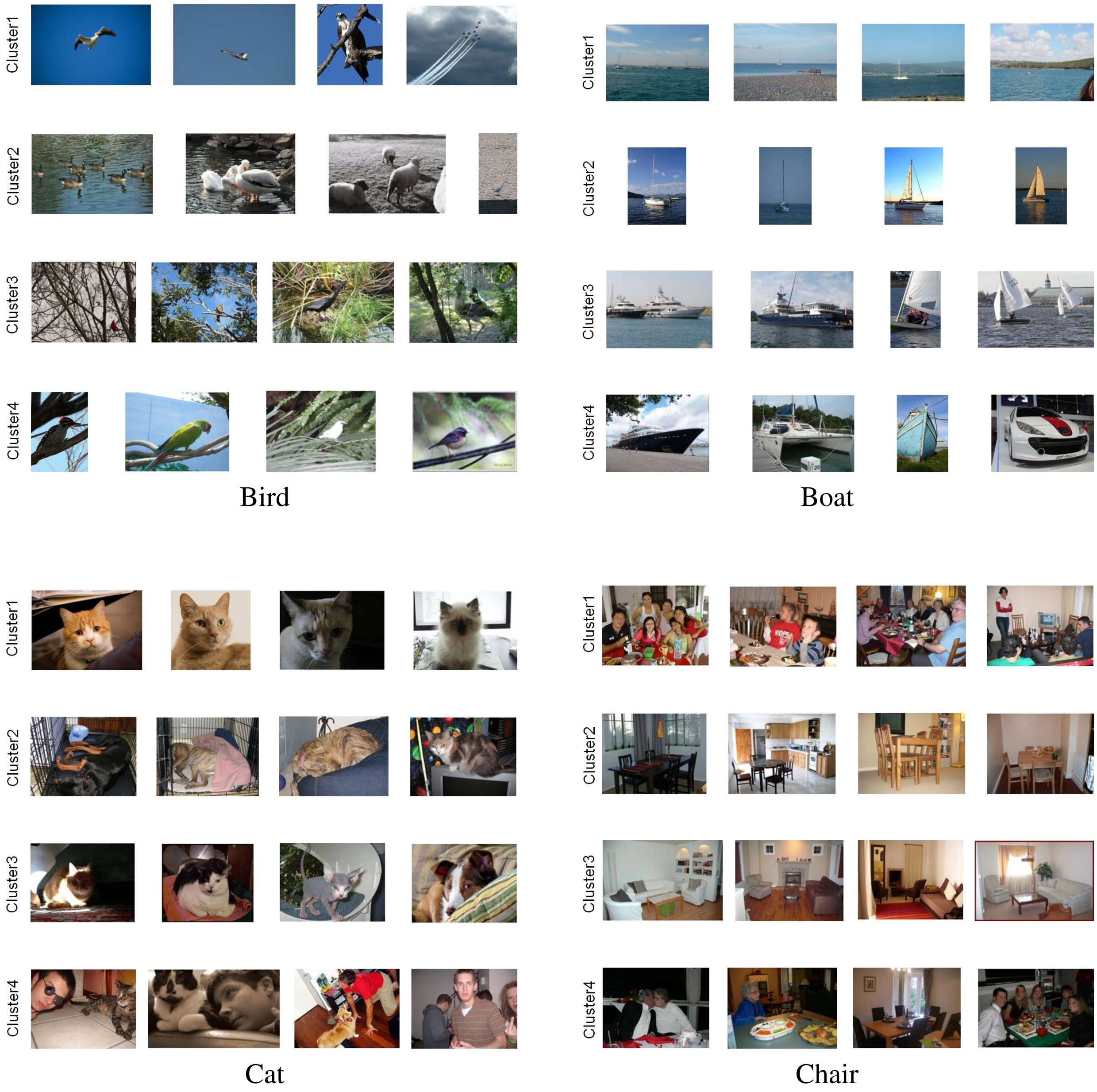


Figure 4.12: Results on VOC2007 Object-Image Classification: Object-Image categories exhibit a large visual diversity due to significant variation in camera viewpoint and scene structure. The ‘bird’ category contains separate subcategories for birds flying in sky, birds in water, birds on cluttered branches and birds oriented left perched on a branch. The ‘Boat’ category has clusters for ocean views, two different types of boats, and boats anchored in a harbor. ‘Cat’ category has clusters for cat faces, sleeping cats, cats in a specific pose and cats with people images. Finally the ‘Chair’ category contains clusters of group dining shots, dining areas, living room, and people seated in front of dining table. The subcategories are discovered in a data-driven manner and help in obtaining improved basic-level categorization performance.

linear SVM, the subcategory models were trained in parallel. Further, linear classifiers are faster and less memory-intensive at classification time: Using a non-linear SVM typically results in several hundreds or thousands of support vectors (depending on size of the dataset) that need to be stored in memory, and to which distances in the kernel space need to be computed at test time for every query sample. In comparison, the subcategory approach involves just a few dot-product operations (depending on the number of subcategories K). Finally, linear classifiers are much more interpretable in that their weights indicate which features are important for classification, and how they affect predictions. In many applications, being able to understand how a classifier makes predictions is almost just as important as having a high accuracy [159].

Nonetheless, we also report results obtained for our approach by training individual subcategories using a non-linear SVM to compare the performance. As shown in table 4.1, the performance of our system in this case surpasses that of a single non-linear SVM indicating that our contribution is still significant even in case of non-linear classifiers. However, since the non-linear SVM can represent more complicated decision boundaries and separate odd-shaped classes better, it stands to benefit less compared to the linear SVM.

4.6.2 MIT-SUN397 Scene-Image Classification

We use the Scene Understanding (SUN) database for our scene classification experiments. The SUN database is a collection of about 100,000 images organized into a set of 899 scene categories [168]. It is constructed by identifying words in a dictionary corresponding to various types of places, scenes, and environments. For our experiments, we use the subset of 397 well-sampled categories. These 397 fine-grained scene categories are arranged in a 3-level tree: with 397 leaf nodes (subordinate categories) connected to 15 parent nodes at the second level (basic-level categories) that are in turn connected to 3 nodes at the third level (superordinate categories) with the root node at the top. This hierarchy was not considered in the original experimental evaluations in [168] but used as a human organizational tool (in order to facilitate the annotation process e.g., annotators navigate through the three-level hierarchy to arrive at a specific scene type (e.g. ‘bedroom’) by making relatively easy choices (e.g. ‘indoor’ versus ‘outdoor’ at the higher level)).

Our goal is to train a classifier that can identify images as belonging to one of the 15 basic-level categories.[3] We use the images from all the subordinate categories in a basic-level category

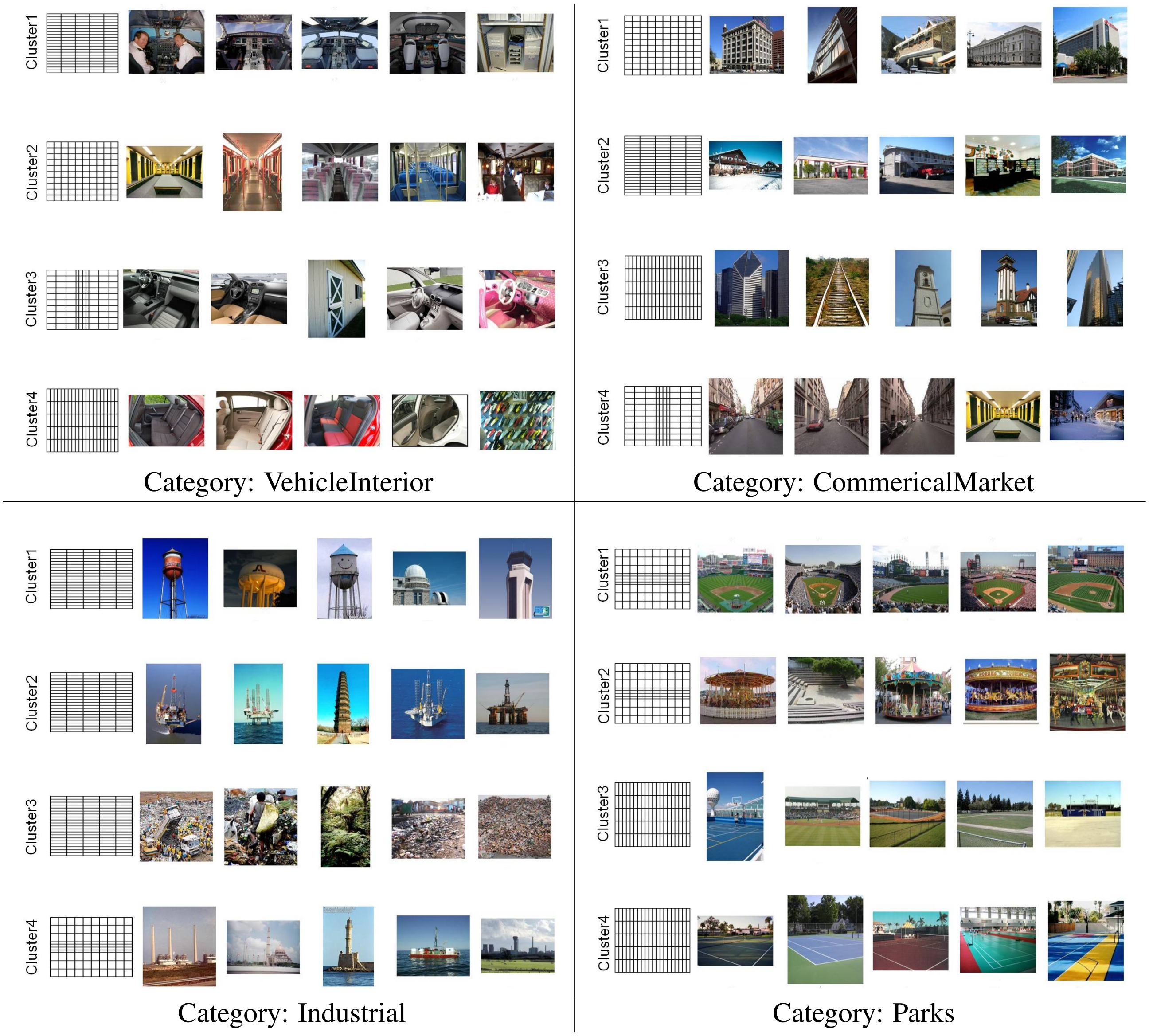
[3] The 15 basic-level categories are ‘shoppingNdining’, ‘workplace’, ‘homeNhotel’, ‘vehicleInterior’, ‘sportsNleisure’, ‘cultural’, ‘waterNsnow’, ‘mountainsNdesert’, ‘forestNfield’, ‘transportation’, ‘historicalPlace’, ‘parks’,

to build the data corresponding to that basic-level category. The data was split into half training and half testing. The classifiers are all trained in a 'one-vs-all' fashion where instances belonging to a specific category are considered positive examples, and the rest of the instances (belong to all the other categories except the chosen one) serve as negative examples in the training process. While training subcategory classifiers, instances belonging to a particular subcategory (within a category) are treated as positives and the rest of the instances belong to that category are ignored (treated as *don't care* examples). The number of subcategories K was set to be 50 (to tackle the larger intra–category diversity in this dataset). We evaluate performance using the A.P. metric as used in PASCAL VOC image classification task [39].

	linearSVM	RBF-SVM	Ours	Ours+adapt	gtruth	gtruth+adapt
shoppingNdining	22.4	44.6	33.6	**34.2**	33.1	32.8
workplace	13.5	24.3	19.5	**20.6**	18.3	19.7
homeNhotel	21.6	40.1	28.2	**28.3**	27.5	27.6
vehicleInt	15.9	32.0	31.7	30.7	33.8	**34.7**
sportsNleisure	12.1	28.3	21.1	**21.1**	19.7	21.0
cultural	16.2	31.4	23.3	**23.5**	22.6	23.2
waterNsnow	31.2	50.4	35.0	36.9	38.9	**39.7**
mountainsNdesert	18.9	44.5	28.9	**29.7**	27.5	29.5
forestNfield	21.8	61.1	39.5	38.9	42.0	**42.4**
transportation	11.6	24.3	20.0	19.9	**20.6**	19.9
historicalPlace	19.1	38.5	30.8	**31.6**	30.4	31.0
parks	14.1	29.2	27.7	**27.1**	24.7	25.3
industrial	9.0	28.6	22.0	**23.2**	20.5	19.4
housesNgardens	14.1	30.0	25.4	25.7	27.1	**28.3**
commercialMarkets	11.4	21.3	19.2	19.4	20.6	**21.3**
MEAN	16.9	35.2	27.1	27.4	27.2	**27.7**

Table 4.2: Results on SUNS397: The first two columns display the baseline results obtained using a single global linear and nonlinear SVM. The third column is the result obtained using our visual subcategory classifiers. The fourth column is our approach using an adaptive feature representation scheme. The fifth column is our approach when the clusters are initialized using ground-truth subordinate categories of a basic-level category. The last column is the same as the previous, but with the adaptive feature representation for each subcategory.

'industrial', 'housesNgardens', 'commercialMarkets'.



Category: VehicleInterior

Category: CommericalMarket

Category: Industrial

Category: Parks

Figure 4.13: SUN397 Scene-Image Classification Results: Scene-Image categories exhibit a large visual diversity due to significant variation in camera viewpoint and scene structure. The ‘vehicleInterior’ classifier contains separate subcategories for cockpit, bus interior, car front seat and car back seat. ‘commercialMarket’ is composed of different types of buildings, skyscrapers, and street/alley scenes. The ‘industrial’ category has water towers, oil rigs, land fills, and outdoor industrial scenes. Finally the category ‘park’ has baseball fields, carousals, outdoor tennis fields as subcategories. It is interesting to note that using a completely unsupervised approach, it is possible to discover the subcategories that mostly correspond to the human annotated fine-grained categories of the SUN397 dataset (even subtle ones such as car frontseat, car backseat).

Unsupervised subcategories are semantically interpretable

Table 4.2 presents the results obtained using our approach and the baseline methods. Our approach based on visual subcategories achieves a score of 27.1% confidently outperforming the baseline linear SVM 16.9%. The utility of our approach becomes more evident as we take a closer look at the classification results of discovered subcategories (Figure 4.13). Many of the subcategories discovered correspond to the semantic subordinate categories. For example, the basic-level category vehicleInterior contains clusters for "cockpit", "bus interior", "car front seat", and "car back seat" that all correspond to the fine-grained categories constituting this basic-level category. (Since the basic-level categories are built by merging the corresponding fine-grained categories of this dataset, we can make this comparison here.) Subsequently, this allows deeper reasoning about the image rather than simply assigning the category label. For example, instead of simply classifying an image as vehicleInterior, we could now say that it is a cockpit image.

Unsupervised subcategories alleviate the need for human supervision

Given the above result, we seek to quantitatively analyze the benefit of gathering human-annotated subordinate categories over the unsupervisedly discovered visual subcategories. To this end, we ran an experiment where the subcategories in our framework are initialized using the ground-truth subordinate categories. More specifically, instead of initializing the subcategories using Kmeans based initialization in our latent SVM framework, we initialize them using the ground-truth labels provided in the SUN dataset. The mean A.P. obtained using this initialization is 27.2%[4]. Note that this result is very similar to the one obtained using our unsupervised subcategories of 27.1% (see table 4.2). This is interesting because it indicates that human supervision for creating the fine-grained subcategories to train a basic-level category classifier may not be of great benefit compared to the unsupervised visual subcategories. Our observations here are also supported by the recent findings in [29], wherein semantic similarity was found to be correlated to visual similiarity at the bottom of the ImageNet [27] hierarchy, i.e., when the basic-level category is sliced into extremely small subsets. However to acquire these fine-grained subcategories, one needs to expend significant amount of human annotation effort.

[4] We also ran an experiment where the subcategory classifiers were trained using the ground-truth initialization but without the latent SVM reclustering. The result obtained using this setup was 1.1% lower, indicating that the latent reclustering actually helps over directly using the human ground-truth.

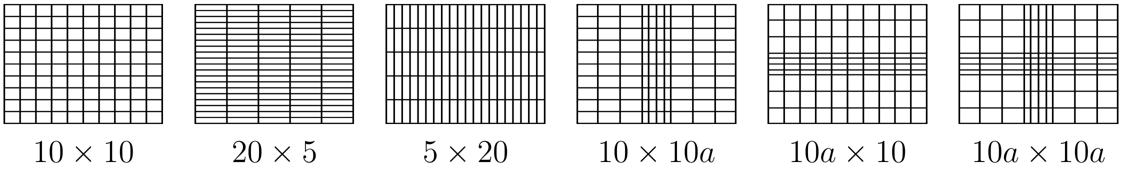


Figure 4.14: Different GIST feature representations used in the image classification experiments. Each subcategory chooses the feature representation that results in the best possible discriminative classifier.

4.6.3 Adapting Feature Representation

As studied in Section 4.5, an important benefit of the subcategory-based approach is that it offers the flexibility of choosing different feature representations and models for different subsets of data. In this section, we will present one possible approach to adapt the feature representation per subcategory for scenes.

Adapting the feature representation for each scene subcategory is crucial to accommodate the significant diversity in scene structure of the different visual subcategories [137]. Rather than using a fixed $m \times n$ uniform grid feature representation (in our experiments, the baseline system uses 10×10), we let each subcategory choose the best representation for itself from a dictionary of feature representations. We varied the representation by not only varying m, n but also by using a non-uniform griding. The following representations were considered: 5×20, 20×5, $10a \times 10$, $10 \times 10a$, $10a \times 10a$ (Figure 4.14). The initial clustering is performed using a 10×10 representation. Subsequently, the feature representation is adapted to the cluster by training separate classifiers using each of the above feature representations and picking the one that performs well on a validation dataset.

Table 4.2 reports results obtained using our approach with an adaptive feature representation. Figure 4.13 displays the different subcategories and their corresponding feature representation chosen. We observe an intuitive relationship between the subcategory and its chosen feature representation. For example, in case of the ‘alley’ scene, the classifier chooses a griding that focuses on the vertical gradients compared to a regular griding that treats all grids equivalently. This adaptation not only results in cleaner subcategories, but also improves classification results.

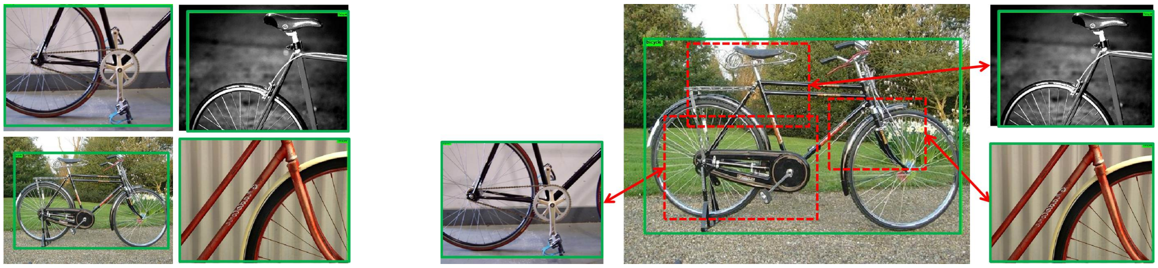
4.7 Conclusion

In this chapter, we have explored the use of visual subcategories for training better basic-level category models. Our results on the object detection and image classification tasks demonstrate that it is not only possible to attain a boost in performance but also gain a degree of interpretability. Furthermore, visual subcategories enable the use of simpler models by leveraging the increased alignment of instances within a subcategory.

Follow-up Work. There have been several interesting works that have been recently published, which have also shown the benefit of using unsupervised subcategories in improving categorization performance [3, 64, 173]. It is encouraging to see other researchers exploring similar ideas to those presented in this chapter in their work.

Chapter 5

Object Instance Sharing



(a) 4 “bicycles”, no correspondence

(b) 4 “bicycles” with correspondence

Figure 5.1: Four images taken directly from the PASCAL VOC dataset with human-annotated bounding boxes (green). (a) Despite each of the four instances having the same label (“bicycle”), their bounding boxes are not aligned so they cannot be used together as training data for a single classifier. (b) By bringing the instances into correspondence (red boxes), we can now use them to provide more training data to a classifier. This is especially important for heavily occluded/truncated instances, where data shortage is a big problem.

In Chapter 4, we have seen that subcategories are an effective tool to deal with the intra-category diversity problem. By partitioning the training data into smaller groups, the diversity within them is reduced leading to simpler classification problems, and thereby improved recognition performance. However this benefit comes at the cost of leaving very few training samples per subcategory i.e., the problem of data fragmentation. This causes the subcategory models to overfit to the training samples and thereby leads to poor generalization. In this chapter, we propose an approach to alleviate this problem by enabling the training data to be reused multiple

[1] Parts of this work have been described in Divvala et al. [31].

times. Unlike most contemporary object detection approaches that assume each object instance in the training data to be uniquely represented by a single bounding box, we allow an object instance to be described by multiple bounding boxes in this work. The additional bounding box annotations are determined based on the alignment of an object instance with the other training instances in the dataset. They act as extra training data for learning the subcategory models.

5.1 Data Fragmentation Problem with Subcategories

Recall the procedure for training a sliding-window object detector. The standard approach is to first turn each human-labeled bounding box into a feature vector using some feature descriptor, e.g., HOG, and then train a classifier, e.g., SVM, on a stack of these feature vectors to discriminate them from the rest of the visual world. We have seen that this is a reasonable strategy for older datasets, such as “INRIA person”, where object instances are largely in correspondence, i.e. aligned such that each feature vector dimension has the same visual meaning for all object instances. However, modern datasets, such as PASCAL VOC [117], are much less restricted and do not guarantee good correspondence, with often huge variations between annotated bounding box instances, as can be seen on Figure 5.1(a).

As discussed in the previous chapter, modern approaches tackle the intra-category diversity problem by using mixture models (or subcategories). Applying a mixture model to the four images in Figure 5.1(a) would likely result in each being assigned to a separate subcategory and trained with others of its kind. While reasonable, this assumes that a lot of training data is available for each subcategory. But this is often not the case, especially for occluded/truncated instances (to paraphrase Tolstoy: all good instances look the same, each occluded instance is occluded in a different way).

What we propose in this chapter is the idea of *training data reuse*. Conceptually, we would like to allow different object subcategories to be able to share (subregions of) each others training instances by providing *extra* correspondences between instances that were not part of the original human-supplied bounding box annotations, as shown in Figure 5.1(b). We operationalize this by two complementary operations: bounding box shrinking, which aims to find subregions of an instance that could be shared; and bounding box enlarging, which aims to create new subcategories by enlarging instances to include their local context. We show that these operations create more training data for each subcategory, and thus improve object detection performance, especially for occluded/truncated instances.

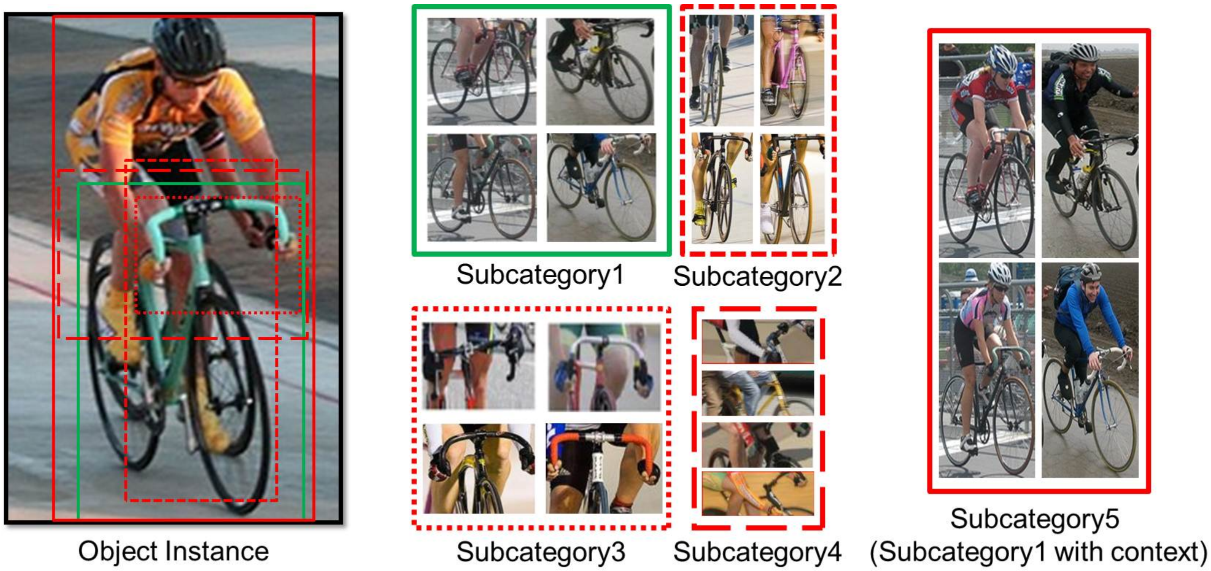


Figure 5.2: (left) A bicycle instance with its ground-truth bounding box shown in solid green. (center) Four (of the 25) subcategories discovered by our approach (few sample instances within each subcategory are shown). We allow the bicycle instance to be used multiple times with different bounding box representation for training the subcategory models. The different bounding box extents used per subcategory model are color coded accordingly e.g., subcategory3's match is shown using red dotted box, subcategory4's match shown in red dashed box, etc. (right) Subcategory1 shown after adaptively enlarging the bounding box to include local contextual cues around it.

Overview

Consider the image shown in Figure 5.2(left). The human-labeled "bicycle" bounding box is indicated by the solid green box. Given this ground-truth framing for the object instance, it is most similar to instances in the "45°-view bicycle" subcategory, so, in a standard mixture-model detector, it would be assigned to subcategory1. However, by relaxing the bounding box framing for this instance, subregions of it can also match to the other subcategory models (subcategory2, subcategory3, subcategory4) as shown using the red bounding boxes. Furthermore, looking *outside* the bounding box might also allow us to capture consistencies in the local context surrounding the object, discovering new subcategories such as "person riding a bicycle" (subcategory5). This observation suggests that by relaxing the human bounding box annotation, one can allow each training instance to be reused multiple times, with different bounding box extents. The new bounding boxes can either be cropped or enlarged versions of the original annotation depending on the alignment with the other training instances.

Of course, the idea of treating human-labeled bounding box annotations as something less than "ground-truth" is not new in object detection. In fact the very criterion for evaluating detection performance in PASCAL VOC allows for just 50% overlap between the predicted detection and the ground-truth bounding box to account for poor alignment due to inaccuracies and arbitrariness of human annotations [40]. In [44, 161], improvement in detection was demonstrated by using latent bounding box fitting, where the human-annotated bounding box is treated as being partially *latent* i.e., the bounding box is allowed to move within a local neighborhood (down to 70% overlap). Intuitively, this can be understood as locally "wiggling" the bounding box representation such that it best *aligns* with the rest of the object instances within a category (or subcategory) as shown in Figure 5.3.

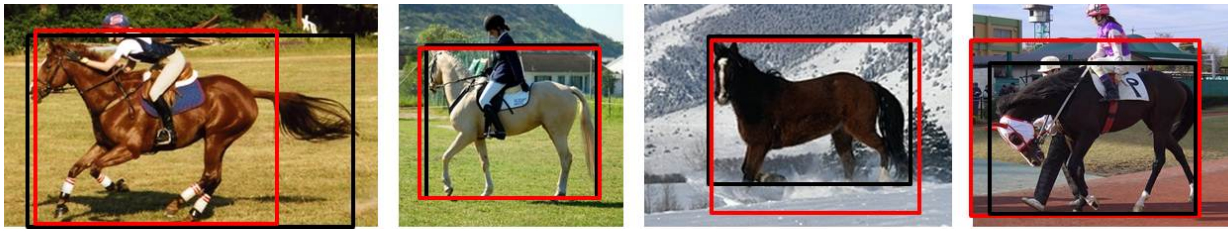


Figure 5.3: Given four instances from the profile-view horse subcategory (black bounding box indicates ground-truth), latent bounding box fitting method [44, 161] searches for the box representation (red box) that best aligns with the rest of the instances. In this case, the tail of the first horse instance is ignored, while the missing feet of the third horse are hallucinated by using an extended box.

In this work, we apply a very similar mechanism, but rather than just making local adjustments, we use it to *search* for bounding box representations that capture new correspondences between instances in the training data. The main difference is that the latent bounding box fitting assumes that each object instance is represented by a *single* bounding box belonging to a single subcategory, whereas our aim is to find *many different* bounding boxes for the same instance, so that it can be shared across multiple subcategories.

The idea of reusing object instances is particularly attractive in gathering extra data for subcategories composed of truncated instances. Truncation is a common occurrence in modern datasets where the object of interest lies partially outside the image area or is occluded e.g., the bicycle instances in Figure 5.1. Analogous to the heavy-tailed distribution of object categories [133], most training instances within a category are in canonical viewpoints and poses. Due to this lack of sufficient training samples, most detectors do not perform well on truncated instances. Our approach allows canonical non-truncated instances to be reused, providing extra data in training the subcategories corresponding to truncated instances. As shown in Figure 5.2, the impoverished bicycle handle subcategory can now use the bicycle handle from the (more common) frontal-view bicycle instance (see Figure 5.7 for more examples).

In the special case where the new bounding box is larger than the object instance, the extra spatial extent will capture information about the local context of that object. Context plays an important role in aiding object detection [35]. Information around the bounding box often provides useful contextual cues (local pixel context [26, 166]). Nonetheless most sliding-window detection approaches continue to use features computed only within the object bounding box to train the classifier. This is because the local context around the bounding box is highly multimodal for the harder PASCAL or MIT-SUN09 datasets e.g., a horse jumping over a fence appears in a different context compared to a close-up horse shot. However, this could be overcome by simply using a large number of subcategories as we will show in this chapter.

5.2 Related Work

The simplest way of reusing instances is by perturbing existing instances [44, 59, 89, 119] e.g., creating shifted, rotated, or mirrored versions. Most related is the recent work on instance-sharing [48, 94, 113, 163], with the key difference that our focus is on reusing instances within the **same** category and dataset. Also related are the works on transfer learning, where the idea is to reuse data via sharing model parameters or features [42, 56, 57, 62, 133, 174].

There have been a few recent works addressing truncation. Girshick et al., [60] proposed an extension of the deformable parts model of Felzenszwalb et al. [44]. However, their approach involves a hand-defined grammar specific for ‘person’ class. The training procedure described in [156] involves additional supervision as it requires manually extending the bounding boxes to indicate how far each truncated box ought to extend into the boundary region (and thus presents results only for two of the 20 VOC classes).

There has been a renewed interest towards incorporating local contextual cues [91, 132, 153] for training object models. In [132], detectors are trained for visual phrases that are composed of objects and the typical context surrounding them e.g., “person riding horse”, “dog lying on sofa”, etc. However their proposed method requires human-annotation of the phrases. In [91], adaptive local context models around the object of interest are separately learned and subsequently used to post-process detection results. We focus on integrating the contextual information directly into the detector, rather than post-processing the detection results.

5.3 Instance Sharing via Enhanced Bounding Box Correspondence

We begin with the latent SVM based mixture model framework introduced in Section 4.2. Consider a classification problem with a training dataset of n labeled examples $D = (< x_1, y_1 >, \ldots, < x_n, y_n >)$, with $y_i \in \{-1, 1\}$. We would like the examples to be clustered into K disjoint subcategories, and separate classifiers to be trained per subcategory. The subcategory memberships z_i are treated as latent variables. The following objective function is used:

$$\arg\min_{w} \frac{1}{2}\sum_{k=1}^{K} ||w_k||^2 + C\sum_{i=1}^{n}\epsilon_i, \tag{5.1}$$

$$y_i.s_{i,z_i} > 1 - \epsilon_i,\ \epsilon_i > 0, \tag{5.2}$$

$$z_i = \arg\max_{k} s_{i,k}, \tag{5.3}$$

$$s_{i,k} = w_k.\phi_k(x_i) + b_k, \tag{5.4}$$

where w_k denotes the separating hyperplane for the kth subcategory, $\phi_k(x_i)$ corresponds to the feature representation of an instance i in subcategory k, s_{i,z_i} indicates the score for instance i corresponding to the z_ith subcategory, and C controls the relative weight of the hinge-loss term. Equation (5.3) is referred to as the latent (discriminative) reclustering step where the latent cluster assignments z_i are iteratively estimated given the model parameters w_k learned in the previous iteration.

Calibration. Recall that the output scores $s_{i,k}$ are assumed to be calibrated appropriately for computing the $\arg\max$ in (5.3). In Section 4.2.1, we introduced a calibration step into the original LSVM formulation (5.1) as:

$$\arg\min_{w} \frac{1}{2}\sum_{k=1}^{K} ||w_k||^2 + C\sum_{i=1}^{n}\epsilon_i \tag{5.5}$$

$$y_i \cdot s_{i,z_i} \geqslant 1 - \epsilon_i,\ \epsilon_i \geqslant 0, \tag{5.6}$$

$$z_i = \arg\max_k g_{i,k}, \tag{5.7}$$

$$g_{i,k} = \frac{1}{1 + exp(A_k.s_{i,k} + B_k)}, \tag{5.8}$$

$$s_{i,k} = w_k \cdot \phi_k(x_i) + b_k, \tag{5.9}$$

$$(A_k, B_k) = \arg\min_{A_k, B_k} - \sum_{i=1}^{n} t_i \log g_{i,k} + (1 - t_i) \log(1 - g_{i,k}), \quad t_i = Or(W_{i,k}, W_i). \tag{5.10}$$

where $g_{i,k}$ is the calibrated version of the raw SVM score $s_{i,k}$, $[A_k, B_k]$ are the sigmoid parameters and $Or(.,.)$ denotes the overlap score. An alternating minimization approach is used for solving it. Given the latent assignment of object instances to subcategories z_i, detectors w_k are first trained for each subcategory. Fixing the detectors w_k, and the latent assignments z_i, the sigmoid parameters A_k, B_k are learned. Finally the detector w_k and the sigmoid A_k, B_k parameters are fixed to update the latent assignments z_i.

5.3.1 Shrinking Ground-truth Boxes

Our key insight is that it is possible to modify the latent reclustering step in a simple way so as to generate additional samples from a single training instance. The reclustering step involves (i) sliding the K subcategory detectors trained in the previous iteration on the image containing the human-annotated bounding box i and (ii) picking the highest-scoring detection window for each subcategory $W_{i,k}$ (with score $s_{i,k}$) that has at least $T\%$ overlap with the ground-truth window W_i i.e., $Or(W_{i,k}, W_i) > T$ where $Or(W_1, W_2) = \frac{|W_1 \cap W_2|}{|W_1 \cup W_2|} \in [0, 1]$ denotes the overlap score [4]. Our key modification to the reclustering step is that the formulation in (5.1) [44] keeps only *one* of the K detection windows $W_{i,k}$, namely the box W_{i,z_i} with the highest score across all subcategories, while we keep *all* the K windows as long as they pass the overlap test. With this modification, we potentially generate up to K new training samples from each training instance, each of them being aligned to one of the subcategories. We set T to a low value (10%[1], instead of 70% used in [44]) to encourage valid detections that may have low overlap over an untruncated instance. For example, in the case of the instance shown in Figure 5.2, the red dotted bounding box corresponding to the bicycle handle will be a valid detection for the bicycle handle subcategory model (subcategory3).

In order to use the multiple samples generated per training instance at each iteration (instead of a single one), we introduce a soft indicator vector $\beta_i = [\beta_{i,1}, \ldots, \beta_{i,k}, \ldots, \beta_{i,K}]$ of length $K \times 1$

[1] We empirically observed that typical truncations cover at least 10% of an unoccluded fully visible object.

into the optimization problem defined in (5.1). The value of $\beta_{i,k}$ represents the contribution of instance i towards subcategory k and is constrained to range between 0 and 1, with 0 indicating no contribution, and 1 indicating full contribution towards updating the subcategory model w_k. This is formulated as the following objective function:

$$\arg\min_{w,\beta} \frac{1}{2}\sum_{k=1}^{K} ||w_k||^2 + C_1 \sum_{i=1}^{n}\sum_{k=1}^{K} \beta_{i,k}\epsilon_{i,k} + C_2 \sum_{i=1}^{n} ||1-\beta_i||, \tag{5.11}$$

$$y_i.s_{i,k} > 1 - \epsilon_{i,k},\ \epsilon_{i,k} > 0, \tag{5.12}$$

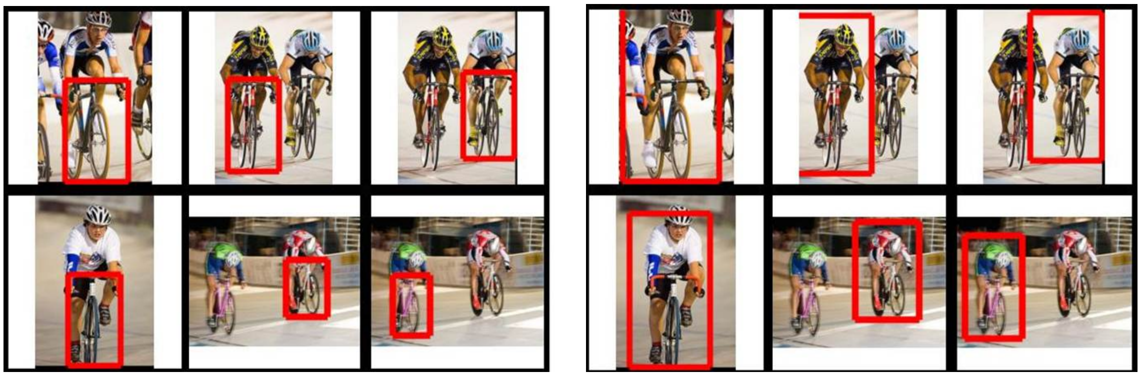
$$s_{i,k} = w_k.\phi_k(x_i) + b_k, \tag{5.13}$$

$$0 \leq \beta_{i,k} \leq 1. \tag{5.14}$$

$\beta_{i,k}$ are initialized using the solution of the previous LSVM optimization problem from (5.1): $\beta_{i,k} = 1$ if $k = z_i$ or $g_{i,k}$ otherwise i.e., all instances originally assigned to the subcategories with their ground-truth bounding box representation will have their contributions set to 1 since they will be fully used, while the samples obtained by describing an instance with new (contracted) boxes will have their β between 0 and 1. Since $g_{i,k}$ is the calibrated SVM score, its value always lies between 0 and 1. The last term in (5.11) is a regularizer over the indicator vector, which encourages each instance to be reused across multiple subcategories i.e., high regularizer signifies β_i set close to unity.

Solving the optimization problem in (5.11) for w and β jointly is a non-convex problem. We use an iterative algorithm based on the fact that solving for β given w and for w given β are convex problems. Note that setting the β's to zero for the new samples (those obtained by relaxing the human-annotation) in the above optimization problem simply returns the original LSVM solution.

Implementation details. For solving (5.11) in our experiments, we iterate only once, as it is sufficient to generate new instances once. Also for improving computation time, we threshold each β so that it will either be 0 or 1. We empirically observed that it is possible for an instance to be reused multiple times with the same bounding box extent, e.g., in the case of the bicycle category, the profile left-view as well as the right-view subcategories confidently score profile view bicycle instances (either left or right facing) with the same bounding box extent. As a result, they both *drift* towards each other. In order to avoid this drift, we apply non-maximum



Before Enlarging

After Enlarging

Figure 5.4: Inclusion of local contextual cues: Bounding boxes within each subcategory are extended (along all four sides) such that the enlarged boxes are contained entirely within the image for at least 80% of the instances. For the bicycle subcategory shown, the boxes are primarily enlarged along the vertical axis to include the “person” on top of the bicycle.

suppression[2] to the top detections across subcategories that have high overlap with each other. In order to avoid detection windows W_i^k that stride too far outside the ground-truth W_i, we supress detection windows that have high *non-overlap* score with the ground-truth $NOr(W_{i,k}, W_i) = \frac{W_{i,k} - |W_{i,k} \cap W_i|}{W_i}$.

5.3.2 Enlarging Ground-truth Boxes

As the object instances within each subcategory are tightly aligned in the appearance space, the local regions around them would also be aligned. Therefore we determine the extent of the local region to grow the box adaptively based on the image statistics containing the subcategory instances. Given the object instances within a subcategory, we determine the largest extent $\lambda = [\lambda_{x_1}, \lambda_{y_1}, \lambda_{x_2}, \lambda_{y_2}]$ to which the human-annotated bounding box can be extended (on all four sides) such that the enlarged box is contained entirely within at least 80% of the images in the subcategory. This is done by computing the distance to the image boundary along each side and picking the largest value not exceeding the extent in at least 80% of the instances (Figure 5.4). All the bounding boxes within the subcategory are grown by this margin:

$$x_1' = x_1 - \lambda_{x_1} W, \quad x_2' = x_2 + \lambda_{x_2} W \quad y_1' = y_1 - \lambda_{y_1} H, \quad y_2' = y_2 + \lambda_{y_2} H, \tag{5.15}$$

[2] Given multiple overlapping detections, they are sorted by their score and the highest scoring detection is greedily selected while skipping those that have at least 50% overlap with a previously selected detection.

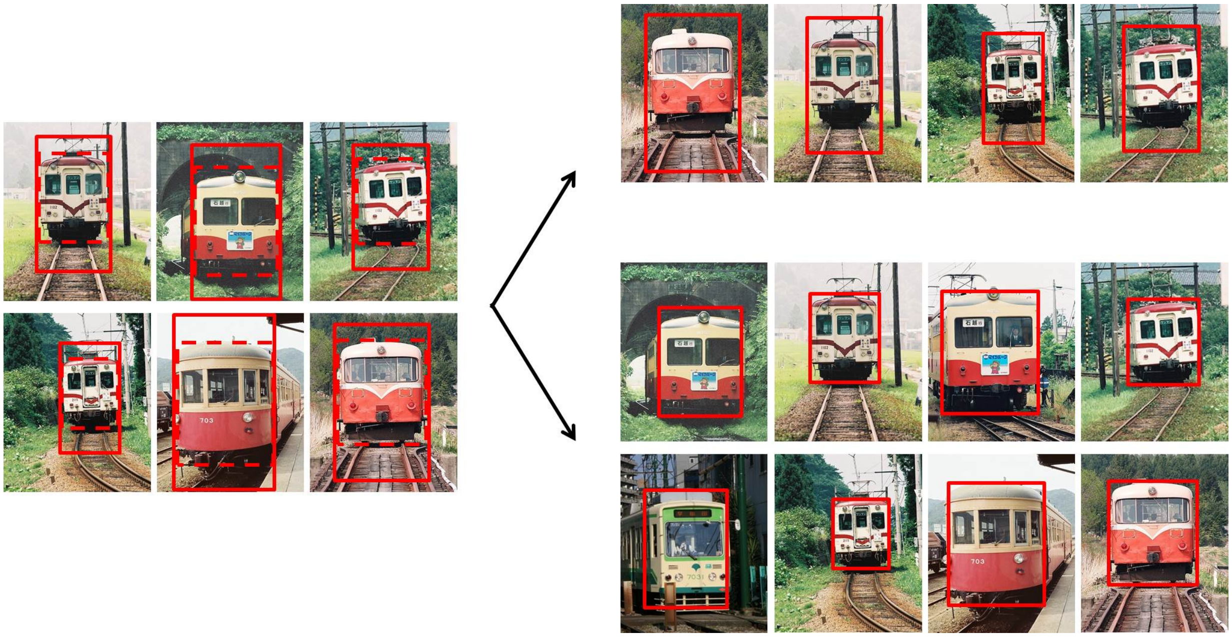


Figure 5.5: (Left) A 'train' subcategory with bounding boxes shown before enlargement (red dotted) and after enlargement (red solid). The appearance of "railway tracks" (local contextual cue) is common across most of the instances within this 'frontal view train' subcategory. (Right) After latent reclustering, the 'frontal view train' instances that have the "railway tracks" appearing in front emerge to be a separate subcategory (top), while the rest of the instances (without the "railway tracks") are assigned to a different subcategory (bottom).

where $W = x_2 - x_1$ and $H = y_2 - y_1$. Figure 5.8 displays some of the subcategories with their extended bounding boxes. We use the extended bounding boxes as training instances for learning the subcategory models as described in Eq (5.1). The model dimensions for each subcategory are also extended by a similar margin as in Eq (5.15) to account for the bounding box extension. We initialize the model using the solution of the previous (unextended bounding box) LSVM optimization function.[3] We emphasize that the latent refitting step during the reclustering process (Eq 5.3) again plays a crucial role in fixing any misalignment of the extended boxes derived from the initialization step (5.15) i.e., individual boxes can be adjusted so as to improve alignment with the rest of the instances within the subcategory (See Figure 5.5).

At testing time, we use the extended subcategory models for detecting objects in the conventional sliding-window paradigm. However, prior to evaluation, we shrink the candidate detection windows so as to comply with the evaluation protocol of having at least 50% overlap with the

[3] The central region of the extended model is initialized using the model from the previous step and the extended regions are initialized with zeros.

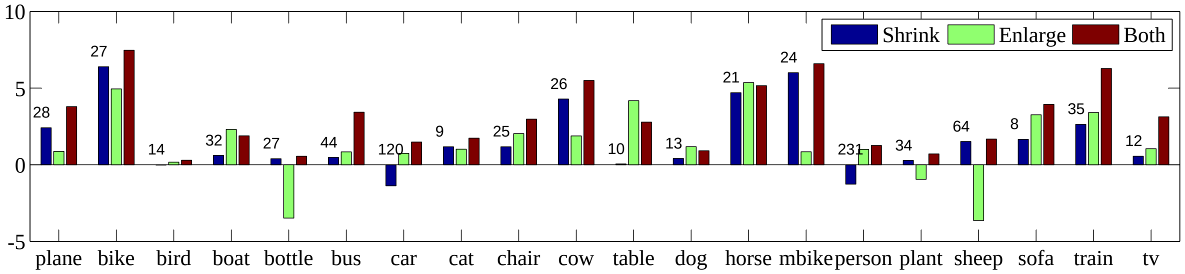


Figure 5.6: Performance improvement offered by our approach over the baseline (x-axis: 20 VOC classes, y-axis: difference in A.P.). The numbers on top of the blue bar show the percentage increase in the number of data samples (generated via relaxing the human annotation).

human-annotated (ground-truth) bounding box.

Initialization

Recall that a key step for the success of a mixture model approach is to generate a good initialization of the subcategories. In section 4.2.2, we highlighted that it is important to partition the data such that instances that are visually similar are clustered together. We follow the same approach here, where all the positive instances within a category are warped to a canonical size for extracting HOG features of fixed dimension, and then unsupervised clustering in this feature space is performed to initialize the subcategories.

5.4 Experimental Results

We evaluated the performance of our approach on the PASCAL VOC 2007 dataset [39]. We used the standard PASCAL VOC comp3 test protocol, which measures detection performance by average precision (AP) over different recall levels. As our baseline system, we use the detector initialized using appearance-based clustering as used in Section 4.3 with $K = 25$ subcategories. We use a simple root-template based model with the *deformable parts* turned off for running these experiments (for computational reasons).

Figure 5.6 compares the results obtained using our approach with respect to the baseline for the 20 PASCAL object categories. The first two bars show the improvements achieved by the shrinking and the enlarging steps respectively. The mean relative improvement (over the baseline) across 20 classes for shrinking is 6.8% (from 0.24 to 0.26), while for enlarging is 5.6%

(0.24 to 0.25). We also evaluated the result obtained by combining the final subcategory models from each and evaluating them together. The third bar displays the improvements offered by this combined system. The shrinking and enlarging ideas are complimentary to each other and combining them together offers additional boost in performance (mean relative improvement of 12.8%, from 0.24 to 0.28).

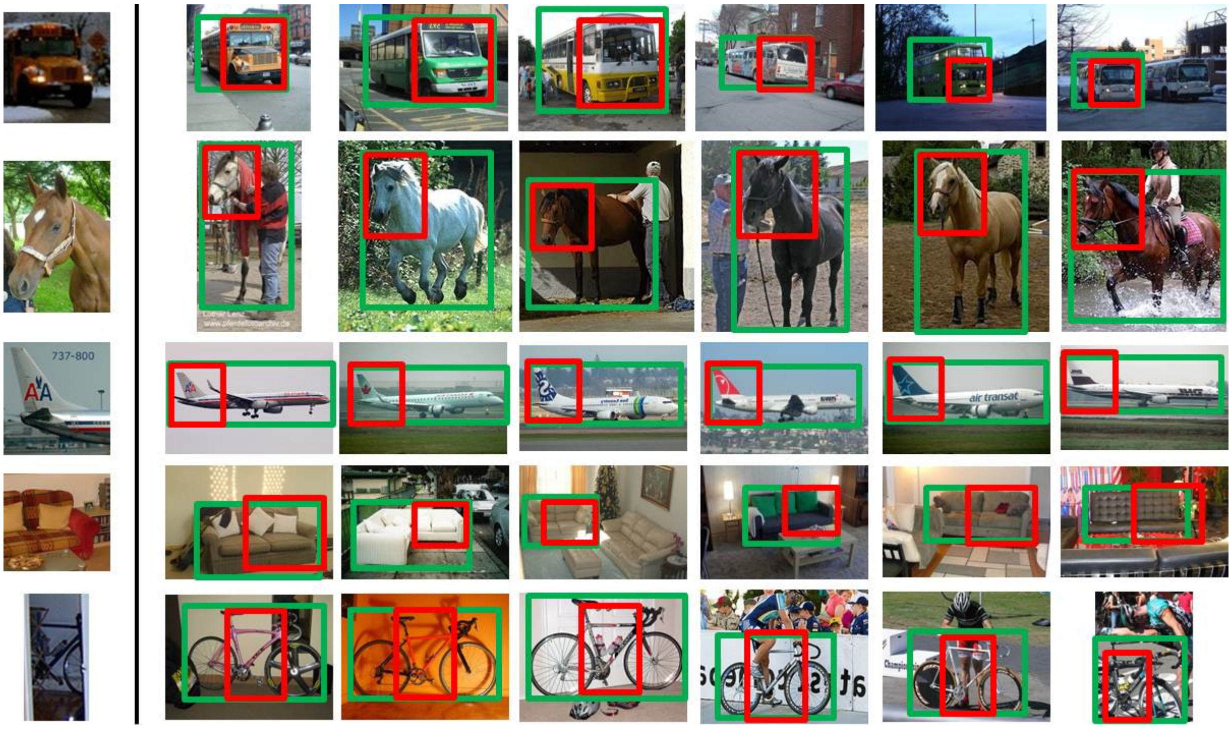


Figure 5.7: Subcategories composed of only a few instances, specifically in case of truncation, can gather more data from other training examples. Each row displays (left) a sample training instance from a subcategory, (right) new samples generated from existing training instances. Red box is the new sample, green box is the human annotation.

Effect of shrinking human annotation. As observed in Figure 5.6, our shrinking approach almost always improves the results of the baseline system, except for a marginal drop in the case of the car and person category. Atop the blue bar, we show for each class the percentage increase in the number of samples used for training the object model. On average (across the 20 classes), there is a 40% increase in the number of samples used. Figure 5.7 displays some of the qualitative results for a few impoverished subcategories. We also analyzed the performance gain specifically for detecting truncated instances. We measured the change in A.P. by exclusively evaluating the detector on the truncated instances before and after using the additional samples. We noticed a 30% relative improvement in the mean A.P. across 20 classes.

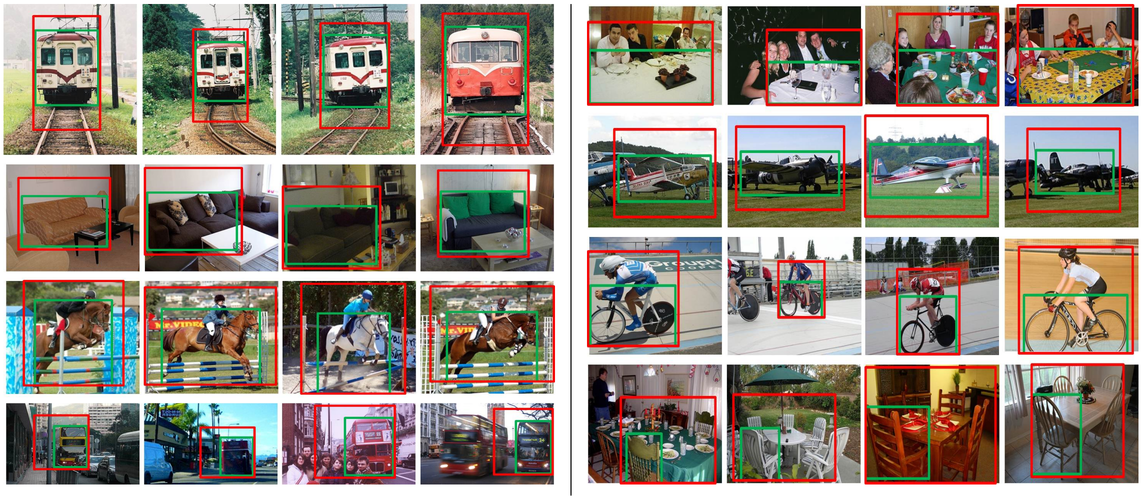


Figure 5.8: Human-annotated bounding boxes (green box) are automatically enlarged (red box) to leverage local contextual cues (adapted to the subcategory). There is a wide variation in the types of context captured per subcategory. (left) row1: rail tracks for train, row2: wall for sofa, row3: horizontal fence and vertical side bars for horse, row4: sidewalk for bus. (right) row1: people seated at dining table, row2: grass and sky for airplane, row3: person riding bicycle, row4: dining table around a chair. Notice that the local cues do not necessarily correspond to other annotated objects and could include unlabeled regions e.g., rail tracks for train.

Effect of enlarging human annotation. As observed in Figure 5.6, our adaptive enlarging scheme improves performance for all classes except bottle, plant and sheep. Bottles and plants are objects that can typically appear in varied contexts and thus the local context around them can be misleading [35]. Figure 5.8 displays some qualitative results for a few subcategories. Observe that different subcategories capture different types of local context. For e.g., in case of horse jumping over a fence, the fence and the vertical bars act as discriminative cues in improving the detection of that subcategory. This context would not be valid for a close-up horse-face subcategory. Thus a monolithic category-based detector would not be able to benefit from local context by simply enlarging the bounding box. Figure 5.9 shows the top detections retrieved for a bicycle subcategory before and after bounding box enlargement. Enlargement of boxes leads to fewer false positives amongst the top confident detections.

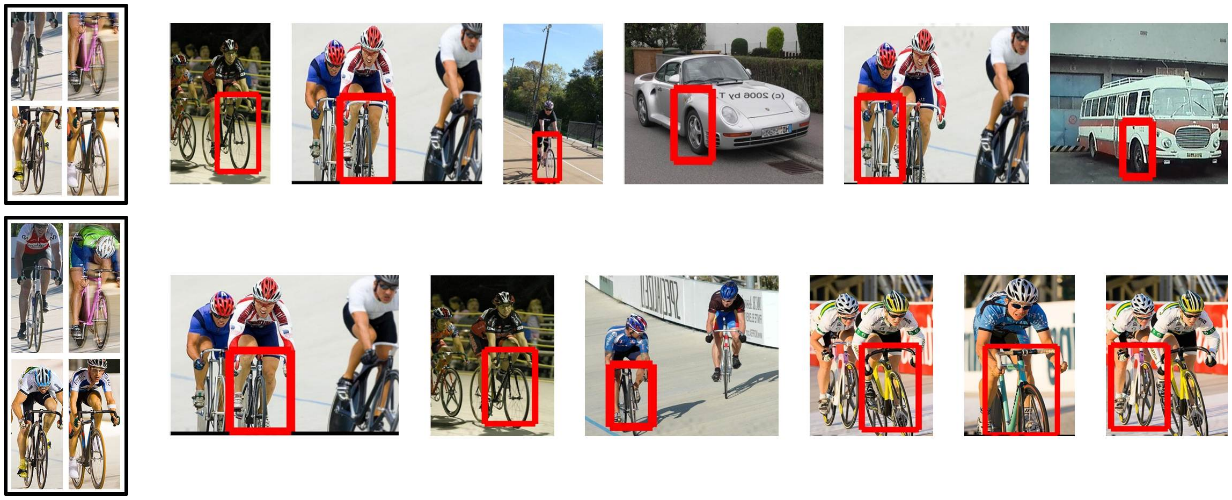


Figure 5.9: Detection performance improves with the inclusion of local context. Top row shows result obtained before enlarging ground-truth boxes, while bottom row shows result after enlargement for one of the bicycle subcategories. In each row, a few training instances for the subcategory are displayed in the left most column, followed by its top 6 detections in the test set. (In the top row, the fourth and the sixth detections are false positives.) Enlargement of boxes leads to fewer false positives amongst the top confident detections.

5.5 Towards Leveraging Unlabeled Data

Until now we have explored ways to reuse existing labeled training data for addressing the data fragmentation problem. However, semi-supervised methods such as [172] have been suggested in the past to take advantage of billions of unlabeled images freely available on the web (Facebook, Flickr) and to mitigate the need for large sets of labeled data. In this section, we describe preliminary experiments in alleviating the data fragmentation problem with subcategories.

Our approach is as follows: Given a few labeled samples within each subcategory, an initial classifier is learned. This classifier is then applied on the unlabeled images. The top detections are considered as potential candidates to be included into the training set. The classifier is retrained and the process is repeated. This procedure is referred as *self-training*, since the classifier uses its own predictions to teach itself [172].

We ran our experiments on the PASCAL VOC "train" object category. We first generated the visual subcategory detectors as described in Section 4.2. We use the 6.5 million Flickr images from [70] and retrieve about N_u (=50000) images from this collection using a keyword search for "train". We then run each of the visual subcategory detectors on all the N_u unlabeled images and retrieve the top M_i detections (above a given threshold) for the ith subcategory. We use the M_i

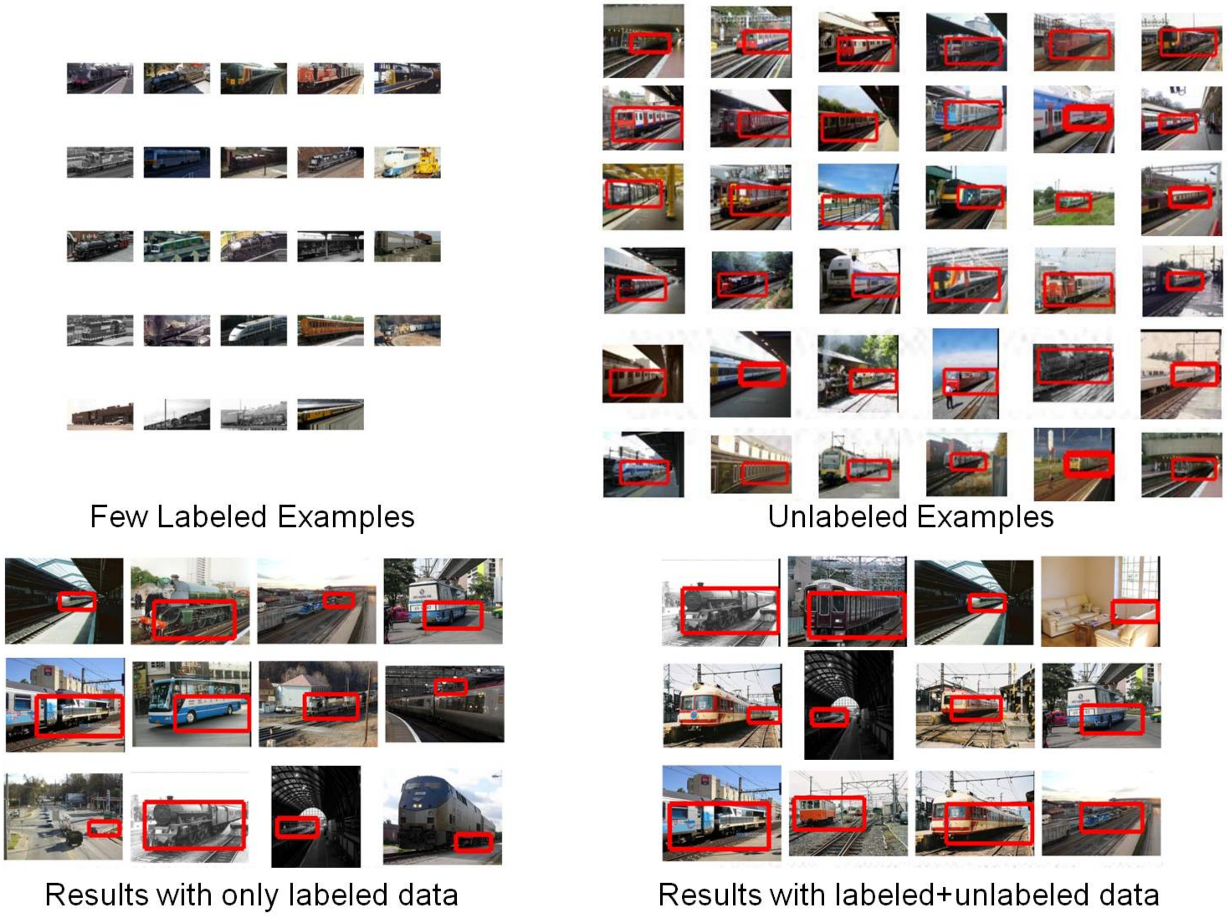
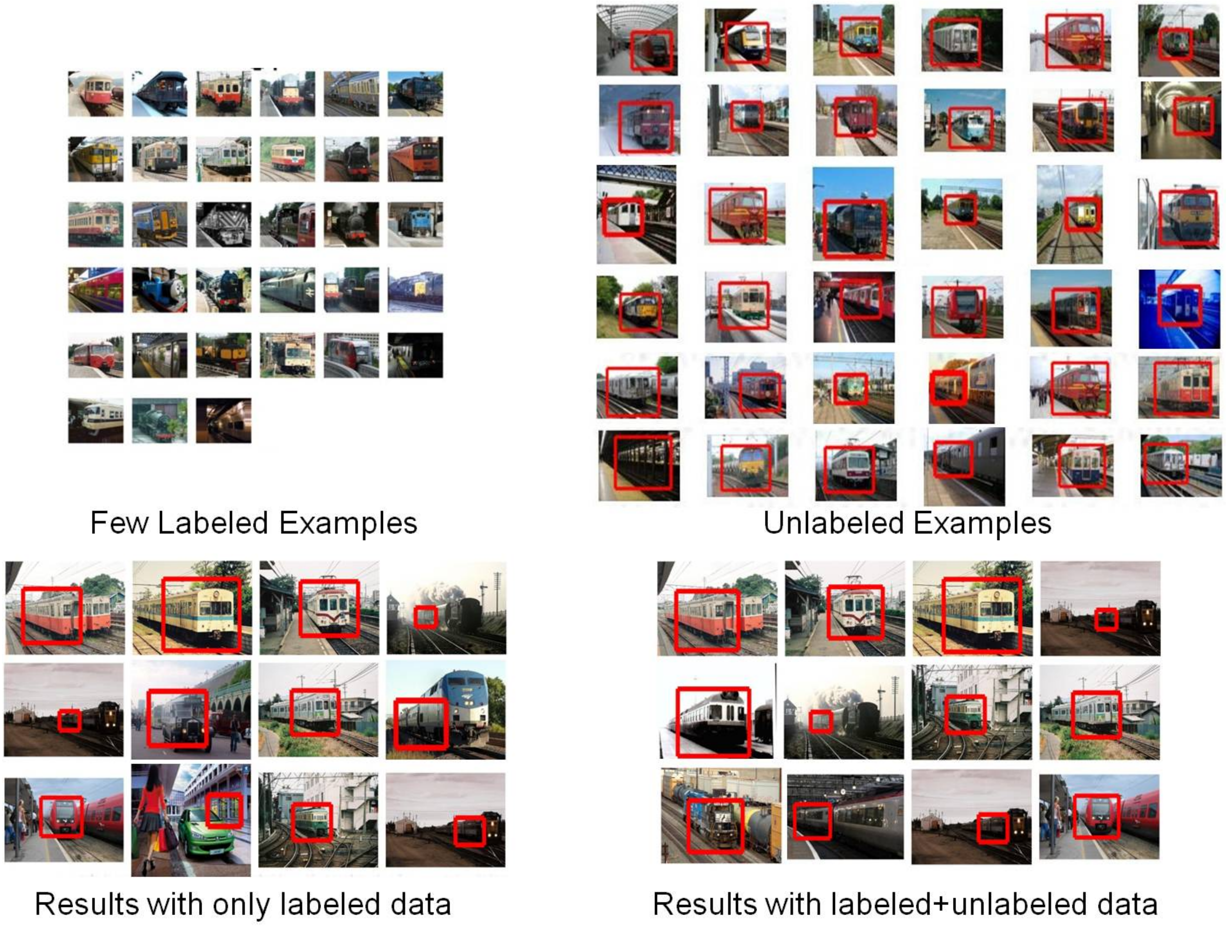

Few Labeled Examples
Unlabeled Examples
Results with only labeled data
Results with labeled+unlabeled data

Figure 5.10: Unlabeled data helps in populating the subcategories and improving their performance. We display the labeled samples belonging to a subcategory (top-left), the unlabeled samples retrieved by running the corresponding subcategory detector on Flickr images (top-right), top detection results by using the subcategory detector trained only on labeled data (bottom-left), top detection results using detector trained from labeled+unlabeled data (bottom-right).

detections for retraining the subcategory detector. In Figure 5.10 and 5.11 we display the pool of top M_i detections retrieved by running the initial subcategory detectors on the unlabeled images. Notice that since our subcategory detectors operate in the low recall-high precision regime, its top detections are often correct and help in gathering more instances of the given subcategory. Quantitatively, we observed a 2% improvement in the A.P. (over the supervised classifier baseline) by augmenting the unlabeled samples and retraining the subcategory detectors.

Future Extensions. These preliminary results illustrate the difficulty of designing approaches



Few Labeled Examples

Unlabeled Examples

Results with only labeled data

Results with labeled+unlabeled data

Figure 5.11: Unlabeled data helps in populating the subcategories and improving their performance. We display the labeled samples belonging to a subcategory (top-left), the unlabeled samples retrieved by running the corresponding subcategory detector on Flickr images (top-right), top detection results by using the subcategory detector trained only on labeled data (bottom-left), top detection results using detector trained from labeled+unlabeled data (bottom-right).

that truly benefit from weakly-labeled data. In particular, they emphasize two key issues that need to be explored in this direction: (i) While our initial experiments are based on a simple self-learning based algorithm, it is important to investigate other advanced semi-supervised frameworks for leveraging unlabeled image collections. (ii) In our experiments, we noticed that the inferred labels of the unlabeled samples do not change over the learning iterations. This probably is because the top detections added from unlabeled data were all far away from the existing decision boundary and thus resulted in no update to the classifier. Simply increasing the pool of unlabeled samples is not a good option since it might result in adding many noisy instances and will in turn result in the drifting of the classifiers away from the training data distribution. To tackle this problem, an independent “verifier” process can be employed that filters the unlabeled candidate hypothesis to be added to the training pool [126]. (iii) Finally, an interesting avenue to explore is using the new set of samples discovered from unlabeled images to recluster the data and regenerate the visual subcategories. As more data is obtained, reclustering might aid in producing more homogeneous subcategories. Further, with the availability of more data, it is also possible to increase the number of subcategories used.

5.6 Conclusion

Current detection approaches assume each human-labeled bounding box uniquely describes an object instance. In this work, we have used the human-labeled bounding box as only a rough indication of object presence. We described each object instance using multiple bounding boxes based on its alignment with other instances in the dataset. Our approach helped in enriching impoverished subcategories with additional data as well as in the inclusion of local contextual cues. We list a few possible future extensions below.

Use of more powerful detectors. While the performance of the proposed approach has been evaluated using a simple root-template based model in this work, future work could consider integrating the proposed instance sharing formulation with the deformable parts model [44] or the multi-scale model presented in Section 4.4. With more powerful detectors, it is possible to better identify sub-regions of training instances that could be shared and used across multiple subcategories. This would lead to a greater performance boost to be obtained from the use of the proposed formulation.

Other Applications. While we have seen the benefit of sharing instances across multiple subcategories using our proposed formulation for the object detection task, it is also possible to apply this idea to other tasks. For example, in case of the MIT SUN image classification dataset [168], it is possible for an image to belong to two different but related subcategories (e.g., an image within the ‘forest’ category could belong to ‘forest-path’ as well as ‘forest-road’ subcategories). In this case the instance could be used for training multiple subcategory models. We did observe a marginal 0.5% improvement in results by allowing sharing of instances across scene subcategories in our preliminary investigation on the MIT SUN dataset. While in case of the detection experiments, instances were shared across subcategories with different feature representation (depending on the bounding box extent), in case of scenes, the instances were shared with the same feature representation. Future work could explore combining the feature adaptation idea explored in Section 4.6.3 with this sharing idea.

Sharing parameters vs. instances. In this work, we have considered using sub-regions of untruncated subcategory instances for populating subcategories comprising of truncated instances and training their object models. Another interesting alternative is to directly use the sub-regions of object models learned for untruncated subcategories for detecting instances of truncated subcategories [8] (instead of sharing data instances). Such an approach allows the model parameters that are learned for untruncated subcategories (where there is more data) to be used for obtaining the model parameters for truncated subcategories (where there is relatively less data). While the proposed instance sharing approach generates models for truncated subcategories in a data-driven manner i.e., based on the truncations observed within the training data, the parameter sharing approach can potentially generate models for all possible truncations of an object.

However, the model parameters obtained by considering the truncated sub-regions of the untruncated models may be different from those obtained by learning them directly from truncated data instances due to feature resolution and the learning procedure. For example, in case of the truncated bicycle handle subcategory, the root template size is 8×8 resolution and the untruncated frontal view bicycle subcategory (of which the handle is a part of) is of 11×7 resolution. Considering the sub-region corresponding to the handle within this frontal view template might yield a 3×3 resolution template for the bicycle handle. Thus the model parameters learned for capturing the appearance of the bicycle handle would be different in either case. Future work could analyze the effects of these differing resolutions on the detection performance. Finally it is also possible to explore a hybrid approach that allows sharing of both parameters as well as instances in a single formulation.

Chapter 6

Conclusion

> "I never think of the future. It comes soon enough."
>
> Albert Einstein

This thesis studied two important factors influencing the performance of the sliding window object detection approach. In the first part of the thesis, we focused on analyzing the importance of context. From our thorough empirical evaluation of the role of various sources and uses of context in a contemporary object detection task, we have demonstrated that contextual reasoning is a critical piece of the object recognition puzzle. We showed that it is possible to integrate the detector learning and context modeling steps together, which not only improves the resulting object models but also makes the learning process computationally efficient. The utility of context extends beyond object detection, and we demonstrated its use for the task of image parsing. In the second part of the thesis, we focused on evaluating the importance of subcategories. Our empirical analysis convincingly demonstrated the benefit derived from the use of unsupervised subcategories. Beyond performance gains, we showed that subcategories are attractive for their conceptual simplicity and computational tractability. We found that their use can potentially alleviate the need for deformable parts in the deformable parts model for many object categories and applications. We also presented a simple approach for sharing training instances across subcategories so as to alleviate the data fragmentation problem.

Object recognition is one of the critical pieces of the ultimate scene understanding problem, which encompasses many disparate challenges that may interact with object recognition, such as material and texture classification, object segmentation, object tracking and trajectory prediction, etc. While exploring such possible interactions is the ultimate goal, this thesis aimed to see how far a clear focus on the problem of object recognition would take us. There is undoubtedly a rich set of future work concerning the improvement of object recognition algorithms. A few areas of future investigation are highlighted below.

Weaker Supervision. Most current detection approaches assume the availability of a clean set of labeled images with bounding boxes for each object category. While this assumption is not very strong, it must be noted that gathering images for every object category from the web and supplying bounding box annotations is a cumbersome process [40, 146]. This would definitely be an issue of concern for scaling the existing paradigm to deal with thousands of categories. To circumvent this issue, new methods are desired that can learn object models directly from a set of web images with or without any image level labels. In the former case (having image level labels), the task corresponds to the problem of weakly supervised object localization [114], while in the case of the latter, the task corresponds to the much harder object discovery problem [152]. While it is encouraging to see recent efforts along these directions, the detection accuracies in these scenarios are abysmal, leaving great scope for improvement. Finally, another interesting alternative is to explore the role of active learning [158].

Larger Scale Analysis. Is it practical to extend the current recognition pipeline to thousands of object categories? Several current design choices within the sliding window detection pipeline that are catered towards dealing with the 20 PASCAL VOC classes setting might need to be re-analyzed and reworked in order to achieve scalability to a larger number of categories. While there has been recent work towards dealing with large number of classes for the task of image classification [28, 168], there has not been much progress in the case of object detection. The task of object detection is more involved than image classification due to the practical issues of running the sliding window setup (apart from the annotation issue discussed above). One possible solution to deal with this problem is to leverage ideas from hashing [24, 77]. Hashing based techniques can allow efficient ways of testing object templates on an input image, instead of explicitly convolving them at run-time.

Deeper Understanding. While drawing bounding boxes around objects of interest is the current practice in object detection, we would like the detection algorithms to provide a deeper interpretation of the image. For example, rather than simply labeling a bounding box as “horse”, it may be more useful to label it as being a “front-facing horse” or a “horse jumping over a fence”, or describe it using any other additional information such as part locations, segmentations, etc.

The subcategory based approach studied in this thesis can offer such a benefit, as it is possible to transfer any meta data associated with the subcategory onto its corresponding detection, which is not possible in the case of a monolithic category based approach. Other recent work that focused along this direction include the exemplar SVM approach [100], and the attribute transfer approach [17, 41, 88]. Another related area of research is the topic of fine grained visual categorization [1]. Unlike the subcategory approach that implicitly discovers the subtle distinctions amongst the instances within a category (e.g., “car”), the fine grained approach aims to explicitly capture the subtle visual distinctions between similar looking categories (e.g., “Ferrari”, “Porsche”). Future work could explore other ways of devising detection algorithms that can offer richer interpretation of the objects in an image.

Better Feature Representations. Throughout this thesis we have relied on the use of the Histogram of Oriented Gradients (HOG) feature representation [26]. While HOG works very well for many man-made object categories (e.g., car, bicycle), it fails to perform for natural object categories (e.g., sheep, cat, bird etc). One possible solution to address this limitation is the use of additional feature representations such as Bag Of visual Words (BOW) [157], which encodes texture like cues (complementary to the shape like cues encoded by HOG). Nonetheless, HOG has remained the most popular and exclusive choice due to its computational efficiency. There have been a few recent works that have advocated new *higher-order* representations based on unsupervised feature learning [7, 90]. Future works could investigate integrating such features into the sliding window paradigm.

Bibliography

[1] Fine grained visual categorization workshop. In *CVPR*, 2011. 131

[2] Inria datasets. http://lear.inrialpes.fr/data. 77

[3] Omid Aghazadeh, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Mixture component identification and learning for visual recognition. In *ECCV*, 2012. 109

[4] Bogdan Alexe, Viviana Petrescu, and Vittorio Ferrari. Exploiting spatial overlap to efficiently compute appearance distances between image windows. In *NIPS*, 2011. 85, 117

[5] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *NIPS*, 2003. 82, 99

[6] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. In *PAMI*, 2011. 18, 99

[7] A. B. Ashraf, S. Lucey, and T. Chen. Re-interpreting the application of gabor filters as a manipulation of the margin in linear support vector machines. In *PAMI*, 2010. 131

[8] Y. Aytar and A. Zisserman. Enhancing exemplar svms using part level transfer regularization. In *BMVC*, 2011. 128

[9] Aharon Bar-Hillel and Daphna Weinshall. Subordinate class recognition using relational object models. In *NIPS*, 2006. 81

[10] Adrian Barbu and Song-Chun Zhu. Generalizing swendsen-wang to sampling arbitrary posterior probabilities. *PAMI*, 2005. 93

[11] Juan Bekios-Calfa, Jose M. Buenaposada, and Luis Baumela. Revisiting linear discriminant techniques in gender recognition. In *PAMI*, 2011. 102

[12] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. On manifold regularization. In *AISTAT*, 2005. 65

[13] T.L. Berg, A.C. Berg, J. Edwards, M. Maire, R. White, Yee-Whye Teh, E. Learned-Miller, and D.A. Forsyth. Names and faces in the news. In *Proc. CVPR*, 2004. 37

[14] Irving Biederman. On the semantics of a glance at a scene. In M. Kubovy and J. R. Pomerantz, editors, *Perceptual Organization*, chapter 8. Lawrence Erlbaum, 1981. 36, 38, 39, 40

[15] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, 1998. 72

[16] Lubomir Bourdev, Subhransu Maji, Thomas Brox, and Jitendra Malik. Detecting people using mutually consistent poselet activations. In *ECCV*, 2010. 78, 81

[17] Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Describing people: A poselet-based approach to attribute classification. In *ICCV*, 2011. 131

[18] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, 2001. 45

[19] Steve Branson, Catherine Wah, Florian Schroff, Boris Babenko, Peter Welinder, Pietro Perona, and Serge Belongie. Visual recognition with humans in the loop. In *ECCV*, 2010. 82

[20] Peter Carbonetto, Nando de Freitas, and Kobus Barnard. A statistical model for general contextual object recognition. In *Proc. ECCV*, 2004. 28, 35, 37

[21] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines. 2001. 102

[22] Feng Chen, Chang-Tien Lu, and Arnold P. Boedihardjo. On locally linear classification by pairwise coupling. In *ICDM*, 2008. 80, 81

[23] O. Chum and A. Zisserman. An exemplar model for learning object classes. In *CVPR*, 2007. 78, 81, 82, 90, 92

[24] O. Chum, J. Philbin, and A. Zisserman. Near duplicate image detection: min-hash and tf-idf weighting. In *BMVC*, 2008. 130

[25] Navneet Dalal. Finding people in images and videos. In *INRIA PhD Thesis*, 2006. 27

[26] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, 2005. 18, 28, 35, 37, 45, 54, 77, 78, 96, 115, 131

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale

hierarchical image database. In *CVPR*, 2009. 78, 81, 92, 107

[28] Jia Deng, Alex Berg, Kai Li, and Li Fei-Fei. What does classifying more than 10,000 image categories tell us? In *ECCV*, 2010. 130

[29] T. Deselaers and V. Ferrari. Visual and semantic similarity in imagenet. In *CVPR*, 2011. 78, 92, 101, 107

[30] Santosh Divvala, Alexei Efros, Martial Hebert, and Svetlana Lazebnik. Unsupervised patch-based context from millions of images. In *CMU Tech Report*, 2010. 59

[31] Santosh K. Divvala, Alexei A. Efros, and Martial Hebert. Object instance sharing by enhanced bounding box correspondence. In *BMVC*, 2012. 111

[32] Santosh K. Divvala, Alexei A. Efros, and Martial Hebert. How important are 'deformable parts' in the deformable parts model? In *ECCV Workshop on Parts and Attributes*, 2012. arXiv:1206.3714. 77

[33] Santosh Kumar Divvala, S. Achar, and C.V. Jawahar. Autonomous image-based exploration for mobile robot navigation. *ICRA*, 2008. 58

[34] Santosh Kumar Divvala, Alexei A. Efros, and Martial Hebert. Can similar scenes help surface layout estimation? *CVPR 2008, IEEE Workshop on Internet Vision*, June 2008. 59, 62, 66

[35] Santosh Kumar Divvala, Derek Hoiem, James Hays, Alexei A. Efros, and Martial Hebert. An empirical study of context in object detection. In *CVPR*, 2009. 35, 57, 115, 123

[36] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. In *PAMI*, 2011. 58

[37] S. Edelman. *Representation and recognition in vision*. MIT Press, 1989. 26

[38] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Proc. ICCV*, pages 1033–1038, Corfu, Greece, 1999. 62

[39] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge. http://pascallin.ecs.soton.ac.uk/challenges/VOC. 27, 36, 40, 47, 58, 87, 105, 121

[40] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. In *IJCV*, 2010. 114, 130

[41] A. Farhadi, I. Endres, and D. Hoiem. Attribute-centric recognition for cross-category

generalization. In *CVPR*, 2010. 131

[42] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Proc. CVPR*, 2005. 115

[43] L. Fei-Fei, R. Fergus, and A. Torralba. Learning and recognizing object categories. In *ICCV Short Course*, 2009. 27

[44] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 2010. 18, 20, 78, 80, 81, 82, 83, 84, 87, 90, 94, 95, 96, 98, 114, 115, 117, 127

[45] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Discriminatively trained deformable part models, release 3. http://people.cs.uchicago.edu/ pff/latent-release3/, . 97

[46] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Discriminatively trained deformable part models, release 4. http://people.cs.uchicago.edu/ pff/latent-release4/, . 17, 18, 87, 88, 90, 92, 94, 96, 97

[47] Pedro Felzenszwalb, D McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. *CVPR*, June 2008. 13, 18, 28, 35, 40, 46, 95, 96, 97

[48] R. Fergus, H. Bernal, Y.Weiss, and A. Torralba. Semantic label sharing for learning with many categories. In *ECCV*, 2010. 115

[49] Jason Forest, Maria Rifqi, and Bernadette Bouchon-Meunier. Class segmentation to improve fuzzy prototype construction: Visualization and characterization of non homogeneous classes. In *IEEE International Conference on Fuzzy Systems*, 2006. 81

[50] Dmitriy Fradkin. Clustering inside classes improves performance of linear classifiers. In *IEEE International Conference on Tools with Artificial Intelligence*, 2008. 80, 81, 102

[51] Zhouyu Fu and Antonio Robles-Kelly. On mixtures of linear svms for nonlinear classification. In *Structural, Syntactic, and Statistical Pattern Recognition*, 2008. 80, 81, 84, 102

[52] A. Gallagher and T. Chen. Estimating age, gender and identity using first name priors. In *CVPR*, 2008. 38

[53] A Gallagher, C Neustaedter, Jiebo Luo, L Cao, and Tsuhan Chen. Image annotation using personal calendars as context. In *ACM Multimedia*, 2008. 38

[54] Carolina Galleguillos and Serge Belongie. Context based object categorization: A critical

survey. *Technical Report UCSD CS2008-0928*, 2008. 38

[55] Carolina Galleguillos, Andrew Rabinovich, and Serge Belongie. Object categorization using co-ocurrence, location and appearance. In *CVPR*, 2008. 28, 35, 38

[56] Tianshi Gao and Daphne Koller. Discriminative learning of relaxed hierarchy for large-scale visual recognition. In *ICCV*, 2011. 115

[57] Tianshi Gao, Benjamin Packer, and Daphne Koller. A segmentation-aware object detection model with occlusion handling. In *CVPR*, 2011. 115

[58] Dariu M. Gavrila. Smart cars for safe pedestrians. In *Intelligent Vehicles*, 2012. 27

[59] D.M. Gavrila and J Giebel. Virtual sample generation for template-based shape matching. In *CVPR*, 2001. 115

[60] R. Girshick, P. Felzenszwalb, and D. McAllester. Object detection with grammar models. In *NIPS*, 2011. 115

[61] Stephen Gould, Rick Fulton, and Daphne Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, 2009. 15, 59, 60, 61, 65, 66, 68

[62] Gregory Griffin and Pietro Perona. Learning and using taxonomies for fast visual categorization. In *CVPR*, 2008. 115

[63] Chunhui Gu and Xiaofeng Ren. Discriminative mixture-of-templates for viewpoint classification. In *ECCV*, 2010. 78, 82, 84, 86, 92

[64] Chunhui Gu, Pablo Arbelaez, Yuanqing Lin, Kai Yu, and Jitendra Malik. Multi-component models for object detection. In *ECCV*, 2012. 109

[65] Abhinav Gupta and Larry S. Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *ECCV*, 2008. 38

[66] Isabelle Guyon and Andre Elisseeff. An introduction to variable and feature selection. In *JMLR*, 2003. 99

[67] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *ECCV*, 2012. 84

[68] Hedi Harzallah, Cordelia Schmid, Frdric Jurie, and Adrien Gaidon. Classification aided two stage localization. In *PASCAL VOC Challenge*, 2008. 81

[69] James Hays and Alexei A Efros. Scene completion using millions of photographs. *SIGGRAPH*, 26(3), 2007. 62, 63, 64

[70] James Hays and Alexei A. Efros. im2gps: estimating geographic information from a single image. *CVPR*, 2008. 38, 41, 42, 66, 124

[71] Xuming He, Richard S. Zemel, and Miguel Á. Carreira-Perpiñán. Multiscale conditional random fields for image labeling. In *CVPR*, 2004. 61

[72] Geremy Heitz and Daphne Koller. Learning spatial context: Using stuff to find things. In *Proc. ECCV*, 2008. 28, 35

[73] D. Hoiem, A. Efros, and M. Hebert. Recovering occlusion boundaries from a single image. *ICCV*, 2007. 37, 43, 45

[74] D. Hoiem, A. Efros, and M. Hebert. Recovering surface layout from an image. *IJCV*, 75 (1), 2007. 37, 41, 44, 60, 61, 63, 66, 69

[75] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Putting objects in perspective. *IJCV*, 80(1), 2008. 28, 35, 37, 38, 43, 54, 55, 56, 57

[76] Robert Jacobs, Michael Jordan, Steven Nowlan, and Geoffrey Hinton. Adaptive mixture of local experts. In *Neural Computation*, 1991. 80

[77] P. Jain, S. Vijayanarasimhan, and K. Grauman. Hashing hyperplane queries to near points with applications to large-scale active learning. In *NIPS*, 2010. 64, 130

[78] Nathalie Japkowicz. Supervised learning with unsupervised output separation. In *ICAISC*, 2002. 80, 81, 84

[79] T. Joachims. Making large-scale svm learning practical. In *Advances in Kernel Methods - Support Vector Learning, B. Schlkopf and C. Burges and A. Smola (ed.), MIT-Press*, 1999. 45, 46

[80] K. E. Johnson and A. T. Eilers. Effects of knowledge and development on subordinate level categorization. In *Cognitive Dev.*, 1998. 81

[81] K. Koh, S.-J. Kim, and S. Boyd. An interior-point method for large-scale l1-regularized logistic regression. In *Journal of Machine Learning Research*, pages 1519–1555, June 2007. 41, 42, 46

[82] B. V. K. Vijaya Kumar, Abhijit Mahalanobis, and Richard D. Juday. *Correlation Pattern Recognition*. Cambridge University Press, 2005. 27

[83] P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In *NIPS*, 2010. 99

[84] Sanjiv Kumar and Martial Hebert. A hierarchical field framework for unified context-based classification. In *Proc. ICCV*, 2005. 37

[85] Jean-François Lalonde, Derek Hoiem, Alexei A. Efros, Carsten Rother, John Winn, and Antonio Criminisi. Photo clip art. *SIGGRAPH*, 26(3), 2007. 56

[86] Jean-François Lalonde, Srinivasa G. Narasimhan, and Alexei A. Efros. What does the sky tell us about the camera? In *ECCV*, 2008. 38

[87] C. H. Lampert. Detecting objects in large image collections and videos by efficient subimage retrieval. In *ICCV*, 2009. 64

[88] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 131

[89] I. Laptev. Improvements of object detection using boosted histograms. In *BMVC*, 2006. 115

[90] Quoc V. Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S. Corrado, Jeffrey Dean, and Andrew Y. Ng. Building high-level features using large scale unsupervised learning. In *ICML*, 2012. 131

[91] C. Li, D. Parikh, and T. Chen. Extracting adaptive contextual cues from unlabeled regions. In *ICCV*, 2011. 115

[92] L.-J. Li and L. Fei-Fei. What, where and who? classifying event by scene and object recognition. In *ICCV*, 2007. 37

[93] L.-J. Li, Richard Socher, and L. Fei-Fei. Towards total scene understanding:classification, annotation and segmentation in an automatic framework. In *CVPR*, 2009. 60, 61

[94] J. J. Lim, R. Salakhutdinov, and A. Torralba. Transfer learning by borrowing examples for multiclass object detection. In *NIPS*, 2011. 115

[95] Ce Liu, Jenny Yuen, Antonio B. Torralba, Josef Sivic, and William T. Freeman. Sift flow: Dense correspondence across different scenes. In *ECCV*, 2008. 38

[96] Ce Liu, Jenny Yuen, and Antonio Torralba. Nonparametric scene parsing: Label transfer via dense scene alignment. In *CVPR*, 2009. 61, 62

[97] Jiebo Luo, M Boutell, and C Brown. Pictures are not taken in a vacuum. In *IEEE Singal Processing Magazine*, 2006. 28, 35, 38

[98] ”M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik. Using contours to detect and localize

junctions in natural images. In *Proc. CVPR*, 2008. 45

[99] Tomasz Malisiewicz and Alexei A. Efros. Improving spatial support for objects via multiple segmentations. In *BMVC*, 2007. 63

[100] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011. 82, 131

[101] A. McCallum. Efficiently inducing features of conditional random fields. In *UAI*, 2003. 99

[102] Stefano Messelodi, Carla Maria Modena, and Gianni Cattoni. Vision-based bicycle/motorcycle classification. *Pattern Recogn. Lett.*, 28(13), 2007. 52

[103] J. Mundy. Object recognition in the geometric era: a retrospective. In *J. Ponce, M. Hebert, C. Schmid, and A. Zisserman. Toward category-level object recognition*, 2006. 26

[104] D. Munoz, J. Bagnell, and M. Hebert. Stacked hierarchical labeling. In *ECCV*, 2010. 61

[105] Daniel Munoz, J. Andrew Bagnell, and Martial Hebert. On two methods for semi-supervised structured prediction. Technical Report CMU-RI-TR-10-02, Robotics Institute, Carnegie Mellon University, 2010. 60, 93

[106] Kevin Murphy, Antonio Torralba, and William T. Freeman. Using the forest to see the trees: a graphical model relating features, objects and scenes. In *Proc. NIPS*. 2003. 28, 35, 43, 46

[107] Nikhil Naikal, Allen Yang, and Shankar Sastry. Informative feature selection for object recognition via sparse pca. In *ICCV*, 2011. 99

[108] S.G. Narasimhan and S.K. Nayar. Vision and the atmosphere. In *IJCV*, 2002. 38

[109] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001. 37, 41, 101

[110] A. Oliva and A. Torralba. Building the gist of a scene: the role of global image features in recognition. *Progress in brain research*, 155:23–36, 2006. 63

[111] Aude Oliva and Antonio Torralba. The role of context in object recognition. *Trends Cogn Sci*, November 2007. 28, 35, 36

[112] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *ECCV*, 2006. 52

[113] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. In *IEEE Transactions on*

Knowledge and Data Engineering, 2010. 115

[114] Megha Pandey and Svetlana Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *ICCV*, 2011. 130

[115] Constantine Papageorgiou and Tomaso Poggio. A trainable system for object detection. In *IJCV*, 2000. 27

[116] D. Park, D. Ramanan, and C. Fowlkes. Multiresolution models for object detection. In *ECCV*, 2010. 78, 81

[117] PASCAL. The pascal object recognition database collection. Website, 2005. http://www.pascal-network.org/challenges/VOC/. 112

[118] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 2000. 55, 84

[119] D. Pomerleau. Neural network perception for mobile robot guidance. In *PhD thesis, Carnegie Mellon University*, 1992. 115

[120] J. Ponce and et al. Dataset issues in object recognition. In J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, editors, *Toward Category-Level Object Recognition*. Springer-verlag LNCS, 2006. 38

[121] J. Ponce, M. Hebert, C. Schmid, and A. Zisserman. Toward category-level object recognition. In *Springer-Verlag*, 2006. 27

[122] Srikumar Ramalingam, Pushmeet Kohli, Karteek Alahari, and Philip Torr. Exact inference in multi-label crfs with higher order cliques. In *CVPR*, 2008. 61

[123] D. Ramanan. Using segmentation to verify object hypotheses. In *CVPR*, 2007. 37, 45

[124] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *Proc. ICCV*, 2003. 60

[125] E. Rosch and B. B. Lloyd. *Principles of categorization*. Cognition and Categorization, 1978. 81

[126] Charles Rosenberg. *Semi-Supervised Training of Models for Appearance-Based Statistical Object Detection Methods*. CMU Ph.D. Thesis, 2004. 127

[127] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade. Neural network-based face detection. In *Proc. CVPR*, 1996. 27

[128] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman. Segmenting scenes by matching image composites. In *NIPS*, 2009. 62

[129] B.C. Russell, A. Torralba, C. Liu, R. Fergus, and W. T. Freeman. Object recognition by scene alignment. In *NIPS*, 2007. 28, 35, 37, 61, 62

[130] B.C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. In *IJCV*, 2007. 44, 56

[131] Bryan C. Russell, Alexei A. Efros, Josef Sivic, William T. Freeman, and Andrew Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *Proc. CVPR*, 2006. 63

[132] M.A. Sadeghi and A. Farhadi. Recognition using visual phrases. In *CVPR*, 2011. 115

[133] Ruslan Salakhutdinov, Antonio Torralba, and Josh Tenenbaum. Learning to share visual appearance for multiclass object detection. In *CVPR*, 2011. 114, 115

[134] Jason Salavon. 100 special moments. http://salavon.com/SpecialMoments/SpecialMoments.shtml. 38

[135] Walter Scheirer, Neeraj Kumar, Peter N. Belhumeur, and Terrance E. Boult. Multi-attribute spaces: Calibration for attribute fusion and similarity search. In *CVPR*, 2012. 84

[136] Henry Schneiderman and Takeo Kanade. A statistical method for 3d object detection applied to faces and cars. In *Proc. CVPR*, 2000. 81

[137] G. Sharma and F. Jurie. Learning discriminative spatial representation. In *BMVC*, 2011. 99, 108

[138] A Shashua, Y Gdalyahu, and G Hayun. Pedestrian detection for driving assistance systems: single-frame classification and system level performance. In *IEEE Intelligent Vehicles Symposium*, 2004. 81

[139] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22(8), August 2000. 86

[140] Jamie Shotton, John M. Winn, Carsten Rother, and Antonio Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006. 28, 35, 37, 60

[141] Ian Simon and Steven M. Seitz. Scene segmentation using the wisdom of crowds. In

ECCV, 2008. 38

[142] Aarti Singh, Robert Nowak, and Xiaojin Zhu. Unlabeled data: Now it helps, now it doesn't. In *NIPS*, 2008. 60

[143] Amit Singhal, Jiebo Luo, and Weiyu Zhu. Probabilistic spatial context models for scene content understanding. In *Proc. CVPR*, 2003. 38

[144] J. Sivic and A. Zisserman. Efficient visual search of videos cast as text retrieval. In *PAMI*, 2009. 64

[145] Thomas M. Strat. Employing contextual information in computer vision. In *ARPA Image Understanding Workshop*, 1993. 36

[146] H. Su, J. Deng, and L. Fei-Fei. Crowdsourcing annotations for visual object detection. In *AAAI Human Computation Workshop*, 2012. 130

[147] Kalyan Sunkavalli, Fabiano Romeiro, Wojciech Matusik, Todd Zickler, and Hanspeter Pfister. What do color changes reveal about an outdoor scene? In *CVPR*, 2008. 39

[148] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2010. 27

[149] Joseph Tighe and Svetlana Lazebnik. Superparsing: Scalable nonparametric image parsing with superpixels. In *ECCV*, 2010. 62

[150] Antonio Torralba. Contextual priming for object detection. *IJCV*, 53(2):169–191, 2003. ISSN 0920-5691. 28, 35

[151] Zhuowen Tu. Auto-context and its application to high-level vision tasks. In *CVPR*, 2008. 61

[152] T. Tuytelaars, C.H. Lampert, M. B. Blaschko, and W. Buntine. Unsupervised object discovery: A comparison. In *IJCV*, 2009. 130

[153] J.R.R. Uijlings, A.W.M. Smeulders, and R.J.H. Scha. The visual extent of an object. In *IJCV*, 2012. 115

[154] S. Ullman. *High-level vision: Object recognition and visual recognition*. MIT Press, 1996. 26

[155] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *IJCV*, 62(1–2):61–81, April 2005. 44, 45

[156] A. Vedaldi and A. Zisserman. Structured output regression for detection with partial occulsion. In *NIPS*, 2009. 115

[157] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009. 131

[158] S. Vijayanarasimhan and K. Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. In *CVPR*, 2011. 130

[159] R Vilalta. Identifying and characterizing class-clusters to explain learning performance. In *AAAI 2006 Spring Symposia: What Went Wrong and Why: Lessons from AI Research and Applications*, 2006. 104

[160] Ricardo Vilalta, Muralikrishna Achari, and Christoph Eick. Piece-wise model fitting using local data patterns. In *European Conference on Artificial Intelligence*, 2004. 82

[161] P. A. Viola, J. Platt, and C. Zhang. Multiple instance boosting for object detection. In *NIPS*, 2005. 20, 114

[162] Paul Viola and Michael J. Jones. Robust real-time face detection. *IJCV*, 57(2), 2004. ISSN 0920-5691. 27

[163] Gang Wang, David Forsyth, and Derek Hoiem. Comparative object similarity for improved recognition with few or no examples. In *CVPR*, 2010. 115

[164] Y Wang and G Mori. Hidden part models for human action recognition: Probabilistic versus max margin. In *PAMI*, 2011. 84

[165] J. Weston, S. Bengio, and N. Usunier. Large scale image annotation: Learning to rank with joint word-image embeddings. In *ECML-PKDD*, 2010. 93

[166] Lior Wolf and Stan Bileschi. A critical view of context. *IJCV*, 2006. 28, 35, 37, 115

[167] D Wolpert. Stacked generalization. In *Neural Networks*, 1992. 93

[168] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. http://people.csail.mit.edu/jxiao/SUN/hierarchy397.zip. 82, 90, 92, 100, 104, 127, 130

[169] Weilong Yang and George Toderici. Discriminative tag learning on youtube videos with latent sub-tags. In *CVPR*, 2011. 81, 84

[170] B. Yao, A. Khosla, and L. Fei-Fei. Combining randomization and discrimination for fine-grained image categorization. In *CVPR*, 2011. 82

[171] H. Zhang, J. Xiao, and L. Quan. Supervised label transfer for semantic segmentation of

street scenes. In *ECCV*, 2010. 60, 61

[172] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, University of Wisconsin Madison, 2008. 60, 124

[173] X. Zhu, C. Vondrick, D. Ramanan, and C. Fowlkes. Do we need more training data or better models for object detection? In *BMVC*, 2012. 95, 109

[174] Alon Zweig and Daphna Weinshall. Exploiting object hierarchy: Combining models from different category levels. In *ICCV*, 2007. 115