Exemplar-based Representations for Object Detection, Association and Beyond

Tomasz Malisiewicz

CMU-RI-TR-11-32

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

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

August 2011

For the vectors,

for supporting my research all these years.

Abstract

Recognizing and reasoning about the objects found in an image is one of the key problems in computer vision. This thesis is based on the idea that in order to understand a novel object, it is often not enough to recognize the object category it belongs to (i.e., answering "What is this?"). We argue that a more meaningful interpretation can be obtained by linking the input object with a similar representation in memory (i.e., asking "What is this *like*?"). In this thesis, we present a memory-based system for recognizing and interpreting objects in images by establishing *visual associations* between an input image and a large database of object exemplars. These visual associations can then be used to predict properties of the novel object which cannot be deduced solely from category membership (e.g., which way is it facing? what is its segmentation? is there a person sitting on it?).

Part I of this thesis is dedicated to exemplar representations and algorithms for creating visual associations. We propose Local Distance Functions and Exemplar-SVMs, which are trained separately for each exemplar and allow an instance-specific notion of visual similarity. We show that an ensemble of Exemplar-SVMs performs competitively to state-of-the-art on the PASCAL VOC object detection task. In Part II, we focus on the advantages of using exemplars over a purely category-based approach. Because Exemplar-SVMs show good alignment between detection windows and their associated exemplars, we show that it is possible to transfer any available exemplar meta-data (segmentation, geometric structure, 3D model, etc.) directly onto the detections, which can then be used as part of overall scene understanding. Finally, we construct a Visual Memex, a vast graph over exemplars encoding both visual as well as spatial relationships, and apply it to an object prediction task. Our results show that exemplars provide a better notion of object context than category-based approaches.

List of Figures

List of Tables

Contents

Acknowledgements

I would first like to thank my advisor, Alexei (Alyosha) Efros, for providing me with the encouragement to pursue the big problems in computer vision. Not only has he been a mentor, but he also played several other key roles in my scientific development: a friend, a critic, a visionary, a philosopher, and most importantly, a collaborator.

It would be silly to think that my dream of building intelligent machines started when I entered graduate school. Before coming to CMU, my family encouraged me to pursue my dreams, and thus none of this work would have been possible without the support of my family, especially my mother Zofia Malisiewicz and my brother Matt Malisiewicz.

I would also like to thank my fellow Robograds for making graduate school truly amazing. In particular, I would like to thank Geoff Hollinger, Boris Sofman, Jonathan Huang, Mark Palatucci, and Hanns Tappeiner for helping me realize that the 2005 Robograds were an amazing bunch. The other visionaries (Santosh Divvala, Saurabh Singh, Abhinav Shrivastava) helped make WARP hacking enjoyable and without their help and willingness to listen to my problems, those intense late-night debugging sessions would not have been possible. And finally, I would like to thank Jennifer Brown without whom my time at CMU would have been significantly less enjoyable.

Regarding funding, I would like to thank the National Science Foundation, for awarding me the NSF Graduate Research Fellowship, which helped fund my research from 2006 until 2009. The final year of my research at CMU was made possible by a research award from the MSR-CMU Center for Computational Thinking.

Chapter 1

Introduction

"Our ineptitude in getting at the record is largely caused by the artificiality of systems of indexing. When data of any sort are placed in storage, they are filed alphabetically or numerically, and information is found (when it is) by tracing it down from subclass to subclass... The human mind does not work that way. It operates by association. With one item in its grasp, it snaps instantly to the next that is suggested by the association of thoughts, in accordance with some intricate web of trails carried by the cells of the brain... Selection by association, rather than by indexing, may yet be mechanized. One cannot hope thus to equal the speed and flexibility with which the mind follows an associative trail, but it should be possible to beat the mind decisively in regard to the permanence and clarity of the items resurrected from storage. Consider a future device for individual use, which is a sort of mechanized private file and library. It needs a name, and to coin one at random, "memex" will do. A memex is a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory.

Vannevar Bush, As We May Think (1945)

Vannevar Bush's vision, as expounded in the above excerpt from his seminal 1945 essay, "As we may think" [Bush, 1945], stems from the realization that there is a great mismatch between the way in which information is typically organized and they way in which the human mind works (i.e., categorical indexing versus association). Concerned with the transmission and accessibility of scientific ideas, Bush faulted the "artificiality of systems of indexing" and proposed the **Memex**[1], a physical device which would be used to retrieve information based on associative links instead of strict categorical indexing. The associative links were to be entered manually by the user and could be of several different types. Chains of links would form into longer associative trails, creating new narratives in the concept space. Bush believed that the Memex would be invaluable to a scientist, acting as an extended smart memory which would not only store the associative trails established during routine research, but could also automatically suggest novel associations which might not be obvious to a human operator.

Motivated by the idea that "the process of tying two items together is the important thing," Bush believed the Memex would revolutionize the way we organize and retrieve information. Bush's Memex, much like Alan Turing's *Turing Machine*, was an attempt to model what it means to be a cognitive agent; however, while Turing proposed the Turing Machine in an attempt to mechanise arbitrary reasoning (algorithm execution), Bush tried to mechanise the associative processes which are analogous to using memory [Skagestad, 1996]. While the Turing Machine became the basis of modern computation, the Memex was seen decades later as pioneering hypertext and the World Wide Web. Considering the key role of the internet in modern life, it is safe to say that many of Bush's ideas are quite alive today. The way in which we find information on the internet today is more similar to Bush's Memex than the way in which a scientist used to find relevant information at the library (back in 1945).

[1]A portmanteau of **mem**ory and ind**ex**, or **mem**ory **ex**tender.

The success of the internet has dramatically changed the way we think about knowledge. In his book, “Everything is Miscellaneous: The Power of the New Digital Disorder” [Weinberger, 2007], David Weinberger expounds the view that we, as a society, should embrace digital knowledge (e.g., Wikipedia, Google) because once information is placed online, it becomes significantly more useful. Unlike books and other physical entities which must be physically organized in 3D space (e.g., on bookshelves), digital and hyper-linked knowledge is much more versatile because there is no need to explicitly assign items to a single subject/category. Like Weinberger, Clay Shirky, a prominent thinker who writes about the internet and other decentralized technologies, argues elegantly against categories in his article, “Ontology is Overrated: Categories, Links, and Tags” [Shirky, 2005]. Shirky believes in the power of two powerful building blocks which can be used to organize information: 1) links, which can point to anything, and 2) tags, which are a way of attaching labels to links. Providing multiple arguments against traditional categorization schemes, Shirky argues that “what we’re seeing when we see the Web is actually a radical break with previous categorization strategies, rather than an extension of them.” Links imply that there will generally exist multiple paths which we could traverse to find that one crucial bit of information we are looking for.

Not all branches of science have fully felt the wrath of this new digital disorder. One might argue that the problem of object recognition, as typically found in the field of computer vision and robotics, faces many problems which, using Bush’s words, are “largely caused by the artificiality of systems of indexing.” The use of categories (classes) to represent visual concepts is so prevalent in computer vision and machine learning that most researchers do not give it a second thought. Faced with a new task, one simply carves up the solution space into classes (e.g., cars, people, buildings), assigns class labels to training examples and applies one of the many popular machine learning tools to obtain a classifier.

In this work, we are advocating a different way of thinking about recognition — not as object naming, but rather as object association. The idea, suggested by evidence from cognitive science, is that the central question of recognition might not be “What is it?” but rather “What is it *like*?” [Bar, 2007]. The etymology of the very word “re-cognize” (to know again) supports the view that association plays a key role in recognition. Under this model, when faced with a novel object, the task is to associate it with the most similar objects in one’s memory. These remembered objects, in turn, provide the meta-data (e.g., object name, geometry, associated actions) needed to interpret the novel object. This allows for a dynamic definition of categories based on data availability and task (e.g., an object can be a vehicle, a car, a Volvo, or Bob’s Volvo).

An important benefit of object association over object naming is that there is no need to divide the world up into rigid, pre-defined categories *a priori*. Instead, each object instance uses its nearest neighbors (in some feature space) to infer its own identity, as general or as specific as the available data allows. For example, if our dataset does not contain many cars, then the best that we can say about a new car instance is that it can be matched to another “car”. But as the number of different cars in the dataset grows, we should be able to find very specific car matches which will allow us to recognize the same object instance as “red Honda Accord”.

1.1 Background on Categorization Theories

> “Restricting the representations derived from scenes to being conceptual amounts to imposing a severe handicap on the visual system”
>
> Shimon Edelman

To understand an image is to recognize its constituents objects where object “recognition” is usually assumed to mean *object naming* — given an image, the goal is to name the depicted objects (and possibly show the objects' spatial extent). But since our language does not have a name for every possible object instance, this requires that object *categories* be used for naming purposes. However, going from objects to object categories is an extremely noisy and lossy process: "a picture is worth a thousand words" — not one or two typically used for categorization. Before discussing the contributions of this thesis, it is worthwhile going back in time to better understand the role categories have played in vision and related disciplines.

Theories of categorization date back to the ancient Greeks. Aristotle defined categories as discrete entities characterized by a set of properties shared by all their members [Aristotle,]. His categories are mutually exclusive, and every member of a category is equal. This classical view is still the most widely accepted way of reasoning about categories and taxonomies in hard sciences.[2] However, as pointed out by Ludwig Wittgenstein, this is almost certainly not the way most of our everyday concepts work (e.g., what is the set of properties that define the concept "game" and nothing else? [Wittgenstein, 1953]). Empirical evidence for typicality (e.g., a robin is a more commonly cited example of "bird" than a chicken) and multiple category memberships (e.g., chicken is both "bird" and "food") further complicate the Aristotelian view. Wittgenstein argued that things referred to by the same name are not necessarily connected by one essential feature — instances belonging to a single concept might be connected by overlapping similarities, under the idea known as **family resemblances**.

Motivated by Wittgenstein's notion of family resemblances, the ground-breaking work of cognitive psychologist Eleanor Rosch [Rosch and Mervis, 1975, Rosch, 1978] demonstrated that humans do not cut up the world into neat categories defined by shared properties, but instead use *similarity* as the basis of categorization. Rosch's **prototype theory** postulates that an object's category is determined by its similarity to

[2]This is the reason why the debate regarding Pluto's status as a planet has been so vicious.

(a set of) prototypes which define each category, allowing for varying (graded) degree of membership. Such prototype models have been successfully used for object recognition [Basri, 1992, Edelman, 1995]. Going even further, **exemplar theory** [Medin and Schaffer, 1978, Nosofsky, 1986] rejects the need for an explicit category representation, arguing instead that a concept can be implicitly formed via all its observed instances. In exemplar theory, there is no need to form an explicit abstraction such as a prototype — categorization is performed by matching a novel input to a large set of training instances. For a great introduction to theories categorization (including prototype and exemplar theories), we refer the reader to Gregory Murphy's Big Book of Concepts [Murphy, 2002].

There is also abundant contemporary evidence in the field of cognitive science which suggests that there is more to visual understanding than can be captured by rigid categories. By studying human subjects, Aude Oliva and colleagues [Brady et al., 2009] have shown that visual memory has a massive storage capacity for object details. Their findings suggest that visual long-term representations are more detailed than previously thought, and show that "visual long-term memory representations can contain not only gist information by also details sufficient to discriminate between exemplars." [Brady et al., 2009] This suggests that humans do not simply abstract-away object details by placing them into neat and distinct categories.

Guided by intuition from theories of human visual perception and evidence from cognitive neuroscience, Moshe Bar outlines the importance of analogies, associations, and predictions in the human brain [Bar, 2009]. He argues that the goal of visual perception is not to recognize an object in the traditional sense of categorizing it (i.e., asking "What is this?"), but instead linking the input with an analogous representation in memory (i.e,. asking "What is this *like*?").[3] Once a novel input is linked with

[3]Bar uses the term "analogy" for the process of establishing a link between a previously seen object and a novel one, while we simply refer to these links as visual associations.

analogous representations, associated representations are activated rapidly and predict the representations of what is most likely to occur next. Bar suggests that *the principle of prediction* is a strong candidate for a universal principle that can explain a majority of the brain's operation [Bar, 2009]. But prediction, as argued by Bar, relies on the ability to establish meaningful analogies (or using our terminology, visual associations).

1.1.1 Problems With Visual Categories

We refer to objects by the categories they belong to so effortlessly in our daily lives that it is not clear why machines should have such a hard time performing the same task. However, it is unclear whether categorization is useful for computer vision. For example, many object categories are functional. These categories often exhibit visual polysemy — object instances that have visually nothing to do with each other (e.g., "chair"). Moreover, categories are language dependent — an object category in one language might not exist in another [Lakoff, 1987]. Yet another source of visual polysemy particular to 2D image sets is view-dependence. Taken on its own, a side-view of a car has visually nothing in common with a frontal view of a car (Figure 1.1 right). Therefore, trying hard to make a single car concept detector fire on both seems counterproductive — by learning category-specific models, we might actually be trying to solve a more difficult problem than is necessary.

In addition, visual object categorization is not even consistent across individuals. Consider, for instance, the LabelMe dataset [Russell et al., 2008] where human labelers can choose any English word/phrase they like for object annotation. Figure 1.1 (left) shows a typical example of visual synonyms — two visually similar objects that have been arbitrarily assigned different labels ("building" vs. "house").

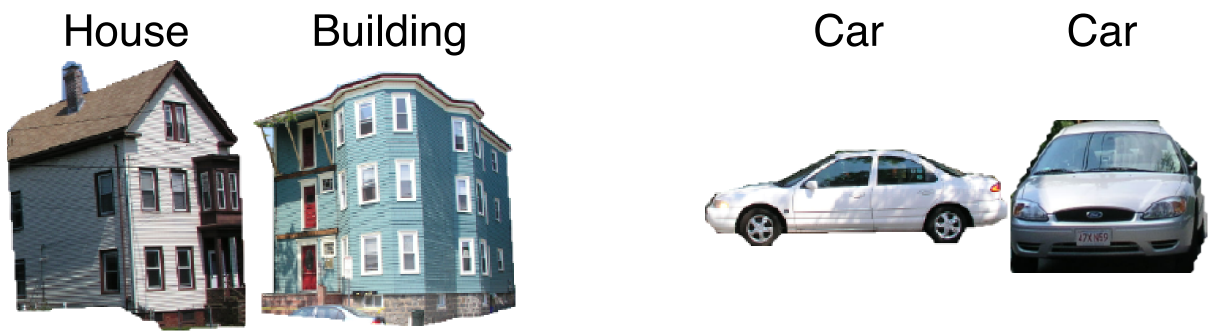


Figure 1.1: Typical examples of visual synonyms and visual polysemy that are common in LabelMe [Russell et al., 2008] annotations. Visual synonyms: two objects that are visually quite similar but have different class labels (left). Visual polysemy: two objects that have nothing in common *visually* but are labeled to be the same class (right).

1.1.2 Why Categorize?

But it might not be too productive to concentrate on the various categorization theories without considering the final aim — what do we need categories for? One argument is that categorization is a tool to facilitate knowledge transfer. E.g., having been attacked once by a tiger, it is critically important to determine if a newly observed object belongs to the tiger category so as to utilize the information from the previous encounter. Note that here recognizing the explicit category is unimportant, as long as the two tigers could be associated with each other. Another argument for using categories is they enable communication. If two agents share a common linguistic vocabulary, then they can effectively exchange information about the world.

But what if there is no single hierarchical organizational scheme that is suitable for organizing the visual world? As argued by Alon Halevy in "The Unreasonable Effectiveness of Data," [Halevy et al., 2009], a part of the world is explained well by parametric/mathematical models, and a part of the world is better explained by data. We argue that the visual world can be sufficiently explained by data, and the search for a single organizational scheme for the visual world might be futile.

1.2 Related Work

Of course, the idea of associating a new instance with something seen in the past has a long and rich history, starting with the British Empiricists [Locke, 1689, Berkeley, 1710, Hume, 1739], and continuing as exemplar theory in cognitive psychology [Medin and Schaffer, 1978, Nosofsky, 1986], case-based reasoning in AI [Schank, 1983], instance-based methods in machine learning [Aha et al., 1991], data-driven transfer in graphics [Ren et al., 2005], etc. In computer vision, this type of non-parametric technique has been quite successful at a variety of tasks including: object alignment [Belongie et al., 2002, Berg et al., 2005], scene recognition [Torralba et al., 2008, Xiao et al., 2010], image parsing [Liu et al., 2009], among others. However, for object detection, data-driven methods such as [Russell et al., 2007, Malisiewicz and Efros, 2008], have not been competitive against discriminative approaches (though the hybrid method of [Chum and Zisserman, 2007] comes close). Why is this? In our view, the primary difficulty stems from the massive amounts of negative data that must be considered in the detection problem. In image classification, where dataset sizes typically range from a few thousands to a million, using the k Nearest Neighbors (k-NN) algorithm to compute distances to all training images is still quite feasible. In object detection, however, the number of negative windows can go as high as hundreds of millions, performing k-NN using both positives and negatives is prohibitively expensive. Using heuristics, such as subsampling or ignoring the negative set, results in a substantial drop in performance.

In contrast, current state-of-the-art methods in object detection ([Dalal and Triggs, 2005], [Felzenszwalb et al., 2010], and their derivatives) are particularly well-suited for handling large amounts of negative data. They employ "data-mining" to iteratively sift through millions of negatives and find the "hard" ones which are then used to train a discriminative classifier. Because the classifier is a linear SVM, even the hard negatives do not need to be explicitly stored but are represented parametrically, in terms of a decision boundary.

However, the parametric nature of these classifiers, while a blessing for handling negative data, becomes more problematic when representing the positives. Typically, all positive examples of a given object category are represented as a whole, implicitly assuming that they are all related to each other *visually*. Unfortunately, most standard *semantic* categories (e.g., “car”, “chair”, “train”) do not form coherent *visual* categories [Malisiewicz and Efros, 2008], thus treating them parametrically results in weak and overly-generic detectors. To address this problem, a number of approaches have used semi-parametric mixture models, grouping the positives into clusters based on meta-data such as bounding box aspect ratio [Felzenszwalb et al., 2010], object scale [Park et al., 2010], object viewpoint [Gu and Ren, 2010], part labels [Bourdev et al., 2010], etc. But the low number of mixture components used in practice means that there is still considerable variation within each cluster. As a result, the alignment, or visual correspondence, between the learned model and a detected instance is too coarse to be usable for object association and label transfer. While part-based models [Felzenszwalb et al., 2010] allow different localizations of parts within distinct detections, the requirement that they must be shared across all members of a category means that these “parts” are also extremely vague and the resulting correspondences are unintuitive. In general, it might be better to think of these parts as soft, deformable sub-templates. The Poselets approach [Bourdev et al., 2010] attempts to address this problem by manually labeling parts and using them to train a set of pose-specific part detectors. While very encouraging, the heavy manual labeling burden is a big limitation of this method.

Meta-data transfer has also been used in a “recognize-then-transfer” setting (e.g., [Thomas et al., 2009, Li et al., 2011]). The approach of [Thomas et al., 2009] adds a meta-data reasoning stage to the recognition system of [Leibe et al., 2004], but since object categories are represented parametrically, a non-trivial meta-data propagation step must be included. It is also worthwhile mentioning the approach of [Savarese and Fei-Fei, 2007], which outperforms [Thomas et al., 2009] and provides a more compact representation of the object category. However both [Thomas et al., 2009] and [Savarese and Fei-Fei, 2007] are category-based and must cope with the problems inherent in dealing with visual categories (see discussion in Section 1.1.1). The approach of [Li et al., 2011] first uses a collection of pose-sensitive object detectors to produce detections in the image and then performs a landmark-based alignment using an iterative algorithm. As will be seen in Chapter 4, our approach does not require any additional alignment steps after detection, nor does it require training separate category-based object detectors to create initial detections.

1.3 Thesis Overview

Part I of this thesis is dedicated to learning exemplar representations for object category detection. Chapter 2 motivates the problem of learning exemplar-specific similarity measures and presents Local Distance Functions and how they can be used with Multiple Segmentation algorithms for localizing exemplars in images [Malisiewicz and Efros, 2008]. Chapter 3 is about Exemplar-SVMs [Malisiewicz et al., 2011], which are a marriage of the exemplar-based methodology with discriminative training found in approaches such as [Felzenszwalb et al., 2010]. Part I shows that by exploiting large amounts of visual data, we can learn exemplar-specific similarity measures, which are superior to learning-free nearest-neighbor approaches. We conclude Part I by showing that with an ensemble of Exemplar-SVMs, we can achieve performance on the PASCAL VOC detection task that is on par with the much more complex latent part-based model of [Felzenszwalb et al., 2010], at only a modest computational cost increase.

Part II is dedicated to the power of exemplar associations — we show results on tasks which go beyond object category detection. We evaluate the Exemplar-SVM approach on a diverse set of meta-data transfer tasks [Malisiewicz et al., 2011] (segmentation transfer, qualitative geometry transfer, and related object priming) as well as visualize the Exemplar-SVM's induced exemplar-exemplar similarity structure on PASCAL. We conclude with a contextual object prediction task [Malisiewicz and Efros, 2009], which requires the formation of a graph over exemplars, which we call the *Visual Memex*, and show an improvement over category-based baselines. In the Appendix, we overview an application of the Exemplar-SVM algorithm to the problem of cross-domain image matching [Shrivastava et al., 2011] (e.g., matching paintings to photos and matching sketches to photos) as well as discuss our contributions in the form of an open source Exemplar-SVM object detection library.

Part I: Learning Exemplar Representations

"Represent all the data with a nonparametric model rather than trying to summarize it with a parametric model, because with very large data sources, the data holds a lot of detail... Now go out and gather some data, and see what it can do."

Alon Halevy, Peter Norvig, and Fernando Pereira,
"The Unreasonable Effectiveness of Data" (Google, 2009)

Despite the benefits, posing recognition as exemplar association is not an easy task. One requirement is a very large dataset, rich enough to contain many different objects and many instances. Recently, with the appearance of large image collections, several systems have shown that simple k-nearest-neighbor (k-NN) approaches can often perform surprisingly well [Torralba et al., 2008, Hays and Efros, 2007, Russell et al., 2007]. However, all these methods match the image as a whole, which effectively limits them to operating on the coarse scene level (there is simply not enough data in the world to observe all possible objects in all possible configurations). To match individual objects within scenes, we must partition the image into chunks which are small enough to be matchable in a reasonably-sized database, but large enough to encode specific objects, not generic "visual words" [Sivic and Zisserman, 2003]. One approach is to allow partial scene matches instead of matching entire scenes [Russell et al., 2009]; however, since we are interested in detecting entire objects, we focus on matching object-sized image chunks (for which we use either multiple segmentations and sliding windows).

In addition to a training dataset, exemplar-based approaches require a suitable notion of visual similarity. In order to make our approach work on a moderately-sized training set, we show that learning visual similarity functions is superior to using a single hand-crafted metric, as is common in nearest-neighbor approaches. We propose two types of similarity functions, *Local Distance Functions* and *Exemplar-SVMs*, as well as algorithms for learning them from data. While the number of learning problems we solve scales linearly with the number of exemplars, per-exemplar learning problems are much easier to deal with because each exemplar's learning problem is simpler than a global category-wise problem. More importantly, per-exemplar similarity functions give us a variable and object-dependent notion of similarity, allowing us to reason about a diverse set of object types.

Chapter 2

Local Distance Functions

When we look at the world around us, we notice that similarity is defined differently for different types of objects in the world. Worse yet, objects can exhibit similarity on many different, often contradictory, levels: shape, size, color, texture, etc. For example, Adelson divides the world into "things" (such as cars, people) and "stuff" (grass, pavement, ice cream, etc.) [Adelson, 2001]. For "things", like cars, object shape is an important cue whereas object color is usually not. But for "stuff", like grass, which does not own its boundaries, shape is useless (in fact, detrimental) but color and texture are extremely important. Therefore, to find what a given object instance is similar to, it is imperative that the right distance metric for that instance be used. But, of course, to know the right distance metric requires knowing what that object is! As is often the case in vision, we are faced with a difficult chicken and egg problem. In the case of distance functions, we tackle this problem by learning a separate combination of elementary distances (such as color, texture and shape) for *each exemplar in our database*. To make things difficult, typical human object labels (which are generally the basic-level categories to which an object belongs to) are not good enough to make the assumption that an exemplar should be similar to all other exemplars with the same label (see Figure 1.1 and discussion in section 1.1.1). We instead propose a largely data-driven approach which weakly uses the object labels to automatically learn for each exemplar a distance function *and* a small set of exemplars which are **visually similar** to the exemplar.

The distance functions defined in this section are positive linear combinations of elementary distances. We use one common distance function parameterization for all exemplars, but learn instance-specific parameters by solving a different supervised learning problem for each exemplar. Each exemplar has its own distance function — we denote exemplar e's distance function as $D_e(\mathbf{x}_i) = D(\mathbf{x}_i|\mathbf{x}_e, \mathbf{w}_e)$, which takes as input some query features $\mathbf{x}_i \in \Re^F$ and returns the similarity between the exemplar, $\mathbf{x}_e \in \Re^F$ and the input $\mathbf{x}_i$ based on the notion of similarity encoded by $\mathbf{w}_e \in \Re^{F+}$.

We use a popular class of distance functions which are also called diagonal Mahalanobis distance metrics. The decision boundary of each distance function is an ellipsoid centered at $\mathbf{x}_e$, where each component of $\mathbf{w}_e$ denotes the elongation of the ellipse along that particular dimension. Since we keep each distance function anchored at the exemplar, we can think of any new point $\mathbf{x}_i$, not in its raw feature space, but in the distance-to-exemplar-e space (see Figure 2.1). Let us define $\mathbf{d}_{ei}$ to be the F dimensional non-negative "distance squared vector" between $\mathbf{x}_e$ and the input $\mathbf{x}_i$, and $\mathbf{d}_{ei}[k]$ the k-th component of this vector. We we can thus write exemplar e's distance function as follows:

$$D_e(\mathbf{x}_i) = (\mathbf{x}_e - \mathbf{x}_i)^T W_e (\mathbf{x}_e - \mathbf{x}_i) \tag{2.1}$$

$$D_e(\mathbf{x}_i) = \mathbf{w}_e^T \mathbf{d}_{ei} \tag{2.2}$$

$$\mathbf{w}_e = diag(W_e) \tag{2.3}$$

$$\mathbf{d}_{ei}[k] = (\mathbf{x}_e[k] - \mathbf{x}_i[k])^2 \tag{2.4}$$

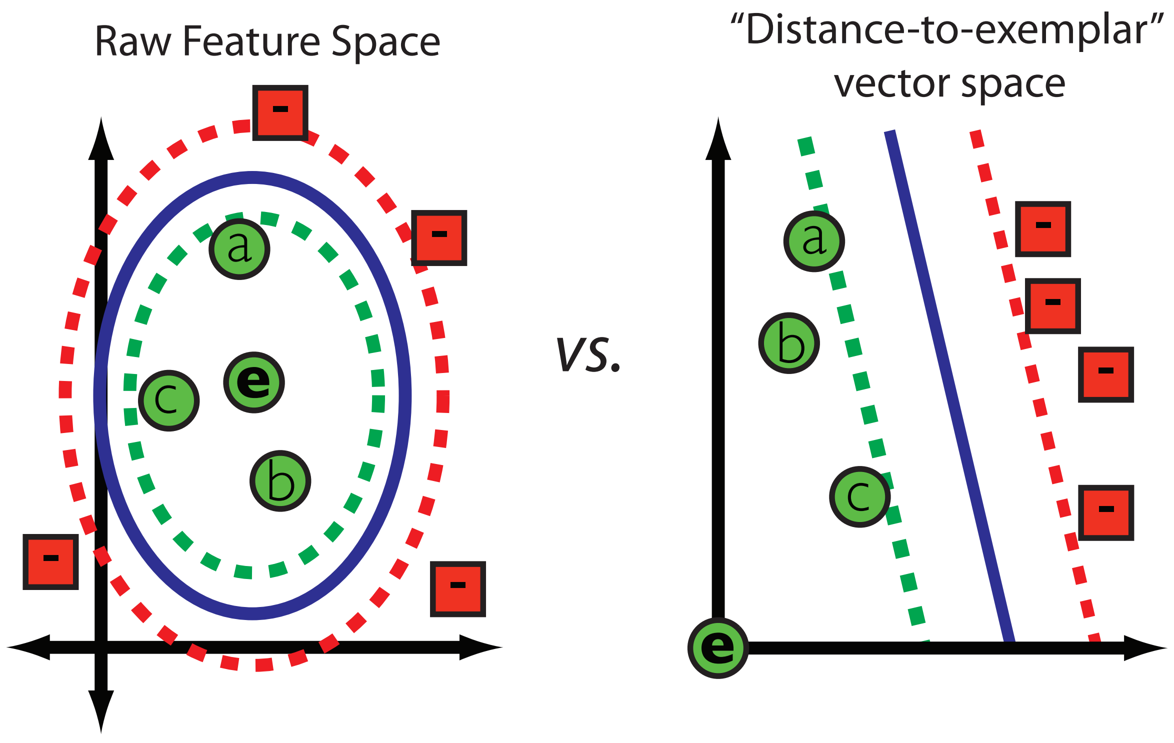


Figure 2.1: **Learning a distance-function in "distance-to-exemplar" vector space**. Instead of performing learning in the raw feature space (where the decision boundary is non-linear), we map the problem to a distance-to-exemplar space (where the decision boundary is linear). The exemplar, indicated by e, is now placed at the origin, because its own distance-to-exemplar will be $\mathbf{0}$. Any data point $\mathbf{x}_i$ can be mapped to the positive point $\mathbf{d}_{ei}$ in the distance-to-exemplar space as indicated by Equation 2.4.

Formulating the problem in a "distance-to-exemplar" vector space, we can see that an application of the distance function is simply a dot product in the new space (Equation 2.2). An immediate advantage is that it allows us to use one of the many well-understood tools geared towards learning linear decision boundaries.

2.1 Learning Distance Functions With Friends

For each exemplar we learn $\mathbf{w}_e$ by forming a supervised-learning problem from the labels associated with training examples. However, since there is a great deal of visual diversity within a single object category (as argued in Section 1.1.1), we want to avoid forcing an exemplar to be similar to all in-class instances. Instead of using all other object instances with the small class as e's positives, we let the learning process choose which examples e is actually similar to (referred to as e's *friends*). For this purpose, we use an exemplar-specific binary vector $\boldsymbol{\alpha}_e$ (whose length is equal to the number of exemplars with the same label as e), where the non-zero elements of $\boldsymbol{\alpha}_e$ indicate which examples are e's friends. Since we do not know which exemplars are visually similar to each other, we are faced with a chicken-and-egg problem and must learn $\mathbf{w}_e$ and $\boldsymbol{\alpha}_e$ jointly. Also, as is common in imbalanced classification problems, we learn a bias term b_e in addition to the orientation of the hyper-plane $\mathbf{w}_e$.

For learning, we start out with a large annotated set of object exemplars $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{y}_i \in \{1, \dots, M\}$ indicates the class that each example belongs to. We independently learn a separate distance function for each exemplar, thus our approach scales linearly with the number of exemplars N. Because for each exemplar's learning problem, we solve a two-class problem, our approach is not sensitive to the overall number of object classes, M. Given a single example, we partition all other objects into the in-class examples set, $C_e = \{j : y_j = y_e\}$, and its complement, the out-class set.

The output of per-exemplar learning is a single weight vector, a scalar bias, and the exemplar's friend indicator vector:

$$[\mathbf{w}_e^*, b_e^*, \boldsymbol{\alpha}_e^*] = \underset{\mathbf{w}, b, \boldsymbol{\alpha}}{\operatorname{argmin}}\, \Omega(\mathbf{w}, b, \boldsymbol{\alpha}) \tag{2.5}$$

Learning the distance function is performed by optimizing the following objective function for each exemplar (since we independently learn each exemplar's distance function, we drop the e subscript for clarity):

$$\Omega(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{\lambda}{2}||\mathbf{w}||^2 + \sum_{i \in C} \boldsymbol{\alpha}_i\, L(-(\mathbf{w}^T \mathbf{d}_i + b)) + \sum_{i \notin C} L(\mathbf{w}^T \mathbf{d}_i + b) - \sigma||\,\boldsymbol{\alpha}\,||^2 \tag{2.6}$$

$$
\begin{aligned}
\mathbf{w} &\geq 0 \\
\boldsymbol{\alpha} &\in \{0,1\}^{|C|}
\end{aligned}
$$

In our formulation, let $L(x)$ be a loss function, such as the hinge-loss $L_{hinge}(x) = max(1-x,0)$ or the squared hinge-loss $L_{hinge-sq}(x) = max(1-x,0)^2$, and let C be the set of exemplars that have the same label as the focal exemplar (i.e., the exemplar whose distance function we are learning). We initially used the constraint that $\sum_i \alpha_i = K$, where K is the target number of exemplars we force to be similar to e [Malisiewicz and Efros, 2008], but later found that using a term which allows each exemplar to select a variable number of friends makes more sense (this is the last term, modulated by σ) [Malisiewicz and Efros, 2009]. On one hand, by optimizing the distance function learning objective function we strive to maintain a max-margin separation between the exemplar plus its set of friends and the set of negative examples. One the other hand, we try to select the friends such that they allow for a large separation, preferring to choose a large number of friends.

Without the $\boldsymbol{\alpha}$ parameter and with no constraint on $\mathbf{w}$, Equation 2.6 is just the primal form of many convex statistical learning techniques (such as Logistic Regression and Support Vector Machines). In our case we use the convention that smaller values of $\mathbf{w}^T\mathbf{d}_i$ imply a greater degree of similarity to exemplar e (the semantics of a distance), so the positivity constraint on $\mathbf{w}$ is meant to ensure that a large elementary distance can never imply a very small (potentially negative) distance. This also ensures that the exemplar has maximal similarity with itself. Since the presence of the binary $\boldsymbol{\alpha}$'s renders the problem non-convex (as is common in latent variable models), we proceed iteratively estimating $\boldsymbol{\alpha}$ given $\mathbf{w}$ and estimating $\mathbf{w}$ given $\boldsymbol{\alpha}$. During each iteration, we are guaranteed to never increase the value of our objective function (Equation 2.6) and thus efficiently find a local minimum. We start with an initial distance function $\mathbf{w}^0$ and break down Equation 2.6 by iteratively solving the following two convex sub-problems:

$$\boldsymbol{\alpha}^k = \operatorname*{argmin}_{\boldsymbol{\alpha}} \sum_{i \in C} \alpha_i L(-\mathbf{w}^k \cdot \mathbf{d}_i) - \sigma || \, \boldsymbol{\alpha} \, ||^2 \tag{2.7}$$

$$[\mathbf{w}^{k+1}, b^{k+1}] = \operatorname*{argmin}_{(\mathbf{w},b)} \frac{\lambda}{2} ||\mathbf{w}||^2 + \sum_{i:\alpha_i^k=1} L(-(\mathbf{w} \cdot \mathbf{d}_i) + b) + \sum_{i \notin C} L(\mathbf{w} \cdot \mathbf{d}_i + b) \tag{2.8}$$

Given $\mathbf{w}^k$, it is trivial to solve for $\boldsymbol{\alpha}^k$ (Equation 2.7) because each $\boldsymbol{\alpha}_i^k$ can be solved for independently:

$$\boldsymbol{\alpha}^k = \operatorname*{argmin}_{\boldsymbol{\alpha}} \sum_{i \in C} \alpha_i L(-\mathbf{w}^{\mathbf{k}} \cdot \mathbf{d}_i) - \sigma || \, \boldsymbol{\alpha} \, ||^2 \tag{2.9}$$

$$\boldsymbol{\alpha}^k = \operatorname*{argmin}_{\boldsymbol{\alpha}} \sum_{i \in C} \alpha_i (L(-\mathbf{w}^{\mathbf{k}} \cdot \mathbf{d}_i) - \sigma) \tag{2.10}$$

$$\alpha_i^k = \operatorname*{argmin}_{\boldsymbol{\alpha}_i} \alpha_i (L(-\mathbf{w}^{\mathbf{k}} \cdot \mathbf{d}_i) - \sigma) \tag{2.11}$$

$$\alpha_i^k = \begin{cases} 0, & \text{if } L(-\mathbf{w}^{\mathbf{k}} \cdot \mathbf{d}_i) \geq \sigma \\ 1, & \text{if } L(-\mathbf{w}^{\mathbf{k}} \cdot \mathbf{d}_i) < \sigma \end{cases} \tag{2.12}$$

The sub-problem of learning $\mathbf{w}^{k+1}$ given $\boldsymbol{\alpha}^k$ (Equation 2.8) takes the form of a classical convex supervised learning problem. Because we use a hinge-loss, we use the publicly available primal SVM solver from [Chapelle, 2007] (with a slight modification to handle the positivity constraint). The entire procedure converges when $\boldsymbol{\alpha}^{k+1} = \boldsymbol{\alpha}^k$, since this implies that $\mathbf{w}^{k+1} = \mathbf{w}^k$. After solving each exemplar's learning problem, we scale the resulting distance function such that a distance value of 1 (instead of 0) corresponds to the decision boundary and a value of 0 (instead of $-\mathbf{w}_{N_F}$) corresponds to perfect similarity.

2.2 Related Work

Local learning has a long history — techniques based on locally weighted linear regression [Atkeson et al., 1997] have been successfully applied to other domains such as robot control. In vision, the work of Frome et al. [Frome and Malik, 2006, Frome et al., 2007] deals with a large number of object categories and performs exemplar-matching for the task of image classification (on Caltech 101). Frome's work, which was the first to propose learning per-exemplar similarity measures, has been very influential in the development of our work. It is perhaps not surprising that Frome's work on learning local-distance functions (like ours) is also heavily influenced by Wittgenstein's idea of family resemblances [Frome, 2007].

However, it is worth highlighting the biggest difference between our work and that of [Frome et al., 2007]: while we address the problem of object localization, [Frome et al., 2007] only considered image classification. At the representational level, [Frome et al., 2007] used local feature matching, allowing different views of the same object to potentially be matched, while we used much more rigid assemblies of features ensuring that drastically different views of an object will not be confused (even if some transformation of local features could place then in correspondence). Additionally, we use the best matching exemplars for interpretation which means that it is much more important in our framework to obtain visually similar matches, while [Frome et al., 2007] only evaluated their approach on category prediction.

Connection to non-convex learning. There is also an important connection between our use of friend indicator variables in Equation 2.6 and the Robust Support Vector Machine training algorithm of [Xu et al., 2006]. They show that the use of indicator variables is equivalent to using the Ramp-Loss, a truncated variant of the hinge loss which returns a constant loss for data points sufficiently far from the margin.

$$\begin{aligned} L_{ramp}(x) &= min(1, L_{hinge}(x)) \\ L_{hinge}(x) &= max(0, 1 - x) \end{aligned}$$

What this means is our distance function formulation could be re-written without any binary variables — the same effect could be obtained by replacing the convex loss function on the positives with the non-convex L_{ramp}. [Xu et al., 2006] also showed that when using an alternating optimization algorithm, it does not matter whether we force α_i to be binary or in the range $[0, 1]$ since the solution will always be either 0 or 1.

2.3 Evaluating Local Distance Functions on LabelMe

We are ultimately interested in parsing a scene into its constituent objects — understanding as much as we can about an input image. Doing so for a reasonably general class of images requires handling a large number of different objects that occur in everyday life. Therefore, the choice of the right training data is of the utmost importance. Of all the currently available datasets, the only one containing a large number of real-world scenes, with a wide variety of everyday objects that are not only labeled but also segmented, is the LabelMe dataset [Russell et al., 2008]. LabelMe is an ongoing online image-annotation effort involving many labelers. As a result, not only are the images user-contributed, spanning a wide range of scenes, but users are free to label each object with any English text string they like, providing a good sampling of the distribution of object names “in the wild”.

As a source of exemplars, we use a subset of LabelMe which consists of over 5000 images. After ignoring tiny objects, we clean up the object annotations by discarding auxiliary words from the labels (using the function provided in LabelMe toolkit), and keep all objects whose unique label occurs at least 5 times. This gives us a total of $12,905$ objects spanning 171 unique labels.

2.3.1 Segment-based Features

We represent each object by $N_F = 14$ different features. The features capture different aspects of shape, texture, color, and image location for an image segment (see Table 2.1). To capture information about shape we compute: the centered object mask in a canonical 32×32 frame, the size of the region, and the size of region's bounding box. To capture texture we compute normalized texton histograms in the interior of the object, and, separately, along the boundaries of the object. For color we compute the mean RGB-value, its standard deviation, as well as a color histogram. Finally, to capture knowledge about the position of the segment in an image, we compute a coarse (blurred) 8×8 absolute segmentation mask as well as the normalized height of the top-most and bottom-most pixel in the region.

2.3.2 Distance Function Learning Details

We define each of our distance functions as a linear combination of elementary distances. We use 14 color, shape, texture, and location features to represent exemplars. Instead of performing learning in one joint feature space created by concatenating the 14 different features, we perform learning in the reduced 14D elementary-distance space. Thus the $\mathbf{d}_i$ in Equation 2.6 is the vector of 14 Euclidean distances between the exemplar whose similarity we are learning (the focal exemplar) and the i-th example. We use $L(x) = max(1 - x, 0)^2$, the hinge-squared loss function, for learning which allows us to use a second order SVM solver [Chapelle, 2007].

Type	Name	Dimension
Shape	Centered Mask	32x32=1024
	BB Extent	2
	Pixel Area	1
Texture	Right Boundary Tex-Hist	100
	Top Boundary Tex-Hist	100
	Left Boundary Tex-Hist	100
	Bottom Boundary Tex-Hist	100
	Interior Tex-Hist	100
Color	Mean Color	3
	Color std	3
	Color Histogram	33
Location	Absolute Mask	8x8=64
	Top Height	1
	Bot Height	1

Table 2.1: **Segment Descriptor:** 14 **Elementary Features used to represent objects.** Elementary distances are simply the L_2 distances between corresponding feature vectors.

For every exemplar we initialize the distance function learning process with $\mathbf{w}^0 = \mathbf{w}_{texton}$, where $\mathbf{w}_{texton}$ has zeros along all elementary distance dimensions except the interior texton histogram feature (a bag-of-words feature). From the diverse combinations of heuristically defined distances/weights we experimented with, $\mathbf{w}_{texton}$ performed the best for a wide array of object types, and we use it as the baseline "learning-free" nearest-neighbor method.

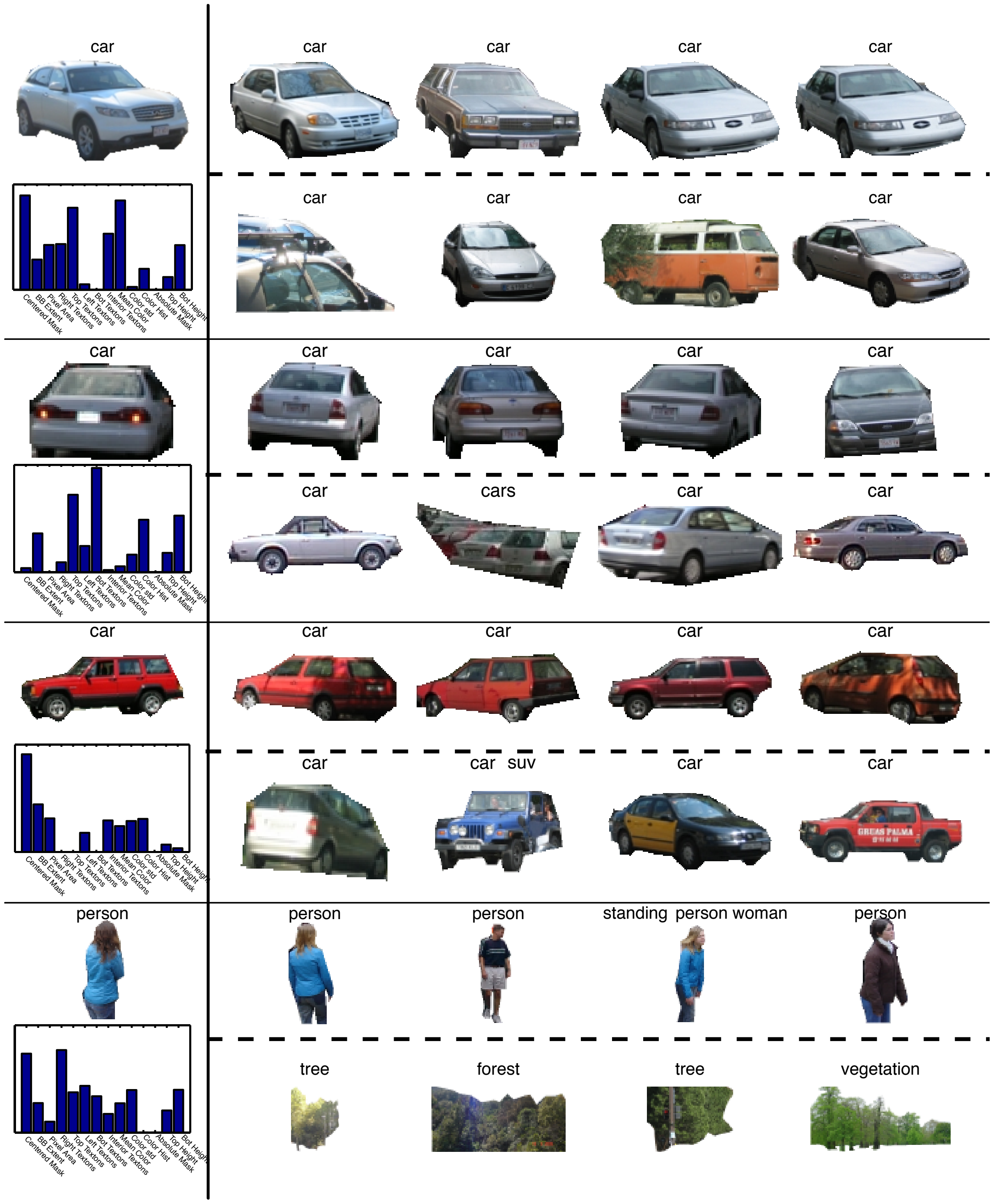


Figure 2.2: **Exemplar Association in the LabelMe Training Set 1/2.** Given an exemplar on the top left, the remaining row shows the top 4 most similar objects after learning a distance function. The distance function is visualized as a distribution over elementary distances and shown in the bottom left. The 4 exemplars on the bottom right are the 4 most similar objects with respect to the texton histogram distance.

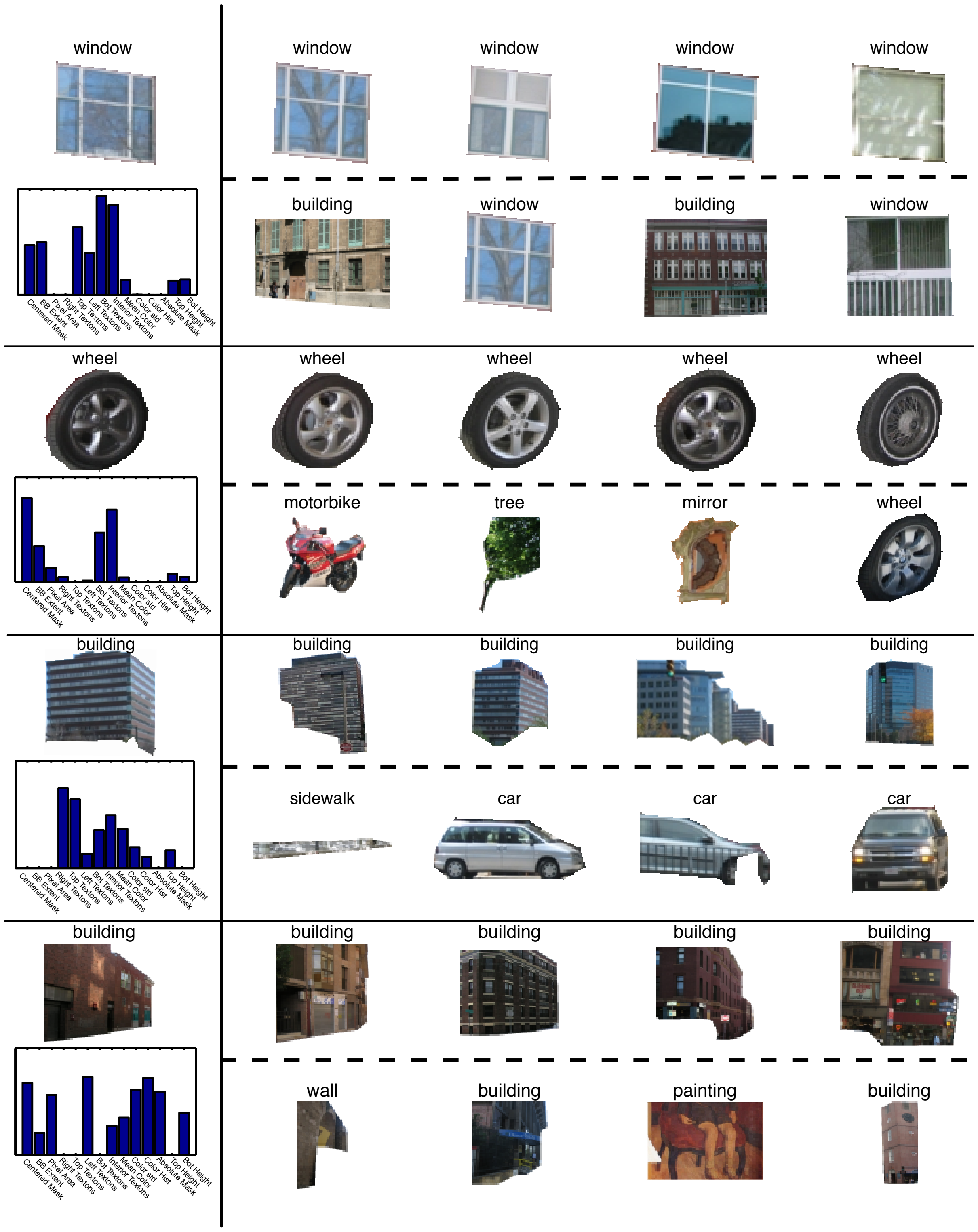


Figure 2.3: **Exemplar Association in the LabelMe Training Set 2/2.** Given an exemplar on the top left, the remaining row shows the top 4 most similar objects after learning a distance function. The distance function is visualized as a distribution over elementary distances and shown in the bottom left. The 4 exemplars on the bottom right are the 4 most similar objects with respect to the texton histogram distance.

Interpreting Learned Distance Functions. After learning the distance functions, we apply each exemplar's distance function to all of the other exemplars seen during training and define the support set of each exemplar as the exemplars which returned a score below 1.0.

$$z \in Supp(e) \leftrightarrow D_e(\mathbf{x}_z) < 1.0 \tag{2.13}$$

In practice the resulting support sets wildly vary in size. For exemplars from generic classes such as "sky" where we expect many skies to be rather similar, the support set is large. The support set can also be very small — which happens when its corresponding exemplar is either not visually distinctive or ambiguously/incorrectly labeled. We prune away the exemplars with an empty support set. Several learned distance functions and the top elements in their support sets are shown and compared to the neighbors given a simple texton-histogram distance in Figure 2.2 and Figure 2.3. The learned distance functions are doing a good job at combining elementary distances to measure similarity. Notice that an exemplar's support does not always contain exemplars with the same label. In particular an exemplar with the label "standing person woman" was deemed similar to the target exemplar with label "person" even though they are distinct labels. We measure how often this happens and show the top few elements of the label confusion matrix in Table 2.2. Notice that most of these confusions correspond to visual

stop sign	sign	7.8%	road highway	road	3.4%
pole	streetlight	6.7%	painting	picture	3.4%
motorcycle	motorbike	6.2%	sidewalk	road	3.2%
mountains	mountain	6.2%	cloud	sky	3.1%
ground grass	sidewalk	3.7%	grass	ground grass	3.1%
grass	lawn	3.6%	mountain	mountains	2.7%

Table 2.2: **Label confusion after distance function learning**. We show the top 12 label confusions discovered after distance function learning. For example, 7.8% of the time a "stop sign" exemplar associated with a "sign" exemplars. This suggests that visual similarity can alleviate some of the problems with ambiguous labels.

synonyms.

2.3.3 Segment Labeling

In order to determine if the distance functions are over-fitting, we consider a segment-labeling task on a held-out subset of LabelMe. Our evaluation uses a test set of 147 outdoor images all coming from one specific subfolder in LabelMe (to minimize the chances of similar data being used for training and testing). This testing subset contains a total of $1,146$ objects. For the segment-labeling task, we represent both exemplars and testing regions by segment-based features (as described in Section 2.3.1) extracted from the ground-truth segmentation masks. The more difficult problem of automatically extracting objects from images when no ground-truth segmentations are given is delayed until Section 2.3.4. We label the testing segment by the label from the single most visually similar exemplar using the learned distances. We consider a set of distance thresholds and compute the precision versus recall curve. Precision measures the probability that a returned label is identical to the ground truth label and in our case recall measures the fraction of segments that get labeled. As a baseline, we compare the performance of our learned distance functions to a nearest-neighbor classifier using a texton histogram distance. The precision-recall curve can be seen in Figure 2.4. If we only interpret the distance functions that return a value below 1.0 (what our learning formulation suggests is the best thing to do), we obtain labels for 60% of objects with a precision of 91%. This suggests that the learned distance functions are providing a very meaningful distance for recognition.

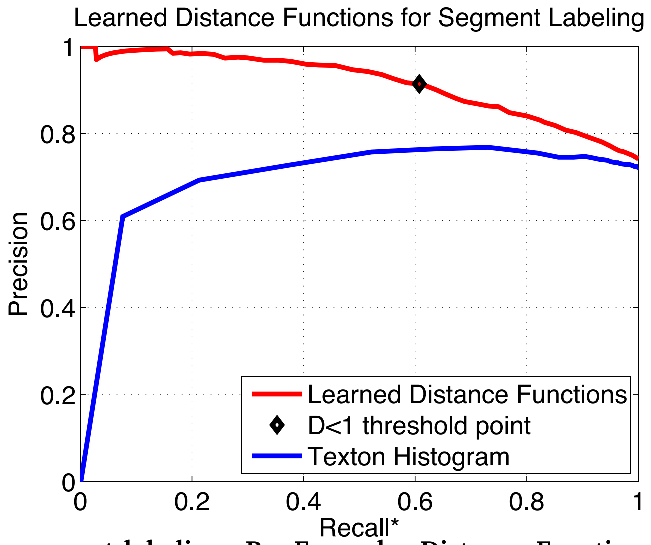


Figure 2.4: **LabelMe segment labeling: Per-Exemplar Distance Functions vs. Texton Histograms.** We learn a different distance function for each exemplar and apply them to a held-out set of objects from LabelMe. Segment features are obtained from ground-truth segmentations. Note that the Distance Functions are significantly better than Texton histograms in the high-precision, low-recall regime.

2.3.4 Parsing Images via Multiple Segmentations

We already saw in Section 2.3 that exemplar-specific distance functions can be used to determine the identity of a ground-truth segment, but how can we use them to segment out objects inside novel, unlabeled images? We tackle this problem in two steps. We use the multiple segmentation approach [Hoiem et al., 2005, Russell et al., 2006, Malisiewicz and Efros, 2007], which was shown to provide good spatial support for many different object types, to generate a large collection of candidate segments. We then keep the non-redundant subset of segments which generated well-scoring visual associations with the set of exemplars.

In [Malisiewicz and Efros, 2007], we argued at great length that using multiple segmentations is better than one — the key insight is that while no single segmentation is likely to produce all good segments, some of the segments in some of the segmentations are very good. We use a variant of the multiple segmentation approach [Russell et al., 2006, Malisiewicz and Efros, 2007], to generate a "soup of segments" in a purely bottom-up fashion (no object knowledge is provided). In particular, we vary the parameters of two segmentation engines — Mean-Shift based EDISON [Comaniciu and Meer, 2002] and Normalized Cuts [Shi and Malik, 2000] — to generate multiple image segmentations for every input image. Since we have shown in [Malisiewicz and Efros, 2007] that some composite objects are very unlikely to come out as a single segment in any segmentation, but can be well approximated by a merge of a few adjacent segments. Therefore, we augment our initial soup of segments by considering the merges of 2 or 3 adjacent segments as discussed in [Malisiewicz and Efros, 2007]. The resulting bottom-up segment representation can provide regions with good spatial support for both shape-free "stuff" objects such as grass, road, and sky as well as fixed-extent "things" such as cars, bicycles, and people. An additional advantage of using a bottom-up mechanism to generate candidate regions is that it is independent of the number of exemplars or object categories used in further processing stages.

The distance functions learned so far are not very good at recognizing bad segments — they never saw any in training! We thus augment the data used in distance function learning to contain a large number (over $30,000$) of bad segments which capture the appearance of patterns that do not correspond to any objects.

After we generate the soup of bottom-up segments, we compute the full matrix of distances between all exemplars and segments. We only consider the distances below 1.0 and the resulting associations are very sparse. On average, less than $.2\%$ of the potential associations are active. Qualitative examples of exemplar association in the soup of bottom-up segments can be seen in Figure 2.5. Quite often, a single segment will associate with many exemplars and we construct a recognition score out of the list of associating distances. Letting E be the list of exemplars associating with segment S

the recognition confidence $s(S, E)$ is constructed as follows:

$$s(S, E) = 1/\sum_{e \in E} \frac{1}{D_e(S)} \tag{2.14}$$

For evaluation purposes, we use the held-out subset of LabelMe defined in Section 2.3. We label each object hypothesis with the most frequently occurring label among its associations. We also retain all segments that associate with at least one exemplar, and thus have multiple (potentially all correct) overlapping object hypotheses. Since we do not want to penalize for these alternative associations we define detection precision as follows: we consider an object hypothesis to be correct if it has a segment overlap score (defined as in [Malisiewicz and Efros, 2007]) of at least 0.5 with a ground truth region that has the same identical label as the hypothesis. We consider all objects in tandem and do not penalize for multiple correct overlapping associations. We vary the recognition confidence to create the precision versus recall curve in Figure 2.6.

In order to quantify how well we segment objects, for each correct detection we measure the overlap score between the associated ground truth regions and the object hypotheses. We show the average overlap score as we vary the recognition confidence and compare that to the average overlap score of the best segment in our soup of segments. The corresponding plot can be seen on the right side of Figure 2.6.

The ability to return a small number of object hypotheses with high quality segmentation masks is crucial for image understanding. Even though the interplay between objects (e.g., [Rabinovich et al., 2007]) is certainly a crucial component for determining the identity of all the scene's visual elements, we can still create meaningful (partial) parses using our local distance functions alone. We create an image parse from our overlapping object hypotheses as follows: given a list of object hypotheses in a single image sorted by their recognition confidence and an initially empty list of objects in the

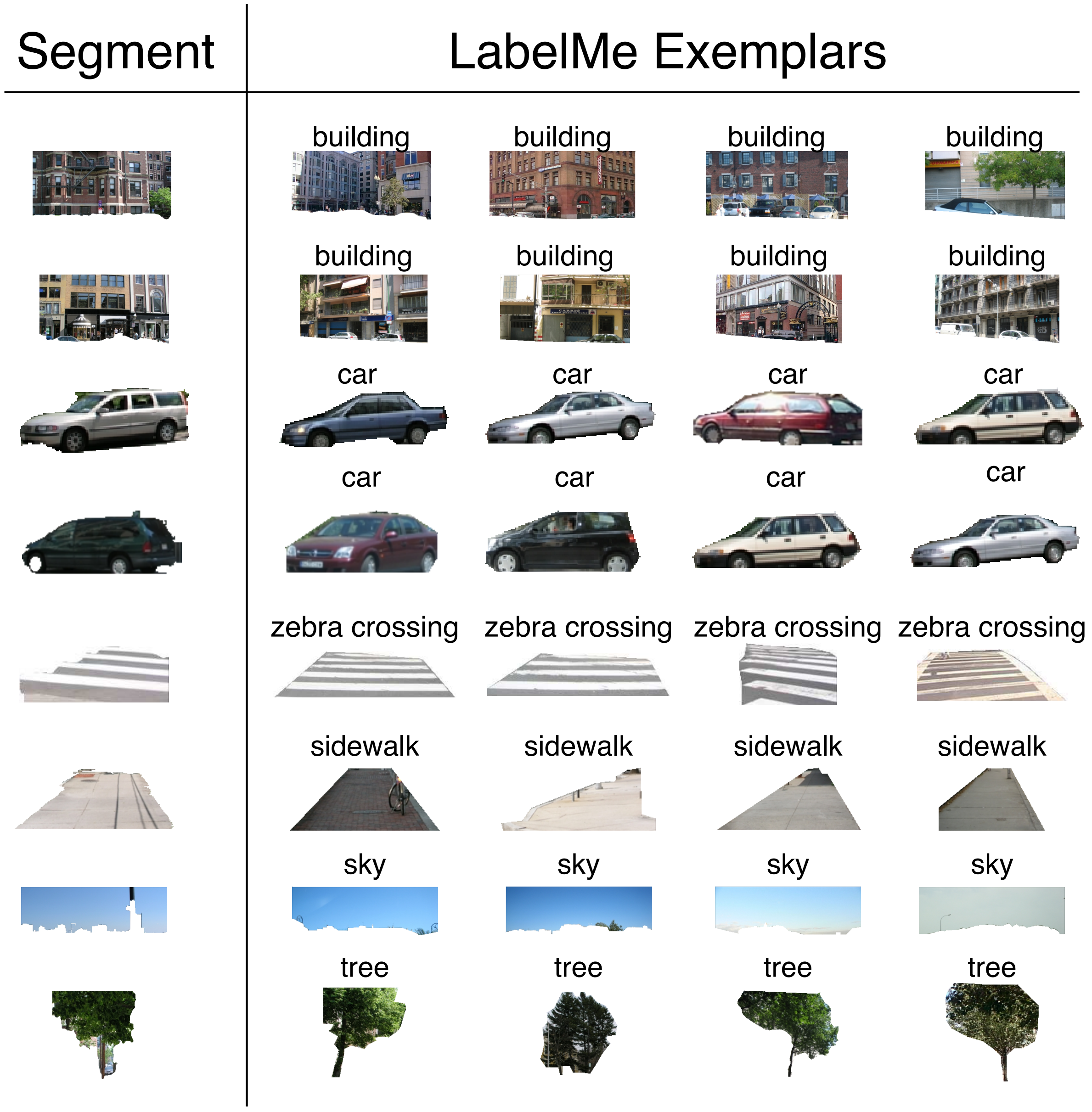


Figure 2.5: **Exemplar Association in the LabelMe test set.** Each example shows a segment generated from multiple segmentations along with its top 4 exemplar associations.

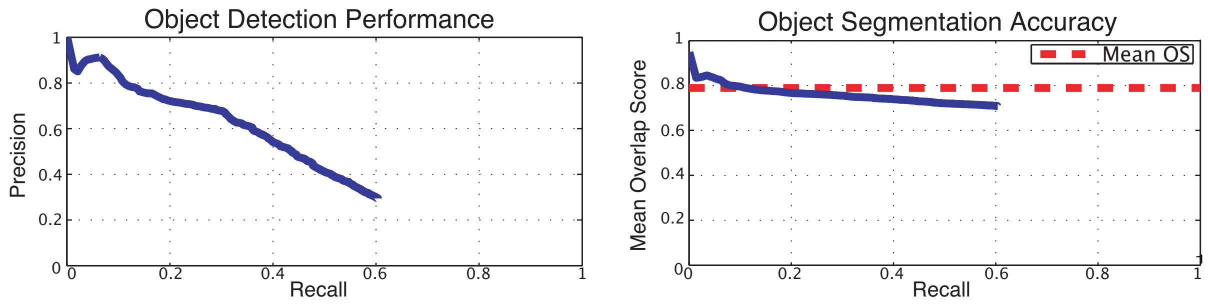


Figure 2.6: **Detecting and Segmenting objects in LabelMe.** Both curves are created by varying the recognition confidence (Equation 2.14). The first plot shows the precision-recall curve for the task of object detection. A detection is deemed correct if it returns the same label as well as has an overlap score (OS) greater than $.5$ with a ground-truth segment. The second plot shows the average segmentation quality of correct detections and compares that to the mean best overlap score of the input multiple segmentations.

parse, we greedily place the current best object hypotheses into the list of objects in the parse while removing all hypotheses that overlap with a score of 0.5 or more. This is equivalent to the typical non-maximum suppression algorithm (as used in [Felzenszwalb et al., 2008]), but applied to free-form segments instead of bounding boxes. Two resulting image parsing examples can be seen in Figure 2.7.

This chapter has demonstrated that multiple segmentations and Local Distance Functions allow us to perform detection and segmentation for a large number of different objects. In addition, we have shown that the integral component of such a segment-then-recognize approach is the learning of exemplar-specific distance functions.

2.3.5 Follow-up Work

Since its publication in 2008, several related approaches have been published and it is worthwhile mentioned them in this thesis. Regarding the “segment-then-recognize” approach, the approach of [Li et al., 2010] has become quite popular. [Li et al., 2010] use graph-cuts to generate a soup of figure-ground segmentations and rank them to perform object recognition. Our local distance function framework has been extended to the problem of recognition from both color and depth sensors [Lai et al., 2011]. In addition, there has been rising interest in local distance functions and many different formulations have been extensively analyzed by Ramanan and Baker [Ramanan and Baker, 2011].

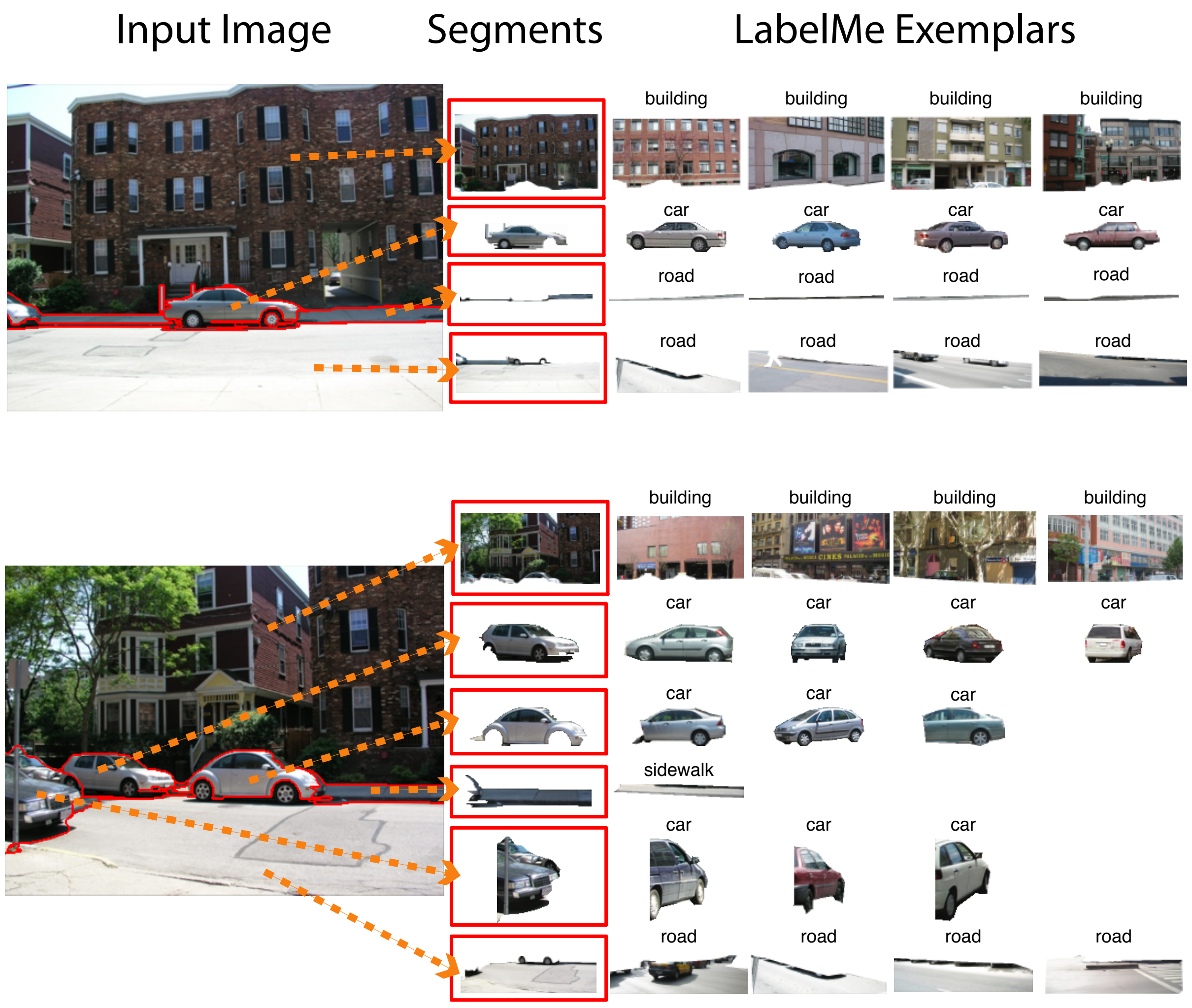


Figure 2.7: **Image Parsing via Multiple Segmentations and Local Distance Functions.** A parse is created by performing segment-based non-maximum suppression after scoring segments with Local Distance Functions.

Chapter 3

Exemplar-SVMs

A mere decade ago, automatically recognizing everyday objects in images (such as the bus in Figure 3.1) was thought to be an almost unsolvable task. Yet today, a number of methods can do just that with reasonable accuracy. But let us consider the output of a typical object detector — a rough bounding box around the object and a category label (Figure 3.1 left). While this might be sufficient for a retrieval task ("find all buses in the database"), it seems rather lacking for any sort of deeper reasoning about the scene. How is the bus oriented? Is it a mini-bus or a double-decker? Which pixels actually belong to the bus? What is its rough geometry? These are all very hard questions for a typical object detector. But what if, in addition to the bounding box, we are able to obtain an *association* with a very similar exemplar from the training set (Figure 3.1 right), which can provide a high degree of correspondence. Suddenly, any kind of meta-data provided with the training sample (a pixel-wise annotation or label such as viewpoint, segmentation, coarse geometry, a 3D model, attributes, etc.) can be simply transferred to the new instance.

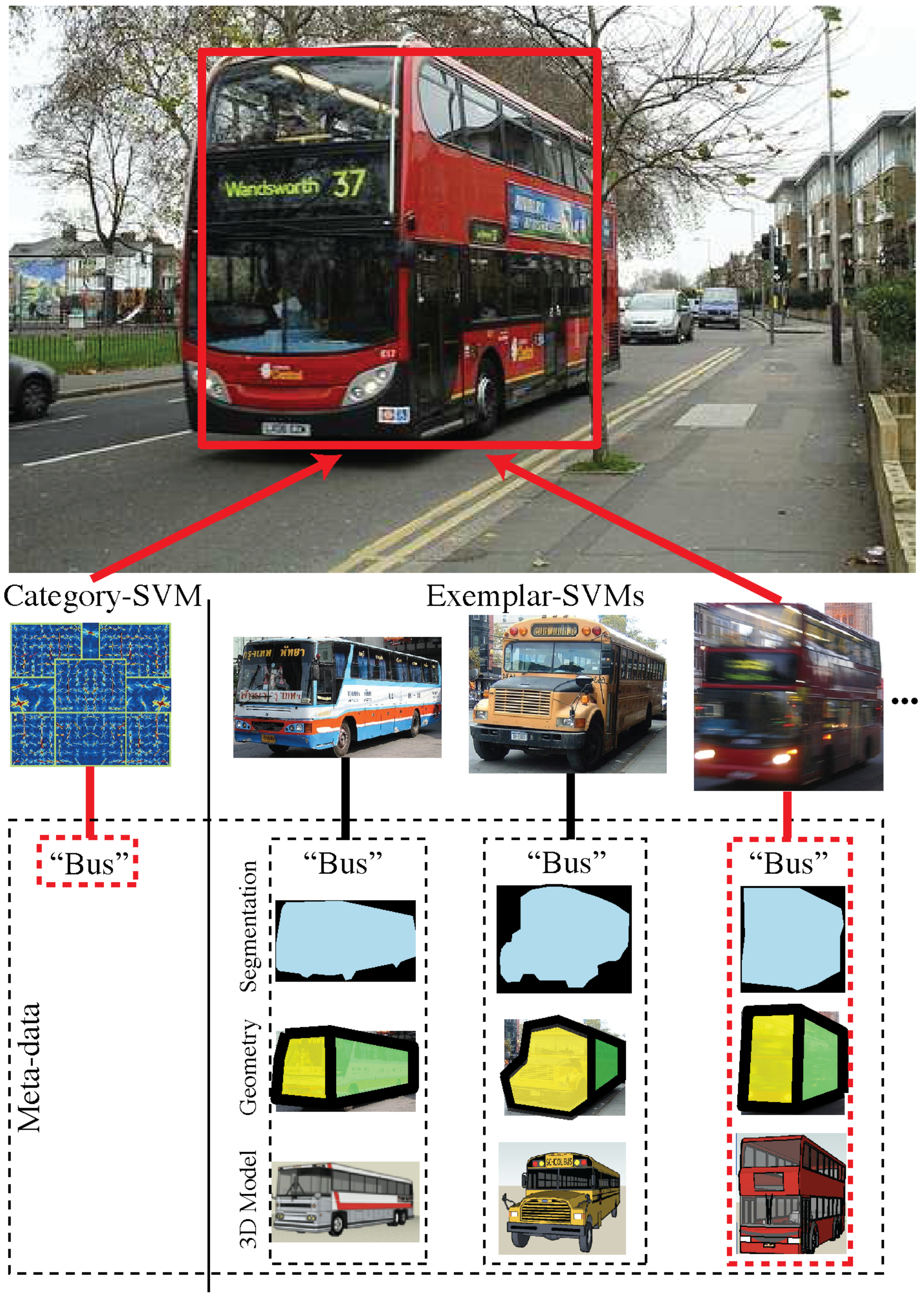


Figure 3.1: **Object Category Detector vs. Ensemble of Exemplar Detectors.** Output of a typical object detector is just a bounding box and a category label (left). But our ensemble of Exemplar-SVMs is able to associate each detection with a visually similar training exemplar (right), allowing for direct transfer of meta-data such as segmentation, geometry, even a $3D$ model (bottom).

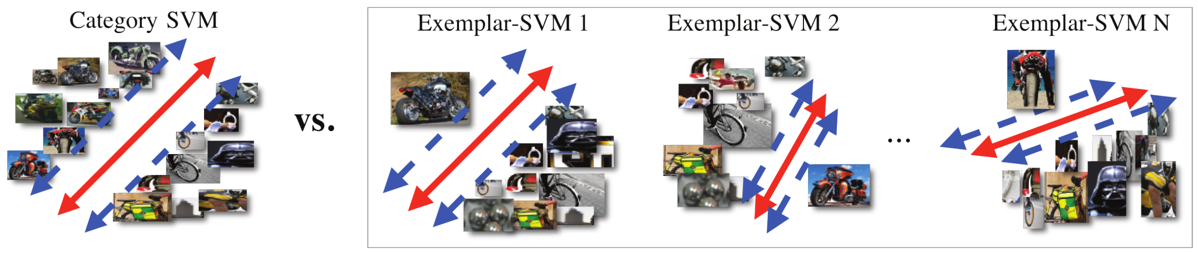


Figure 3.2: **Category SVM vs. Exemplar-SVMs.** Instead of training a single per-category classifier, we train a separate linear SVM classifier for each exemplar in our dataset with a *single* positive example and millions of negative windows. Negatives come from images not containing any instances of the exemplar's category.

What seems desirable is an approach that has all the strengths of a Dalal-Triggs [Dalal and Triggs, 2005] / Felzenszwalb et al. [Felzenszwalb et al., 2010] detector – powerful descriptor, efficient discriminative framework, clever mining of hard-negatives, etc. – but without the drawbacks imposed by a rigid, category-based representation of the positives. To put it another way, what we want is a method that is non-parametric when representing the positives, but parametric (or at least semi-parametric) when representing the negatives. This is the key motivation behind our Exemplar-SVM approach. What we propose is a marriage of the exemplar-based methodology, which allows us to propagate rich annotations from exemplars onto detection windows, with discriminative training, which allows us to learn powerful exemplar-based classifiers from vast amounts of positive and negative data.

Our object detector is based on a very simple idea: to learn *a separate classifier for each exemplar* in the dataset (see Figure 3.2). We represent each exemplar using a rigid HOG template [Dalal and Triggs, 2005]. Since we use a linear SVM, each classifier can be interpreted as a learned exemplar-specific HOG weight vector. As a result, instead of a single complex category detector, we have a large collection of simpler individual *Exemplar-SVM* detectors of various shapes and sizes, each highly tuned to the exemplar's appearance. But, unlike a standard nearest-neighbor scheme, each detector is discriminatively trained. So we are able to generalize much better without requiring

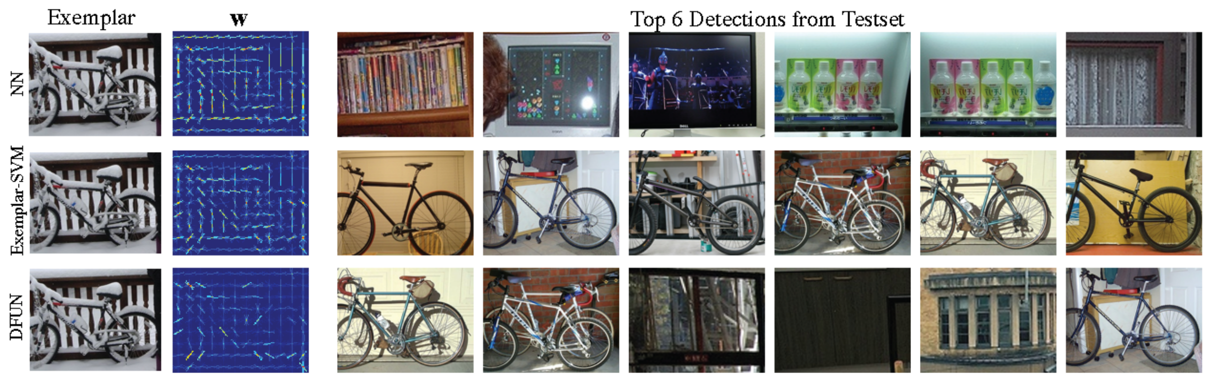


Figure 3.3: **HOG-matching comparison.** Given a bicycle training sample from PASCAL (represented with a HOG weight vector $\mathbf{w}$), we show the top 6 matches from the PASCAL test-set using three methods. **Row 1:** naive nearest neighbor (using raw normalized HOG). **Row 2:** Trained Exemplar-SVM (notice how $\mathbf{w}$ focuses on bike-specific edges). **Row 3:** Learned distance function – an Exemplar-SVM but trained in the "distance-to-exemplar" vector space, with the exemplar being placed at the origin (loosely corresponding to [Frome and Malik, 2006, Malisiewicz and Efros, 2008]).

an enormous dataset of exemplars, allowing us to perform surprisingly well even on a moderately-sized training dataset such as the PASCAL VOC 2007 [Everingham et al., 2010].

The similarity functions we propose in this Chapter take the form of a dot product between an exemplar's learned weight vector $\mathbf{w}_e$ (also referred to as an exemplar-specific template) and $\mathbf{x}_i$, the raw features representing an input object (plus a bias b_e).

$$D_e(\mathbf{x}_i) = \mathbf{w}_e^T \mathbf{x}_i + b_e \tag{3.1}$$

The output of such a similarity function can be interpreted as the signed distance from the hyperplane represented by $\mathbf{w}_e^T \mathbf{x} + b_e = 0$. The use of a hyperplane parameterization is motivated by the great success of templates in the object detection community (e.g., many of the best-performing approaches are the HOG-based Dalal-Triggs/Felzenszwalb et al. family of monolithic detectors). However, unlike monolithic approaches which learn category-specific templates (e.g., $\mathbf{w}_{dog}$, $\mathbf{w}_{car}$, $\mathbf{w}_{chair}$, etc), our templates are exemplar-specific (e.g., $\mathbf{w}_{e_1}, \mathbf{w}_{e_2}, \ldots, \mathbf{w}_{e_N}$) and trained in a per-exemplar fashion. In order to maintain the specificity of the learned template to a single exemplar

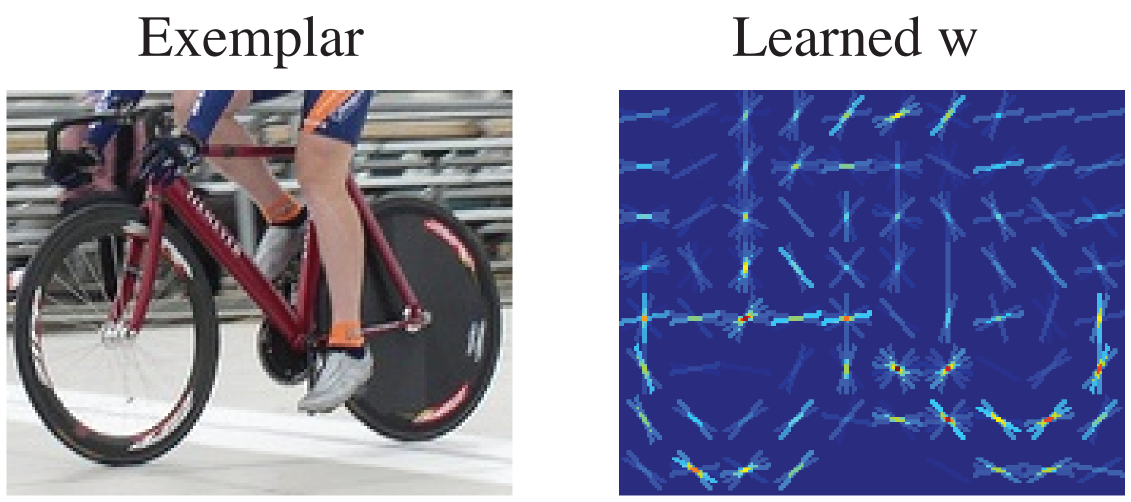


Figure 3.4: **Learning an Exemplar-specific HOG Template**. Since we use linear classifiers, the resulting decision boundary can be interpreted as an exemplar-specific HOG template. The visualization is the positive part of the learned hyperplane.

$\mathbf{x}_e$, we learn the template $\mathbf{w}_e$ with a **single positive instance**. The resulting weight vector can then be interpreted as a template highly tuned to the input exemplar (see Figure 3.4). Since we use the same linear Support Vector Machine (SVM) approach as [Dalal and Triggs, 2005] to learn $\mathbf{w}_e$, but are able to produce a template highly tuned to the exemplar $\mathbf{x}_e$, we dub this approach the **Exemplar-SVM**.

3.1 Who Needs friends? Learning With a Single Positive Instance

One would imagine that training an SVM with a single positive example will badly over-fit. But note that we require far less from a per-exemplar classifier as compared to a per-category classifier – each of our detectors only needs to perform well on visually similar examples. Since each classifier is solving a much simpler problem than in the full-category case, we can use a simple regularized linear SVM to prevent over-fitting. Another crucial component is that, while we only have a single positive example, we have millions of negative examples that we mine from the training set (i.e., from images that do not contain any instances of the exemplar's category). As a result, the exemplar's decision boundary is defined, in large part, by what it is *not*. One of the key contributions of our approach is that we show generalization is possible from a single positive example and a vast set of negatives.

At test-time, we independently run each classifier on the input image (see the sliding window discussion in Appendix section B.2) and use simple non-maximum suppression to create a final set of detections, where each detection is associated with a single exemplar. However, since our independently-trained classifiers might not output directly comparable scores, we must perform calibration on a validation set. The intuition captured by this calibration step is that different exemplars will offer drastically different generalization potential. A heavily occluded or truncated object instance will have poorer generalization than a cleaner exemplar, thus robustness against even a single bad classifier is imperative to obtaining good overall performance. Since our classifiers are trained without seeing any other positive instances but itself, we can use them for calibration in a "leave-all-but-one-out" fashion.

It is worthwhile pointing out some of the key differences between our approach and other related SVM-based techniques such as one-class SVMs [Schlkopf et al., 2001, Chen et al., 2001], multi-class kernel SVMs, kernel-learning approaches [Vedaldi et al., 2009], and the kNN-SVM algorithm [Zhang et al., 2006]. All of these approaches require mapping the exemplars into a common feature space over which a similarity kernel can be computed (which we avoid), but more importantly, kernel methods lose the semantics of single-exemplar associations which are necessary for high quality meta-data transfer.

Given a set of training exemplars, we represent each exemplar E via a rigid HOG template, $\mathbf{x}_E$. We create a descriptor from the ground-truth bounding box of each

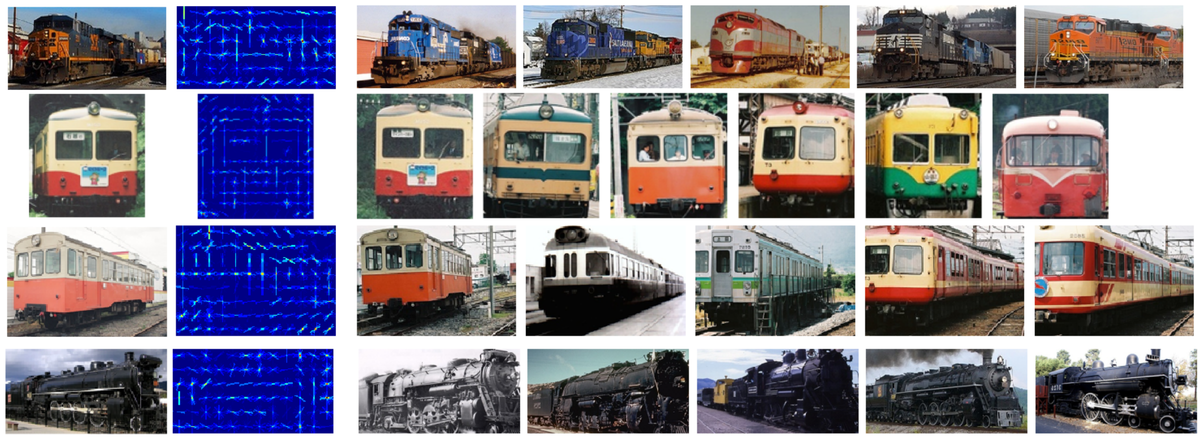


Figure 3.5: **Exemplar-SVMs.** A few "train" exemplars with their top detections on the PASCAL VOC test-set. Note that each exemplar's HOG has its own dimensions. Note also how each detector is specific not just to the train's orientation, but even to the type of train.

exemplar with a cell size of 8 pixels using a sizing heuristic which attempts to represent each exemplar with roughly 100 cells. Instead of warping each exemplar to a canonical frame, we let each exemplar define its own HOG dimensions respecting the aspect ratio of its bounding box (see the description of this process in Appendix section B.2). We create negative samples of the same dimensions as $\mathbf{x}_E$ by extracting negative windows, $\mathcal{N}_E$, from images not containing any objects from the exemplar's category.

Each Exemplar-SVM, $(\mathbf{w}_E, b_E)$, tries to separate $\mathbf{x}_E$ from *all* windows in $\mathcal{N}_E$ by the largest possible margin in the HOG feature space. Learning the weight vector $\mathbf{w}_E$ amounts to optimizing the following convex objective:

$$\Omega_E(\mathbf{w}, b) = ||\mathbf{w}||^2 + C_1 h(\mathbf{w}^T\mathbf{x}_E + b) + C_2 \sum_{\mathbf{x}\in\mathcal{N}_E} h(-\mathbf{w}^T\mathbf{x} - b) \tag{3.2}$$

We use the hinge loss function $h(x) = \max(0, 1 - x)$, which allows us to use the hard-negative mining approach to cope with millions of negative windows because the solution only depends on a small set of negative support vectors.

Figure 3.3 offers a visual comparison of the proposed Exemplar-SVM method against two alternatives for the task of detecting test-set matches for a single exemplar, a snowcovered bicycle. The first row shows a simple nearest-neighbor approach. The second row shows the output of our proposed Exemplar-SVM. Note the subtle changes in the learned HOG vector $\mathbf{w}$, making it focus more on the bicycle. The third row shows the output of learning a distance function, rather than a linear classifier. For this, we applied the single-positive Exemplar-SVM framework in the "distance-to-exemplar" vector space, with the exemplar being placed at the origin (this is conceptually similar to [Malisiewicz and Efros, 2008, Frome and Malik, 2006]). We observed that the centered-at-exemplar constraint made the distance function less powerful than the linear classifier (see Results section). Figure 3.5 shows a few Exemplar-SVMs from the "train" category along with their top detections on the test-set. Note how specific each detector is – not just to the train's orientation, but even the type of train.

3.1.1 Relationship to Local Distance Functions

Since Exemplar-SVMs are trained in a per-exemplar fashion, they bear a strong resemblance to the Local Distance Functions defined in Chapter 2. However, there are some key differences which are worthwhile to explain in detail. Unlike distance functions [Frome and Malik, 2006, Malisiewicz and Efros, 2008], which require a set of positive examples during learning, Exemplar-SVMs only require a **single positive instance**. We additionally use **millions** of image windows as the negatives used to train $\mathbf{w}_e$. The shift towards using *less* positive data and *more* negative data is a notable change in philosophy for learning per-exemplar similarity measures. By virtue of being a hyperplane, the Exemplar-SVM similarity function is not required to be centered at the exemplar and allows us to better adapt to the training data. Using a single positive instance, we abandon the need for any latent variables (which were used to encode a variable number of friends during distance function learning in Section 2), and the resulting problem is convex. This means that the optimization problem involved is much faster to solve (by using highly optimized SVM packages tuned for large-scale problems) and we do not have to worry about local minima (a hallmark of convexity). More importantly, requiring a single positive instance means that our dependence on category labels is much lower than the distance functions in Section 2 (which required the use of categories to define the positive class). This is an important step in the direction of completely moving away from categories during learning.

Because Exemplar-SVMs can be interpreted as exemplar-specific similarity measures, our framework shares some similarities with distance-learning approaches, in particular those that learn per-exemplar distance functions (e.g., [Frome and Malik, 2006, Malisiewicz and Efros, 2008]). However, the crucial difference between a per-exemplar classifier and a per-exemplar distance function is that the latter forces the exemplar itself to have the maximally attainable similarity. An Exemplar-SVM has much more freedom in defining the decision boundary, and is better able to incorporate input from the negative samples (see Figure 3.3 for a comparison, to be discussed later).

3.1.2 Interpreting SVMs trained with a single positive instance

The Representer Theorem [Wahba, 1973] tells us that the solution to an SVM problem (a type of L_2-regularized learning problem) will take the following form:

$$\mathbf{w}_e = \alpha_e \mathbf{x}_e + \sum_{i \in \mathcal{N}_e} \alpha_i \mathbf{x}_i \tag{3.3}$$

Here, α_e is the dual variable associated with the single positive instance, and α_i is a dual variable associated with the negatives. We have observed that all of our Exemplar-SVMs are able to separate the single positive from the negatives, and that the support vectors roughly fall on their respective margin. $\mathbf{x}_e$ is one support vector with $\alpha_e \approx 1$, and we get a small set of negative support vectors with $\alpha_i \approx -1$. All non-supporting vectors have $\alpha_i = 0$. Indicating the set of negative support vectors (also called hard negatives) as NSV, we get:

$$\mathbf{w}_e \approx \mathbf{x}_e - \sum_{i \in \mathtt{NSV}} \mathbf{x}_i \tag{3.4}$$

It now becomes clear that the Exemplar-SVM learns a template whose positive part will resemble the exemplar and the negative part is a sum over negative support vectors.

3.2 Calibration

Using the procedure above, we train an ensemble of Exemplar-SVM, one for each positive instance in the training set. However, due to the independent training procedure, their outputs are not necessarily compatible. A common strategy to reconcile the outputs of multiple classifiers is to perform calibration by fitting a probability distribution to a held-out set of negative and positive samples [Platt, 1999, Frome and Malik, 2006]. This procedure is often called *Platt's Scaling Algorithm*, and generally applied to a held-out set which has the same distribution of samples as the training-data. However, in our case, since each exemplar-SVM is supposed to fire only on *visually similar* examples, we cannot say for sure which of the held-out samples should be considered as positives *a priori*. For example, for a frontal view of an train, only other frontal views of similar trains should be considered as positives. Fortunately, just like during training, what we can be sure about is that the classifier should not fire on negative windows. Therefore, we let each exemplar select its own positives and then use the SVM output scores on these positives, in addition to lots of held-out negatives, to calibrate the Exemplar-SVM.

To obtain each exemplar's calibration positives, we run the Exemplar-SVM on the validation set, create a set of non-redundant detections using non-maximum suppression, and compute the overlap score between resulting detections and ground-truth bounding-boxes. We treat all detections which overlap by more than 0.5 with ground-truth boxes as positives (this is the standard PASCAL VOC criterion for a successful detection). All detections with an overlap lower than 0.2 are treated as negatives, and we fit a logistic function to these scores. Note that, although we cannot guarantee that highly overlapping correct detections will indeed be visually similar to the exemplar, with very high probability they will be, since they were highly ranked by the exemplar-SVM in the first place.

Our calibration step can be interpreted as a simple re-scaling and shifting of the decision boundary (see Figure 3.6) – poorly performing exemplars will be suppressed by having their decision boundary move towards the exemplar and well-performing exemplars will be boosted by having their decision boundary move away from the exemplar. While the resulting decision boundary is no longer an optimal solution for the local-SVM problem, empirically we found this procedure greatly improves the inter-exemplar ordering. Given a detection x and the learned sigmoid parameters (α_E, β_E), the calibrated detection score for exemplar E is as follows:

$$f(\mathbf{x}|\mathbf{w}_E, \alpha_E, \beta_E) = \frac{1}{1 + e^{-\alpha_E(\mathbf{w}_E^T\mathbf{x} - \beta_E)}}$$

While the logistic fitting is performed independently for each exemplar, we found that it gives us a considerable boost in detection performance over using raw SVM output scores. At test-time, we create detections from each classifier by thresholding the raw SVM output score at -1 (the negative margin) and then rescale them using each exemplar's learned sigmoid parameters.

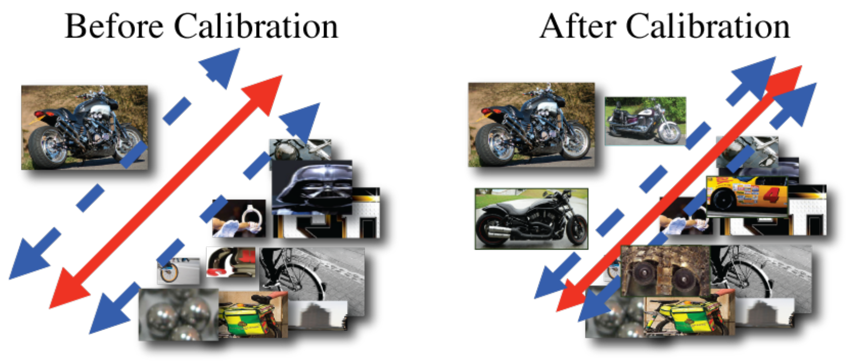


Figure 3.6: **Exemplar-SVM calibration**. The calibration step rescales the SVM scores but does not affect the ordering of the matches, allowing us to compare the outputs of multiple independently-trained Exemplar-SVMs.

3.3 Exemplar Co-occurrence Matrices

After calibration, we can create detection windows for each exemplar in a sliding-window fashion. A common and simple mechanism for suppressing redundant responses is non-maximum suppression (NMS); however, using NMS directly on exemplars means that multiple exemplars will be *competing* for detections windows. However, easy to recognize objects will often have many visual associations created around the object of interest — information which competitive NMS completely disregards. Instead of just using the raw association score, we propose to augment each detection with an exemplar context score which uses the identities and scores of nearby detections to boost the raw detection score. For each detection we generate a context feature similar to [Bourdev et al., 2010, Felzenszwalb et al., 2010] which pools in association scores of nearby (overlapping) detections and generates the final detection score by a weighted sum of the local association score and the context score. A set of K detections $\mathcal{D}$ is produced by applying a set of N exemplars to a single image and performing NMS across detections with the same exemplar id. While NMS makes sure that we throw away redundant detections from the same exemplar, we will still be left with many overlapping detections because many exemplar are similar to each other.

We can think of each detection as a triplet consists of a rectangular detection region $\mathbf{R}_i$, the exemplar id e_i, and the exemplar's visual association score s_i.

$$\begin{aligned} \mathcal{D} &= \{(\mathbf{R}_i, e_i, s_i)\}_{i=1}^{K} && (3.5) \\ \mathbf{R}_i &\in \Re^4 && (3.6) \\ e_i &\in \{1, \ldots, N\} && (3.7) \\ s_i &= \mathbf{w}_{e_i}^T \mathbf{x}_{R_i} + b_{e_i} + 1.0 && (3.8) \end{aligned}$$

Because we thresholded detections at -1.0, adding 1.0 to each Exemplar-SVM's raw output score guarantees that $s_i \geq 0$. We now define the exemplar context feature $\mathbf{f}(\mathbf{R}_i) \in \Re^{N+}$, associated with detection region $\mathbf{R}_i$, as follows:

$$\mathbf{f}(\mathbf{R}_i) = [f_1 \quad f_2 \quad \ldots \quad f_N] \tag{3.9}$$

$$f_j = \max_{(e_k = j) \cap (\mathbf{R}_k \in \mathcal{F}_i)} s_k \tag{3.10}$$

$$\mathcal{F}_i = \{\mathbf{R} : OS(\mathbf{R}, \mathbf{R}_i) > .5\} \tag{3.11}$$

Simply put, for a detection region $\mathbf{R}_i$, we only consider other exemplar detections in the set $\mathcal{F}_i$ (the set of regions which overlap with $\mathbf{R}_i$ by more than 0.5). The j-th component of the exemplar context feature is then the maximum score of a detection with exemplar id e_j. The final score for region $\mathbf{R}_i$ is then:

$$\widehat{s}_i = \mathbf{m}_{e_i}^T \mathbf{f}(\mathbf{R}_i) \tag{3.12}$$

$$M = [\mathbf{m}_{e_1} \quad \mathbf{m}_{e_2} \quad \ldots \quad \mathbf{m}_{e_N}] \tag{3.13}$$

where M is the $N \times N$ exemplar co-occurrence matrix which encodes how often exemplars e_i and e_j simultaneously created a correct detection on the `trainval` set. We essentially loop over all detections in the `trainval` set, and count how often exemplars e_i and e_j produced a correct detection. Once we obtain the final association score $\widehat{s}_i$ using the co-occurrence matrix, we use standard non-maximum suppression to create a final, sparse set of detections per image.

3.4 Evaluation of Exemplar-SVMs

We evaluate the Exemplar-SVM framework two different ways. First, we compare the learned per-exemplar templates against templates computed directly from the positive instance, without the use of any negatives. Second, we compare the functional form of Exemplar-SVMs, against a distance function centered around the exemplar (see Section 2). We use the same exact per-exemplar feature representation in all three cases, and perform training with the same parameters for the two learning-based approaches.

Normalized HOG: To study the effect of learning, we need a way of creating visual similarities without any learning. We found that a simple dot product with a normalized template works well as a nearest-neighbor baseline. The learning-free $\widehat{\mathbf{w}}_e$ is obtained by creating the 0-mean, normalized HOG descriptor directly from the exemplar's features ($\mathbf{x}_e \in \Re^F+$) as follows:

$$\widehat{\mathbf{w}}_e = \mathbf{x}_e - \mu_x \mathbf{1}_F \tag{3.14}$$

$$\mu_x = \frac{\mathbf{x}_e^T \mathbf{1}_F}{F} \tag{3.15}$$

In this equation $\mathbf{1}_F$ is the F dimensional vector of ones, and in order to be 0-mean the final template has both positive and negative parts. The idea behind using the normalized HOG template comes from our observation that the templates obtained from the Exemplar-SVM algorithm are mean 0 and the positive part looks very much like the original raw HOG features. One interpretation of the normalized HOG template is the solution of a “virtual” SVM learning process where negative support vectors are uniformly distributed gradient templates. In other words, the normalized HOG template implicitly assumes that all gradients in the negative world are equally likely, while the Exemplar-SVM approach does not make any assumptions about the negative world (and explicitly mines the negatives from a large set).

HOG Distance Functions: To compare the template dot-product representation of similarity functions against our earlier “centered-at-exemplar” philosophy, we performed experiments on learning using the diagonal Mahalanobis functional form of Equation 2.1. To make sure that the only difference in learning is the change of the functional form, we performed learning using the same single-positive-instance objective function as the Exemplar-SVM, as well as the same strategy for mining negative windows. The only difference is that instead of applying the learned weight vector in raw feature space $\mathbf{w}_e^T\mathbf{x}_i$, we apply it in the distance-to-exemplar space, $\mathbf{w}_e^T\mathbf{d}_{ei}$.

The full PASCAL VOC detection challenge results will be discussed later, but we summarize the results here for completeness. Using the same calibration step (which will be described in Section 3.2) for all 3 methods, we see that Normalized HOG obtains a mAP of $.110$, Local Distance Functions obtain a mAP of $.157$, and Exemplar-SVMs obtain a mAP of $.198$. The full set of results can be seen in Table 3.1.

3.4.1 PASCAL VOC Object Detection Task

We evaluate our Exemplar-SVM framework on the well-established PASCAL VOC benchmark task of object detection. For our experiments, we use a single source of exemplars:

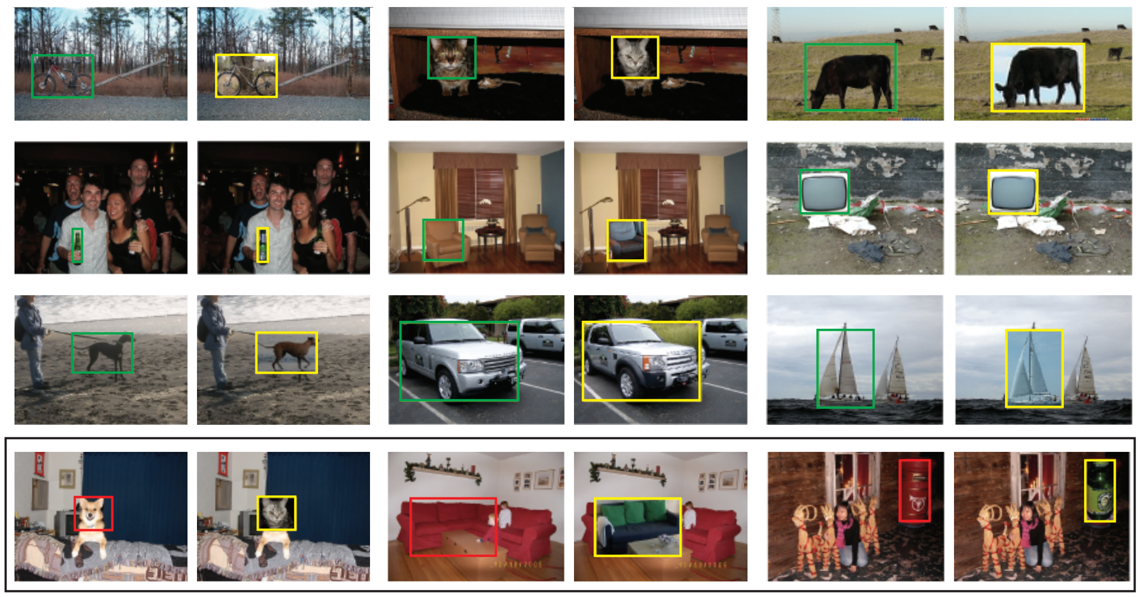


Figure 3.7: **Object Detection and Appearance Transfer.** Each example shows a detection from our ensemble of Exemplar-SVMs along with the appearance transferred directly from the source exemplar, to demonstrate the high quality of visual alignment. Bottom row shows object category detection failures.

the PASCAL VOC 2007 dataset [Everingham et al., 2010] – a popular dataset used to benchmark object detection algorithms. During training, we learn a separate classifier $\mathbf{w}$ for each of the $12,608$ exemplars from the 20 categories in $5,011$ `trainval` images. We mine hard negatives from out-of-class images in the `train` set and perform calibration using all positive and negative images in `trainval` (See Section 3.2).

At test time, each Exemplar-SVM creates detection windows in a sliding-window fashion, but instead of using a standard non-maxima-suppression we use an exemplar co-occurence based mechanism for suppressing redundant responses. For each detection we generate a context feature similar to [Bourdev et al., 2010, Felzenszwalb et al., 2010] which pools in the SVM scores of nearby (overlapping) detections and generates the final detection score by a weighted sum of the local SVM score and the context score. Once we obtain the final detection score, we use standard non-maximum suppression to create a final, sparse set of detections per image.

We report results on the 20-category PASCAL VOC 2007 `comp3` object detection

	Exemplar Methods								Global	
Type	NN	NN+Cal	DFUN	DFUN+Cal	ESVM	ESVM+Cal	ESVM+Co	CZ	DT	LDPM
aeroplane	0.006	0.056	0.044	0.162	0.062	0.204	0.208	0.262	0.127	0.287
bicycle	0.094	0.293	0.299	0.364	0.333	0.407	0.480	0.409	0.253	0.510
bird	0.000	0.012	0.006	0.008	0.092	0.093	0.077	x	0.005	0.006
boat	0.005	0.034	0.093	0.096	0.099	0.100	0.143	x	0.015	0.145
bottle	0.000	0.009	0.005	0.097	0.025	0.103	0.131	x	0.107	0.265
bus	0.006	0.207	0.153	0.316	0.236	0.310	0.397	0.393	0.205	0.397
car	0.010	0.261	0.202	0.366	0.329	0.401	0.411	0.432	0.230	0.502
cat	0.092	0.017	0.092	0.092	0.095	0.096	0.052	x	0.005	0.163
chair	0.001	0.094	0.095	0.098	0.095	0.104	0.116	x	0.021	0.165
cow	0.092	0.111	0.097	0.107	0.118	0.147	0.186	x	0.128	0.166
diningtable	0.001	0.004	0.003	0.002	0.016	0.023	0.111	x	0.014	0.245
dog	0.004	0.033	0.092	0.093	0.094	0.097	0.031	x	0.004	0.050
horse	0.096	0.243	0.253	0.234	0.340	0.384	0.447	x	0.122	0.452
motorbike	0.094	0.188	0.196	0.223	0.287	0.320	0.394	0.375	0.103	0.383
person	0.005	0.114	0.097	0.120	0.124	0.192	0.169	x	0.101	0.362
pottedplant	0.018	0.020	0.012	0.037	0.020	0.096	0.112	x	0.022	0.090
sheep	0.009	0.129	0.040	0.117	0.117	0.167	0.226	x	0.056	0.174
sofa	0.008	0.003	0.047	0.016	0.098	0.110	0.170	x	0.050	0.228
train	0.096	0.183	0.180	0.271	0.205	0.291	0.369	0.334	0.120	0.341
tvmonitor	0.144	0.195	0.229	0.293	0.221	0.315	0.300	x	0.248	0.384
mAP	0.039	0.110	0.112	0.155	0.150	0.198	0.227	x	0.097	0.266

Table 3.1: **Exemplar-SVM Object Detection Results on PASCAL VOC 2007.** The 3 different local similarity functions we compare against are: NN (Normalized Hog Nearest Neighbor), DFUN (learning a diagonal Mahalanobis distance function), and ESVM (Exemplar-SVM). For each method we also perform calibration, as indicated with a +*Cal*, and apply the contextual co-occurrence matrix (+*Co*) to the best performing ESVM method. The two global methods are our implementation of Dalal-Triggs (learning a single global template), and LDPM (Latent deformable part model). Our method beats out the Dalal-Triggs baseline across all categories. Our approach obtains a mAP of 22.7 rivaling the mAP of 26.7 obtained from Felzenszwalb *et* al [Felzenszwalb et al., 2010] discriminatively trained part-based mixture model. CZ stands for [Chum and Zisserman, 2007] and LDPM stands for the Latent Deformable Part-based Model of [Felzenszwalb et al., 2010]. Due to computational reasons, we computed the NN baselines on the person category with 1250 exemplars.

challenge. Figure 3.7 shows several detections (green boxes) produced by our Exemplar-SVM framework. We also show the super-imposed exemplar (yellow boxes) associated with each detection. Following the protocol of the VOC Challenge, we evaluate our system on a per-category basis on the `test` set, consisting of $4,952$ images. We compare the performance of our approach (**ESVM+Co-oc**) to several exemplar baselines apart from the VOC results reported in [Felzenszwalb et al., 2010, Chum and Zisserman, 2007]. These results have been summarized in Table 3.1 as Average Precision per class. Our results show that standard Nearest Neighbor [1] (NN) does not work at all. While the performance improves after calibration (NN+Cal), it is still not comparable to other approaches due to its lack of modeling negative data. We also compared against a distance function formulation similar to the one proposed in [Malisiewicz and Efros, 2008] but learned using a single positive instance. The results clearly indicate that the extra constraint due to a distance function parameterization is worse than using a hyperplane. To highlight the importance of using the co-occurence mechanism above, we also report our results using calibration (ESVM+Cal).

On the PASCAL test set, our full system obtains a mean Average Precision (mAP) of $.227$, which is competitive with with Felzenszwalb's state-of-the-art deformable part-based mixture model. Note however, that our system does not use parts (though they could be easily added) so the comparison is not entirely fair. Therefore, we also compare our performance to Dalal/Triggs baseline, which uses a single category-wise linear SVM with no parts, and attains a mAP of $.097$, which is less than half of ours. We also compared against the PASCAL VOC 2007 winning entry, the exemplar-based method of Chum et al. [Chum and Zisserman, 2007], and found that our system beats it on 4 out of 6 categories for which they submitted results. In [Chum and Zisserman, 2007],

[1]We experimented with multiple similarity metrics and found that a dot product with a normalized HOG template worked the best. The normalized HOG template is created by subtracting a constant from the positive HOG features to make them 0-mean.

the winning entry for many of the categories in PASCAL 2007, object categories are represented by storing exemplars, but the underlying assumption is that all exemplars from the same class share a common distribution of features. Thus, Chum et al. require class-wise *and* aspect-wise labeling of training data to break up broad basic-level object categories such as cars into finer visual sub-categories (e.g., car-left, car-front). Unfortunately this type of labeling is tedious and difficult to obtain for datasets of significant size. Moreover, for some objects (e.g., bottles, soccer balls, grass) it is not clear if aspect-wise labeling is even well-defined.

3.5 Exemplar-SVM Analysis

In this section, we analyze in detail some properties of our Exemplar-SVM object detection system. In particular, we study object detection performance when using different negative sets (by varying their content as well as their size) as well as different positive sets (by varying the number of exemplars to be used at test-time).

3.5.1 Including Same-Category Instances in the Negative Set

In the last section, we showed that the Exemplar-SVM training procedure works with a single positive instance and millions of negatives. We also discussed that this implies that the exemplar's decision boundary is defined, in large part, by what it is *not*. Because the negative set was created by eliminating images containing same-category (or, in-class) instances, it depends on the positive's category. To determine how the performance of our system depends on eliminating same-category instances from the negative set, we repeat the PASCAL VOC 2007 experiments from section 3.4.1, but this time we allow the negatives to come from *any* image, including in-class images. For each exemplar in `trainval`, we use all of the images from the `train` set (excluding the exemplar's own

Type	ESVM*	ESVM*+Cal	ESVM*+Co	ESVM+Co
aeroplane	0.043	0.121	0.114	0.208
bicycle	0.216	0.310	0.392	0.480
bird	0.092	0.032	0.095	0.077
boat	0.094	0.098	0.143	0.143
bottle	0.010	0.036	0.124	0.131
bus	0.155	0.202	0.323	0.397
car	0.185	0.264	0.343	0.411
cat	0.093	0.093	0.035	0.052
chair	0.049	0.077	0.114	0.116
cow	0.117	0.128	0.193	0.186
diningtable	0.011	0.012	0.096	0.111
dog	0.094	0.095	0.053	0.031
horse	0.265	0.258	0.381	0.447
motorbike	0.224	0.235	0.360	0.394
person	0.102	0.128	0.162	0.169
pottedplant	0.016	0.093	0.065	0.112
sheep	0.065	0.159	0.210	0.226
sofa	0.094	0.023	0.121	0.170
train	0.150	0.212	0.302	0.369
tvmonitor	0.185	0.267	0.281	0.300
mAP	0.113	0.142	0.195	0.227

Table 3.2: **Exemplar-SVM results when mining negatives from both out-class and in-class images**. We augment the Exemplar-SVM approach by allowing the negatives to come from any image, including in-class images (indicated by ESVM*). For comparison, the rightmost column (ESVM+Co) is the original method (taken from Table 3.1) where in-class images are eliminated from the negative set.

image) to define the negative set. After learning the **w**'s, we perform calibration (see Section 3.2) and co-occurrence matrix estimation (see Section 3.3) the same way as before (i.e., utilizing the exemplar's category).

As can be seen from the results in Table 3.2, the final mAP is 0.195, which is only slightly below the 0.227 mAP from the original method. This suggests that once the negative set is large enough, the few potentially confusing "positives" in the negative set are not fatal to our system. The percentage drop seems to be fairly consistent for the well-performing categories. Notice that for 4 categories (bird, boat, cow, dog), the ESVM*+Co results were equal to or better than the original ESVM+Co method. The results for cow, a reasonably-performing category, even improved from 0.186 to 0.194.

3.5.2 Size of Negative Set

Because Exemplar-SVMs must be trained with lots of negatives, each exemplar's data-mining step requires approximately 30 minutes of work on a single CPU. While it is easy to parallelize this procedure using a cluster of computer, it is beneficial to further reduce the time of mining. Reducing the number of negatives is one easy way to speed-up the training procedure; therefore, in this section we ask the natural question: how does object detection performance depend on the size of the negative set? We perform this evaluation on 6 VOC categories: bus, cow, diningtable, motorbike, sheep, and train while considering different negative set sizes: 25, 250, 1000, and the original 2500. Figure 3.8 shows the summarized results for the three methods: ESVM, ESVM+Cal, and ESVM+Co. Detailed numbers can be seen in Table 3.3, Table 3.4, Table 3.5, and Table 3.6. We note that increasing the negative set size from 25 to 250 negatives images gives us a much higher boost than increasing it from 1000 to the full 2500 images. This suggests that once each exemplar has seen a sufficient number of negatives, its performance will not significantly improve with more negatives.

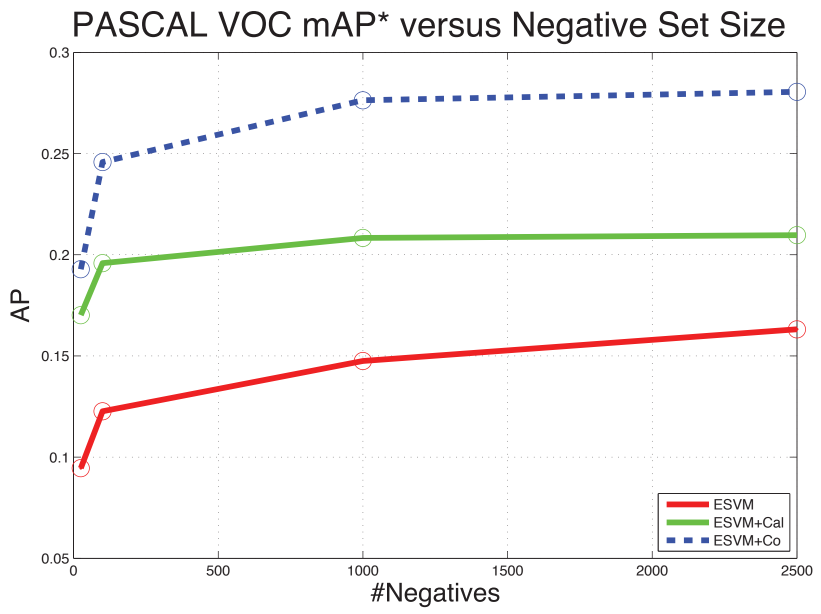


Figure 3.8: **Varying negative set sizes.** We compute the performance of the Exemplar-SVM algorithm as a mAP over 6 PASCAL categories: bus, cow, diningtable, motorbike, sheep, and train. The x-axis indicates the number of negative images used and the y-axis is the PASCAL VOC 2007 resulting mAP score for each of the 3 methods.

Type	ESVM	ESVM+B	ESVM+M
bus	0.103	0.269	0.248
cow	0.100	0.122	0.149
diningtable	0.001	0.048	0.096
motorbike	0.164	0.232	0.253
sheep	0.098	0.154	0.164
train	0.101	0.196	0.256
mAP	0.094	0.170	0.194

Table 3.3: Exemplar-SVM results with 25 negatives.

Type	ESVM	ESVM+B	ESVM+M
bus	0.185	0.296	0.348
cow	0.121	0.156	0.164
diningtable	0.005	0.018	0.112
motorbike	0.248	0.284	0.338
sheep	0.053	0.163	0.198
train	0.124	0.258	0.315
mAP	0.123	0.196	0.246

Table 3.4: Exemplar-SVM results with 250 negatives.

Type	ESVM	ESVM+B	ESVM+M
bus	0.216	0.304	0.384
cow	0.130	0.148	0.178
diningtable	0.012	0.019	0.118
motorbike	0.269	0.302	0.377
sheep	0.074	0.172	0.239
train	0.184	0.305	0.361
mAP	0.147	0.208	0.276

Table 3.5: Exemplar-SVM results with 1000 negatives

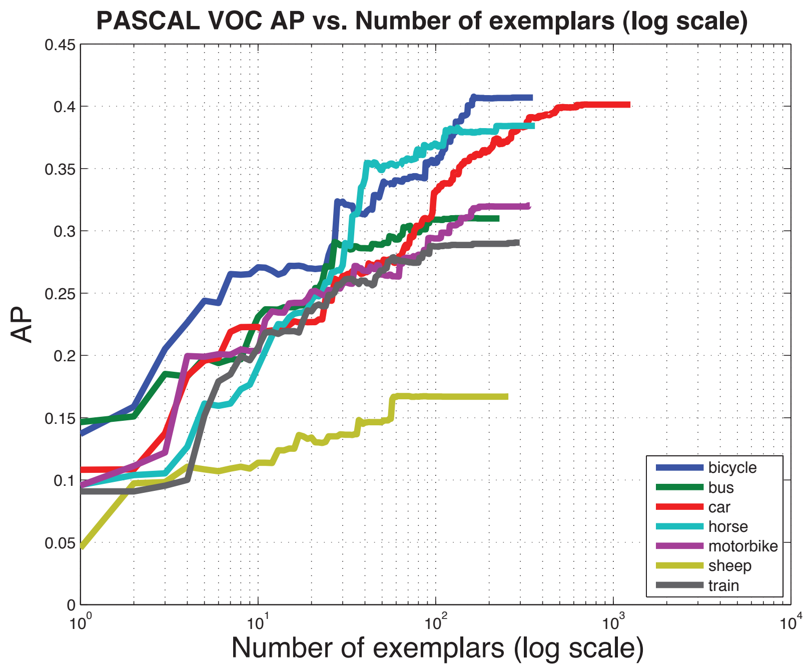
Type	ESVM	ESVM+B	ESVM+M
bus	0.236	0.310	0.397
cow	0.118	0.147	0.186
diningtable	0.016	0.023	0.111
motorbike	0.287	0.320	0.394
sheep	0.117	0.167	0.226
train	0.205	0.291	0.369
mAP	0.163	0.210	0.281

Table 3.6: Exemplar-SVM results with 2500 negatives. This is the same as the main result in the main Table 3.1

3.5.3 Dependence On the Number of Exemplars

Since the complexity of our Ensemble of Exemplar-SVMs scales linearly with the number of exemplars, it is valuable to study the object detection performance of ensembles of varying sizes. We ask: it possible to obtain really good object detection results with a small number of exemplars? For each object category in VOC, we train all of the exemplars using our full approach, and then sort them using the output of calibration (see Section refsec:calibration). Calibration scores each exemplar independently and thus the exemplars which produce the largest number of good matches (on held-out validation data) will come first in the sorting. Given this ordering of exemplars, we consider ensembles of increasing size, starting with the single best exemplar, and incrementally adding the next-best one, until we have all exemplars in the ensemble.

We report the PASCAL VOC performance for varying numbers of exemplars using the per-exemplar calibration strategy, and not the full co-occurrence matrix. The PASCAL VOC AP versus number of exemplars plots can be seen in Figure 3.9. We show the results in two different ways: AP vs. log number of exemplars, and AP vs. percent of exemplars in the category. We note that this curve is for the most part monotonically increasing. As can be seen from this curve, the performance saturates for these categories when about half the exemplars are chosen. The flat tail of each curve suggests that adding the last half of exemplars (many of which are bad) doesn't make our performance drop. This suggests that our calibration step works well at suppressing bad exemplars. Since the run-time complexity is directly proportional to the number of exemplars in the ensemble, this result suggests that we can get much faster object category detectors by constructing an ensemble with no more than half the number of exemplars in each category.



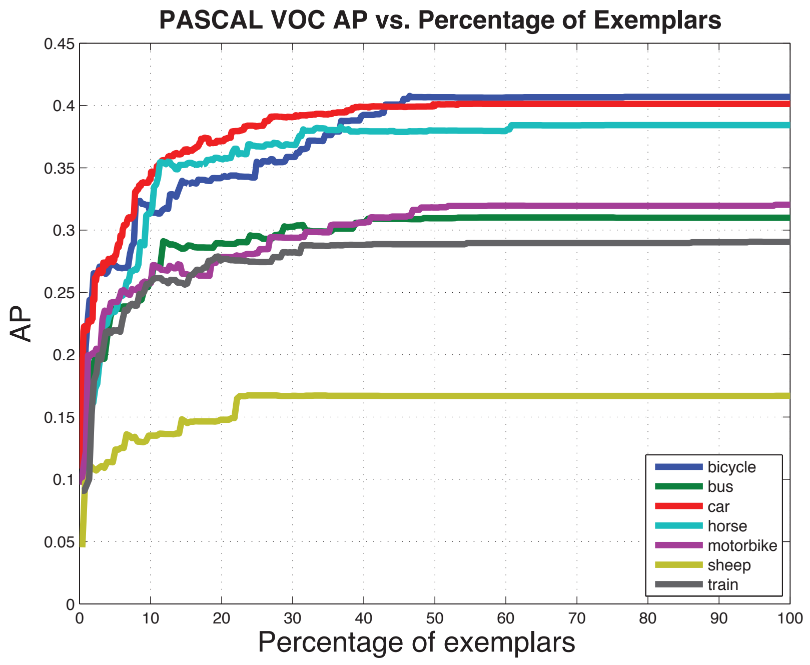


Figure 3.9: **PASCAL VOC Results vs. Number of Exemplars.** We report the VOC AP for ensembles of increasing size. For each category, a sorted list is created from the output of calibration, and ensembles of size 1 through N (N is the number of exemplars in a category) are evaluated on the detection task. The top plot shows performance versus the raw number of exemplars. The bottom plot shows the same information, but using a linear scale, and expressed as a percent of the total number of exemplars within the category.

3.5.4 The Role of Feature Dimensionality

Another important design decision in our Exemplar-SVM framework is the dimensionality of features representing each exemplar. In this section, we vary the feature dimension by varying each exemplar's HOG template size. We repeat the PASCAL VOC experiments with larger-sized templates as well as with smaller-sized templates. Our default template size is roughly 100 cells, with the large templates being roughly 200 cells, small templates being roughly 50 cells, and tiny ones with 25 cells. The HOG initialization/framing procedure is fully described in Appendix B.2. A summary of the results averaged across 6 categories can be seen in Figure 3.10. The full result tables are in Table 3.7, Table 3.8, Table 3.9, and Table 3.10. We note that larger templates slightly improve results for the raw ESVM method and ESVM+Cal; however, a template size of 100 is the best when using the co-occurrence matrix. However, the advantage of using small templates over larger ones is that training small templates as well as evaluating them on test-images is in general much faster.

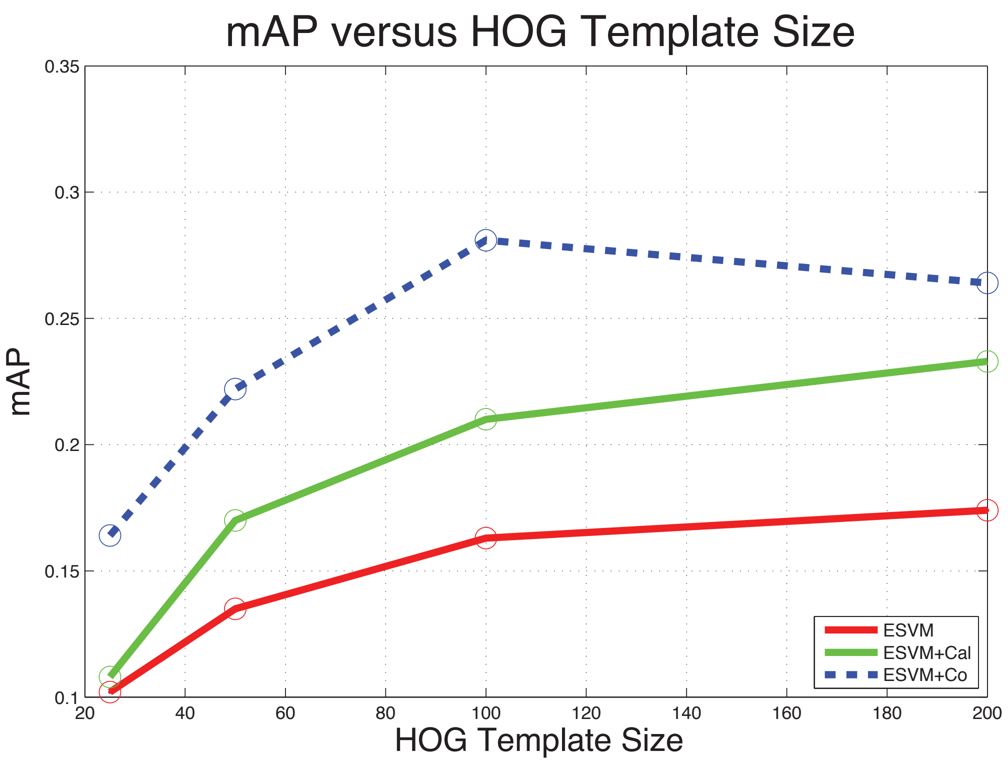


Figure 3.10: **PASCAL VOC Results for Varying HOG Template Sizes.** We compute the performance of the Exemplar-SVM algorithm as a mAP over 6 PASCAL categories: bus, cow, diningtable, motorbike, sheep, and train. The x-axis indicates the size of the HOG templates used and the y-axis is the PASCAL VOC 2007 resulting mAP score for each of the 3 methods.

Type	ESVM	ESVM+Cal	ESVM+Co
bus	0.100	0.125	0.184
cow	0.137	0.129	0.218
diningtable	0.001	0.001	0.006
motorbike	0.180	0.176	0.256
sheep	0.103	0.120	0.204
train	0.093	0.096	0.114
mAP	0.102	0.108	0.164

Table 3.7: Exemplar-SVM results with tiny HOG templates (goalsize: 25, maxdim: 5).

Type	ESVM	ESVM+Cal	ESVM+Co
bus	0.172	0.215	0.316
cow	0.149	0.166	0.223
diningtable	0.007	0.093	0.053
motorbike	0.259	0.269	0.376
sheep	0.113	0.157	0.232
train	0.111	0.117	0.133
mAP	0.135	0.170	0.222

Table 3.8: Exemplar-SVM results with smaller HOG templates (goalsize: 50, maxdim: 8).

Type	ESVM	ESVM+Cal	ESVM+Co
bus	0.236	0.310	0.397
cow	0.118	0.147	0.186
diningtable	0.016	0.023	0.111
motorbike	0.287	0.320	0.394
sheep	0.117	0.167	0.226
train	0.205	0.291	0.369
mAP	0.163	0.210	0.281

Table 3.9: Exemplar-SVM results with default-sized HOG templates (goalsize: 100, maxdim: 12).

Type	ESVM	ESVM+Cal	ESVM+Co
bus	0.232	0.352	0.394
cow	0.132	0.155	0.164
diningtable	0.093	0.096	0.102
motorbike	0.308	0.310	0.373
sheep	0.071	0.180	0.217
train	0.206	0.308	0.337
mAP	0.174	0.233	0.264

Table 3.10: Exemplar-SVM results with large HOG templates (goalsize: 200, maxdim: 15).

Part II: Beyond Object Detection

It affords an immediate step, however, to associative indexing, the basic idea of which is a provision whereby any item may be caused at will to select immediately and automatically another. This is the essential feature of the memex. The process of tying two items together is the important thing.

Vannevar Bush, As We May Think (1945)

We have, up until now, devoted our attention to creating visual associations, or in the words of Vannevar Bush, “tying two items together.” We demonstrated that the visual associations produced by the Exemplar-SVM algorithm can be used for object category detection, much like other category-based detection systems [Dalal and Triggs, 2005, Felzenszwalb et al., 2010, Chum and Zisserman, 2007]. However, the central benefit of our approach is that we are able to establish a direct link between a detection and a single training exemplars. In Chapter 4, we showcase the quality of our visual associations and demonstrate that the alignment between most detections and their associated exemplar is good enough to let us transfer any available exemplar meta-data (e.g., segmentation, geometry, 3D model) directly onto the detection. We perform an evaluation on the PASCAL VOC 2007 dataset, consider the three transfer-tasks (i.e., segmentation transfer, geometry transfer, and related object priming), and visualize the exemplar-exemplar similarity structure as a graph. Our meta-data transfer goes beyond what is required in most object detection tasks (i.e., a category-labeled bounding box), and the resulting interpretations (e.g., object pose, object geometry, object attributes) can then be used as part of overall scene understanding.

In Chapter 5 we consider a contextual object prediction task [Malisiewicz and Efros, 2009] which we tackle by forming a graph over exemplars and their spatial relationships, which we call the *Visual Memex Model*. We apply our “category-free” Visual Memex Model to a variant of Antonio Torralba’s Context Challenge, where the goal is to predict the appearance of a hidden object solely based on the hidden region’s spatial relationship to a set of visible objects in the scene. We evaluate our Visual Memex Model on the Context Challenge task and show an improvement over category-based baselines.

Chapter 4

Association and Meta-data Transfer

We have already shown that Exemplar-SVMs are highly-tuned to the appearance of the source exemplar and that we are able to obtain high quality correspondences between exemplars and detection windows. Such high quality alignment means that Exemplar-SVMs can be used for more than bounding box prediction — any meta-data associated with exemplars can be transferred onto the detection window. The hallmark of our approach is that we do not require a complex image alignment process (unlike [Berg et al., 2005, Liu et al., 2009]) — the exemplar-specific detector already does all the work! We consider meta-data in the form of a pixel-wise labeling such as a segmentation or a geometric labeling and we simply transfer the exemplar-aligned meta-data onto the detection using the transformation estimated between the source exemplar and the detection window (a scalar scale and a 2D translation).

In this chapter we present results on applying an ensemble of Exemplar-SVMs (as defined in Chapter 3) to a set of meta-data transfer tasks: segmentation transfer, geometry transfer, 3D model transfer, as well as related object priming. For the transfer applications we use the Exemplar-SVM method with calibration because even though using the exemplar co-occurence matrix boosts object detection performance, it uses

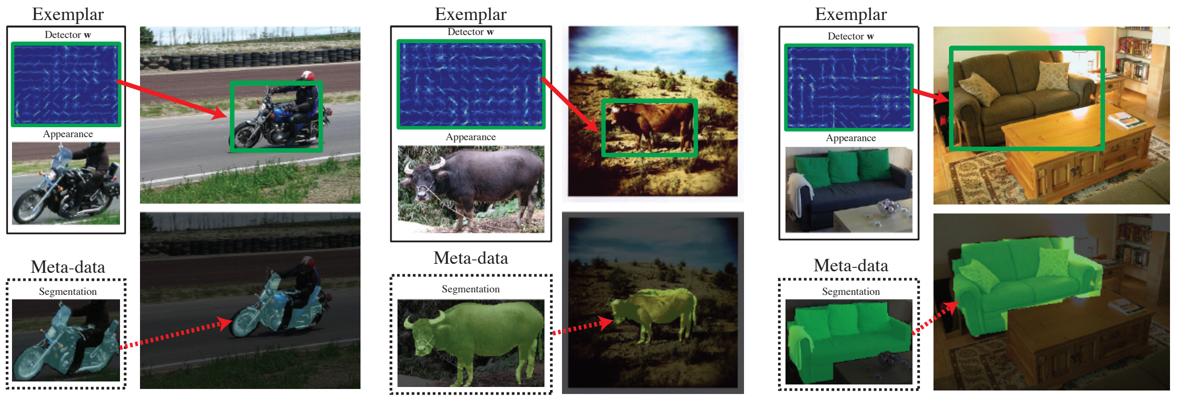


Figure 4.1: **Segmentation Transfer.** Object segmentations are transferred from the exemplar directly onto the detection window.

multiple overlapping exemplars to score windows (at the cost of forgetting which single association is the best). Calibration produces much higher quality alignments because detections are scored independently.

4.1 Segmentation and Geometry Transfer

For the task of segmentation, the goal is to estimate which pixels belong to a given object and which do not. Figure 4.1 shows some qualitative segmentation transfer examples on a wide variety of object classes. For quantitative evaluation, we asked labelers to segment and geometrically annotate all of the instances in the "bus" category in the PASCAL VOC 2007 dataset. For the segmentation task, our method performs at a pixelwise accuracy of 90.6%. For geometry estimation, the goal is to assign labels to pixels indicating membership to one of 3 "left," "front," and "right" dominant orientation classes [Hoiem et al., 2005]. We compare our Exemplar-SVM system against two baselines: (a) Hoiem's pre-trained generic geometric class estimation algorithm [Hoiem et al., 2005]; (b) Using [Felzenszwalb et al., 2010] to detect objects followed by simple NN to create associations. We obtain a 62.3% pixelwise labeling accuracy using our Exemplar-SVM approach as compared to the 43.0% obtained using [Hoiem et al., 2005]

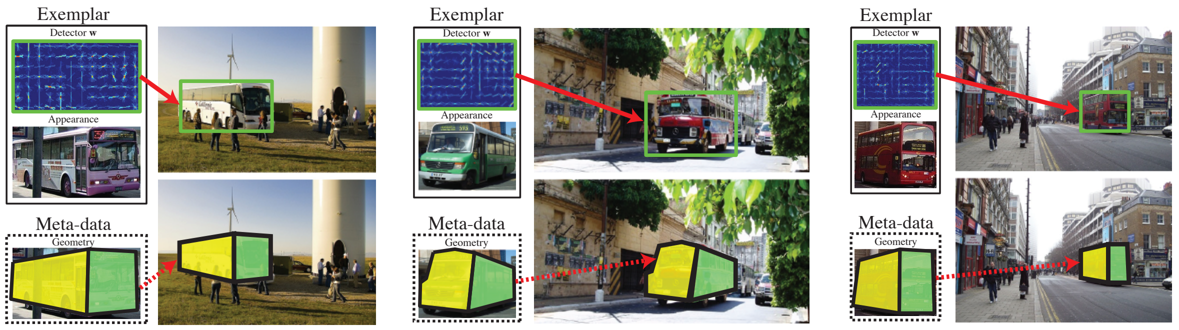


Figure 4.2: **Qualitative Geometry Transfer.** We transfer geometric labeling from bus exemplars onto corresponding detections.

and 51.0% using [Felzenszwalb et al., 2010]+NN. This clearly shows that while our transfer is simple, it is definitely not trivial as it relies on obtaining strong alignment between the exemplar and the detection (see qualitative results in Figure 4.2). Global methods fail to generate such alignments, leading to much lower performance.

4.2 3D Model Transfer

We annotated a subset of chair exemplars with $3D$ models from Google's $3D$ Warehouse (and aligned with Google Sketch-Up $3D$ model-to-image alignment tool). Given a single exemplar, labelers were asked to find the most visually similar model in the $3D$ Warehouse for that instance and perform the alignment. Due to the high quality of our automatically-generated associations, we were able to simply transfer the exemplar-aligned $3D$ model directly onto the detection window without any additional alignment, see Figure 4.3.

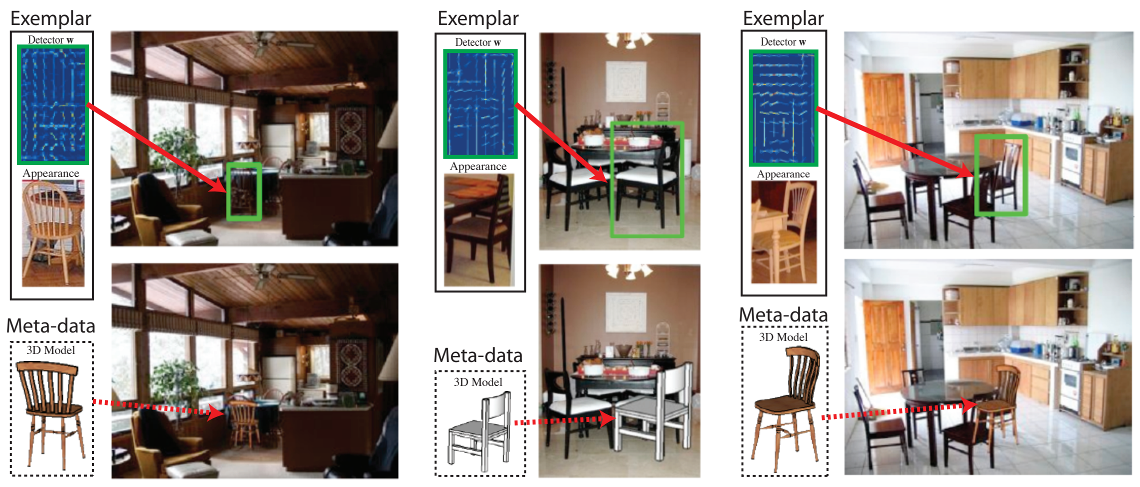


Figure 4.3: **3D Model transfer.** In each of these 3 examples, the green box in the top image shows the detection window, and the bottom shows the automatically transferred $3D$ model.

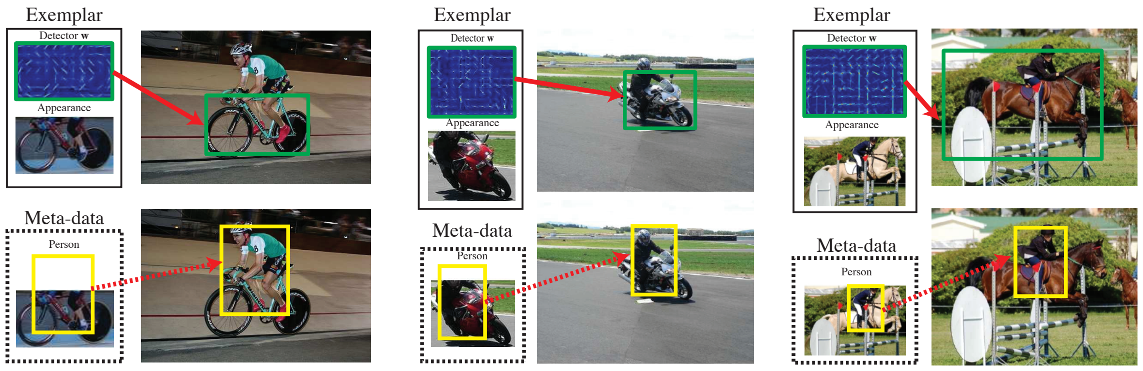


Figure 4.4: **Related Object Priming.** A bicycle/motorbike/horse exemplar is used to predict bounding box for “person”.

4.3 Related Object Priming

Exemplars often show an interplay of multiple objects, thus any other objects which sufficiently overlap with the exemplar can be viewed as additional meta-data belonging to the exemplar. This suggests using detectors of one category to help “prime” objects of another category. We look at the following task: predicting a bounding box for “person” given a detection of category X, where X is either a horse, motorbike, or bicycle (see Figure 4.4 for qualitative results). We say that a person is “riding” an X if its overlap with the X bounding box sufficient ($OS > .1$). We quantitatively evaluated the person prediction performance and compared against a baseline which predicts a person presence based on majority voting. Our method considerably outperforms the baseline (72.46% as compared to 58.67% for the baseline), suggesting that our exemplar associations provide good alignment of exemplars as well as their related objects. The results can be seen in Table 4.1.

Category	Majority Voting	[Malisiewicz et al., 2011]
bicycle	63.4%	**72.8%**
motorbike	50.0%	**67.4%**
horse	62.6%	**77.2%**

Table 4.1: **Is there a person riding this horse?** We predict from our bicycle, motorbike, and horse detectors whether there is a person riding the object. Our approach is better than the majority vote baseline, suggesting that exemplars are useful at predicting nearby, related objects.

4.4 Exemplar In-painting

When the dataset does not come equipped with any meta-data beyond object labels and bounding boxes, it is not possible to perform any kind of transfer. In such a case, we create an "interpretation" via a simple object in-painting using the exemplar's raw appearance, as shown Figure 4.5 and Figure 4.6. In each example we show the source exemplar along with its learned template $\mathbf{w}_e$ on the left, and the detection window and exemplar in-painting on the right. Note how our exemplar in-painting results showcase the quality of our automatic associations, yet again. Some failures of our in-painting can be seen in Figure 4.7 and Figure 4.8. Note that quite often when we produce a failure (defined by category mismatch), are mistakes are often between semantically relevant categories (e.g., a cow mistaken as a sheep, or a horse mistaken as a dog). Even when we estimate the wrong object category, we can still often predict the object's pose quite well.

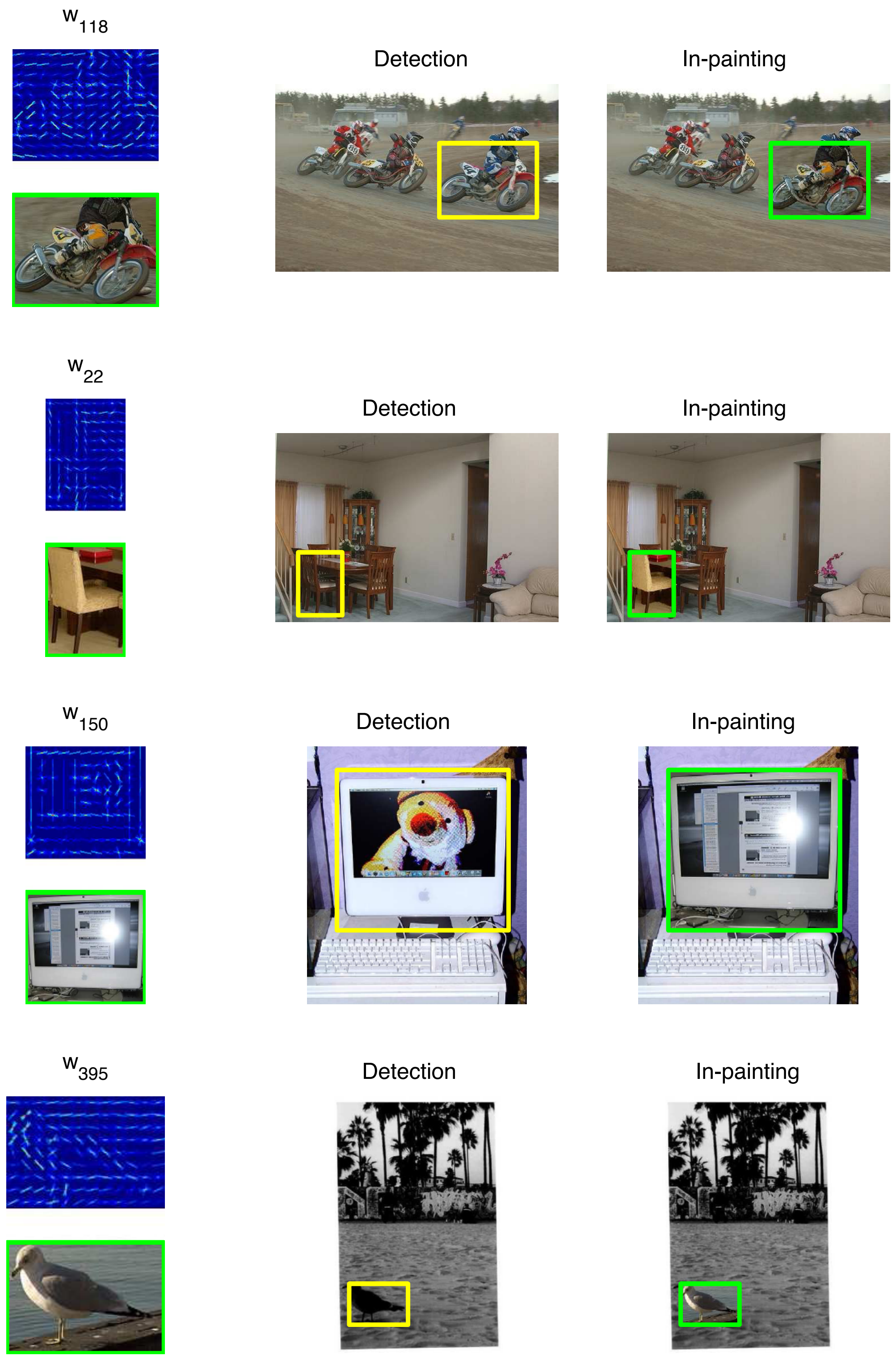


Figure 4.5: **Successful Exemplar Transfers 1**. We some of the top exemplar transfers from the PASCAL VOC 2007 testset. Each example shows: learned exemplar template $\mathbf{w}_e$ and exemplar appearance, detection window in test image, exemplar appearance overlay.

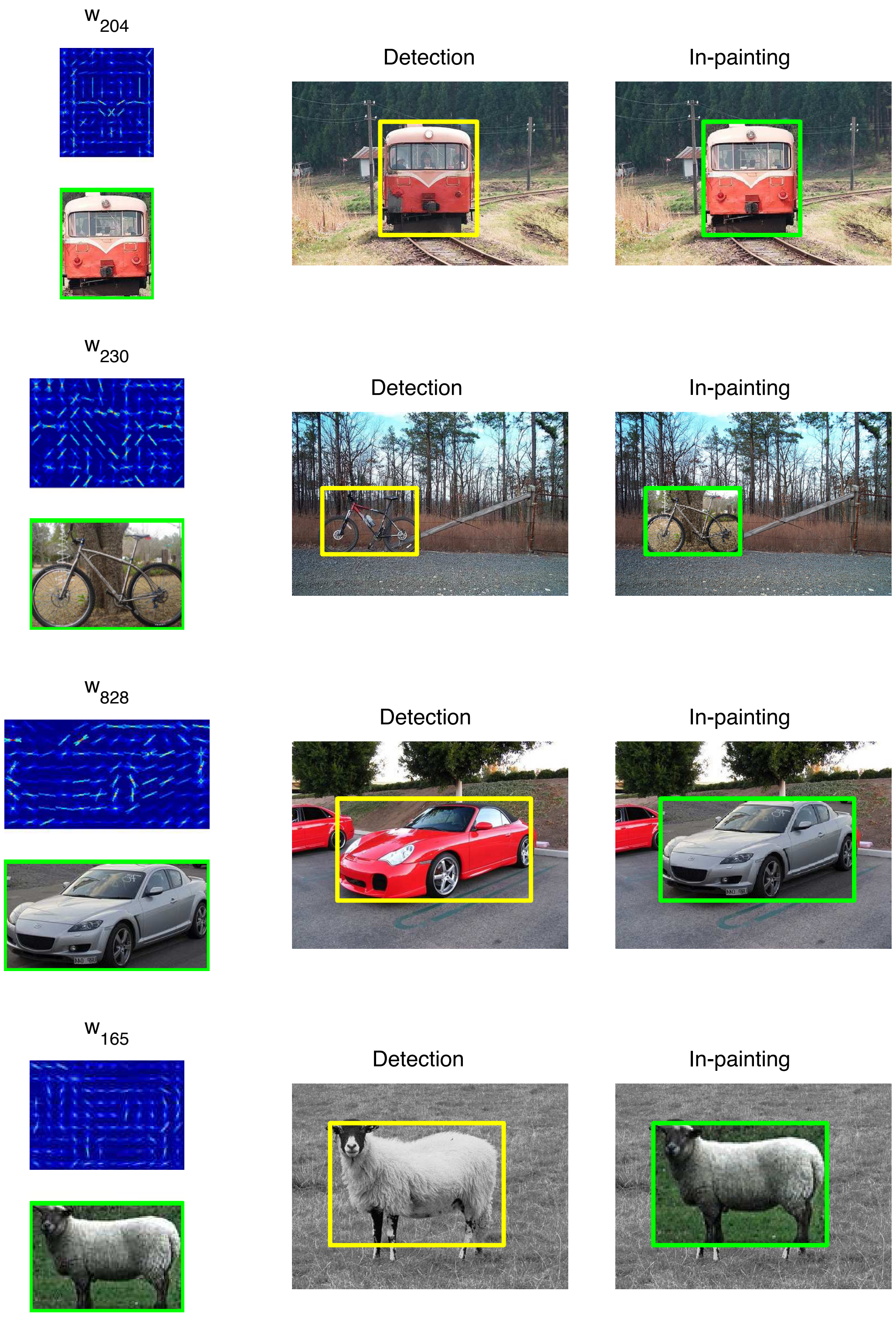


Figure 4.6: **Successful Exemplar Transfers 2**. We some of the top exemplar transfers from the PASCAL VOC 2007 testset. Each example shows: learned exemplar template $\mathbf{w}_e$ and exemplar appearance, detection window in test image, exemplar appearance overlay.

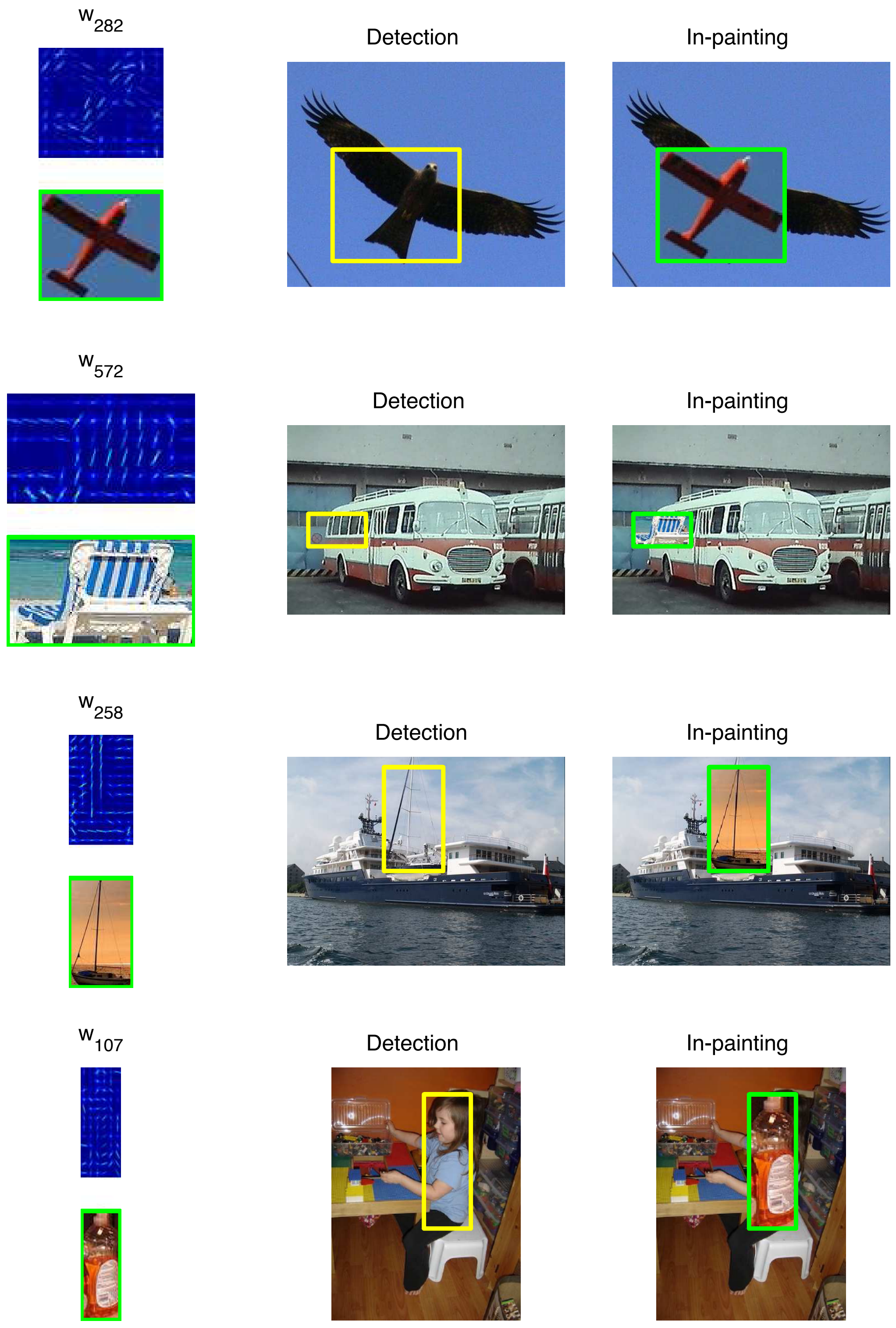


Figure 4.7: **Exemplar Transfer Failures 1**. We some of the top failed exemplar transfers from the PASCAL VOC 2007 testset. Each example shows: learned exemplar template $\mathbf{w}_e$ and exemplar appearance, detection window in test image, exemplar appearance overlay.

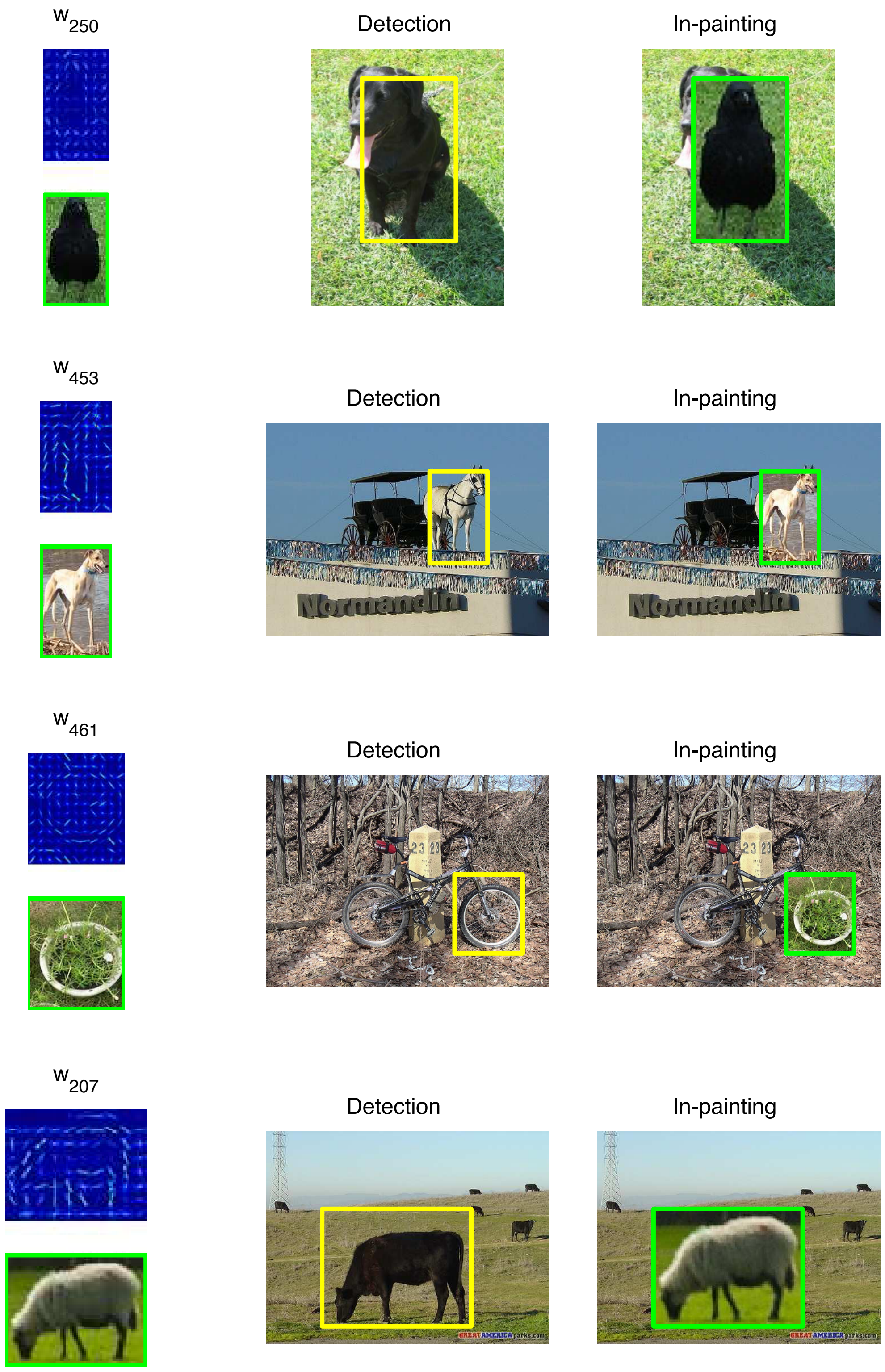


Figure 4.8: **Exemplar Transfer Failures 2**. We some of the top failed exemplar transfers from the PASCAL VOC 2007 testset. Each example shows: learned exemplar template $\mathbf{w}_e$ and exemplar appearance, detection window in test image, exemplar appearance overlay.

4.5 Visualizing Exemplar-Exemplar Associations

"...we see a complicated network of similarities overlapping and criss-crossing"

Ludwig Wittgenstein

In this section, we create a visualization of the exemplar-exemplar visual association structure induced by the Exemplar-SVM algorithm on the PASCAL VOC 2007 dataset. Our visualization is a sort of "Visual Memex" graph, which we visualize using the popular open source graph visualization library Graphviz [Ellson et al., 2001]. We also zoom into portions of this graph to analyze the "visual trail" similarity structure around a particular exemplar.

While the concept of a network of overlapping similarities comes from Wittgenstein's family resemblances [Wittgenstein, 1953], the idea of a visual trail is directly taken from Bush's "As We May Think." The advantage of thinking about exemplars in terms of visual trails is that a trail allows for a single object instance to participate in *multiple* visual trails. To create a graph, we connect two exemplars with an edge if both exemplars fire on each other with a calibrated score greater than 0.7 and if their bounding box overlap is greater than 0.5. While this is a rather strict requirement for creating exemplar-exemplar edges, weaker constraints create graphs which are hard to visualize in two dimensions. For the sake of clarity, we chose these parameters even though they do not always result in a connected graph. We display graphs for several PASCAL VOC object categories using the Exemplar-SVM algorithm. These "Visual Memexes" can be seen in the following pages: train (see Figure 4.9), motorbike (see Figure 4.10), and aeroplane (see Figure 4.11).

It is also important to note that many of the ideas behind Bush's memex are slowly starting to appear in computer vision. One highly relevant approach is the Photobios algorithm [Kemelmacher-Shlizerman et al., 2011] for navigating a large collection of images of the same person. By creating links between visually similar photos of the same person's face, it is possible to render a movie which summarizes the visual evolution of a single person, even creating compelling animations of the face as it transitions from different states (e.g., a movie which transitions from a person smiling to the same person frowning). While this approach has made its way into the popular Face Movies feature of Google's Picasa photo-sharing software, it is important to realize that it draws its success from the maturity of face detection and face recognition technology. The key challenge is in extending these ideas to non-face domains.

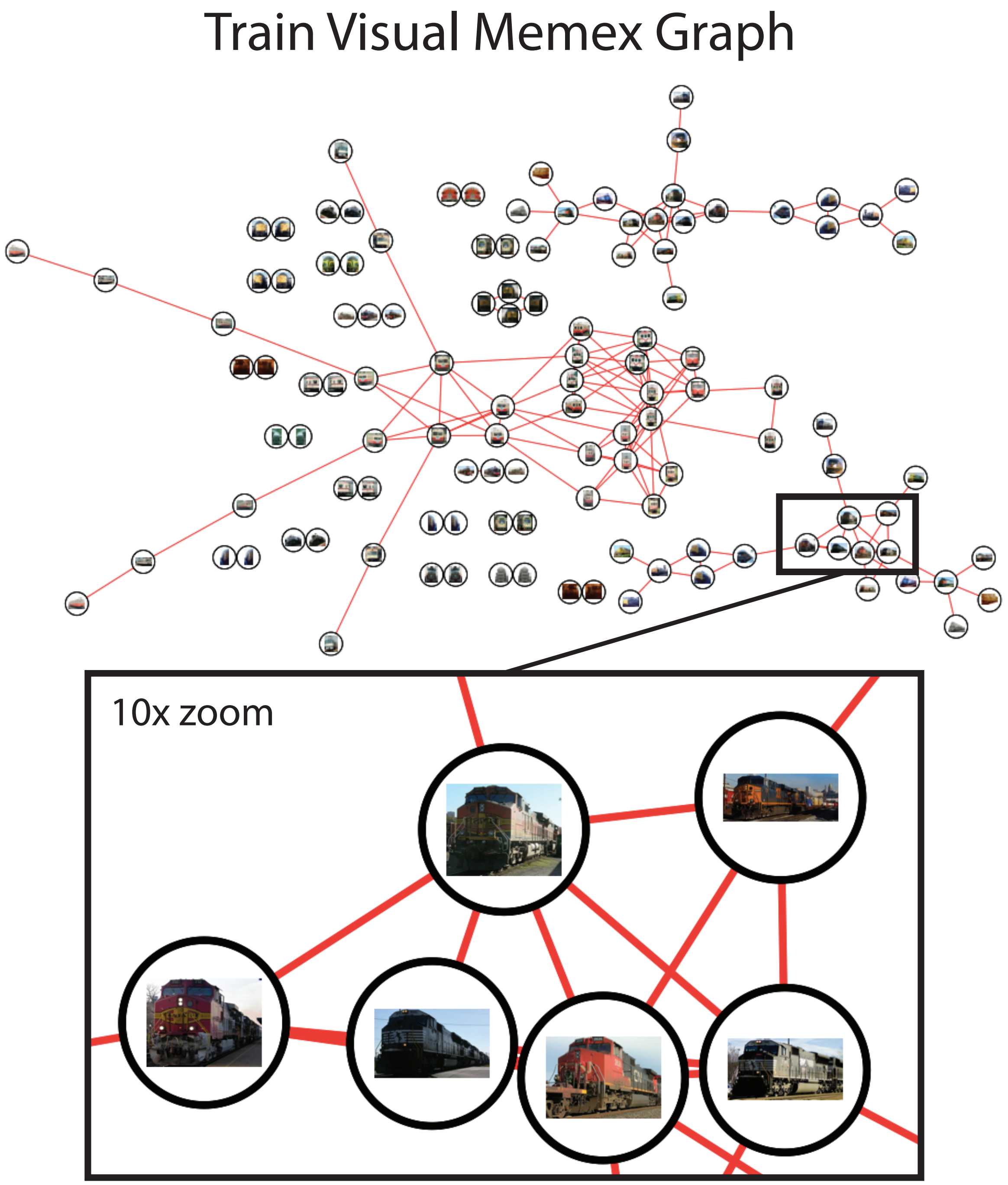


Figure 4.9: **Train Concept Visual Memex.** The similarity graph is created by applying each Exemplar-SVM to in-class images.

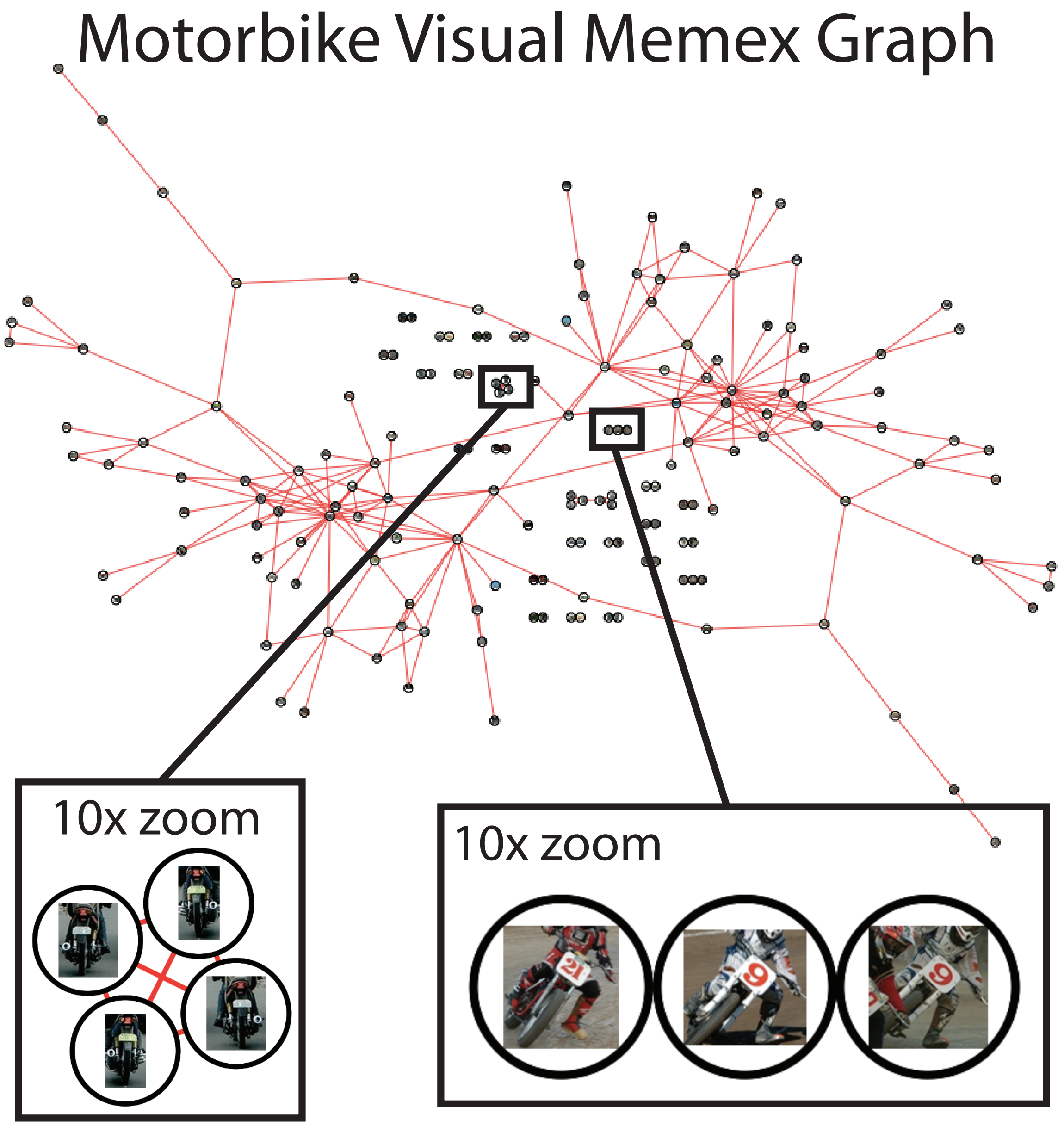


Figure 4.10: **Motorbike Concept Visual Memex.** The similarity graph is created by applying each Exemplar-SVM to in-class images.

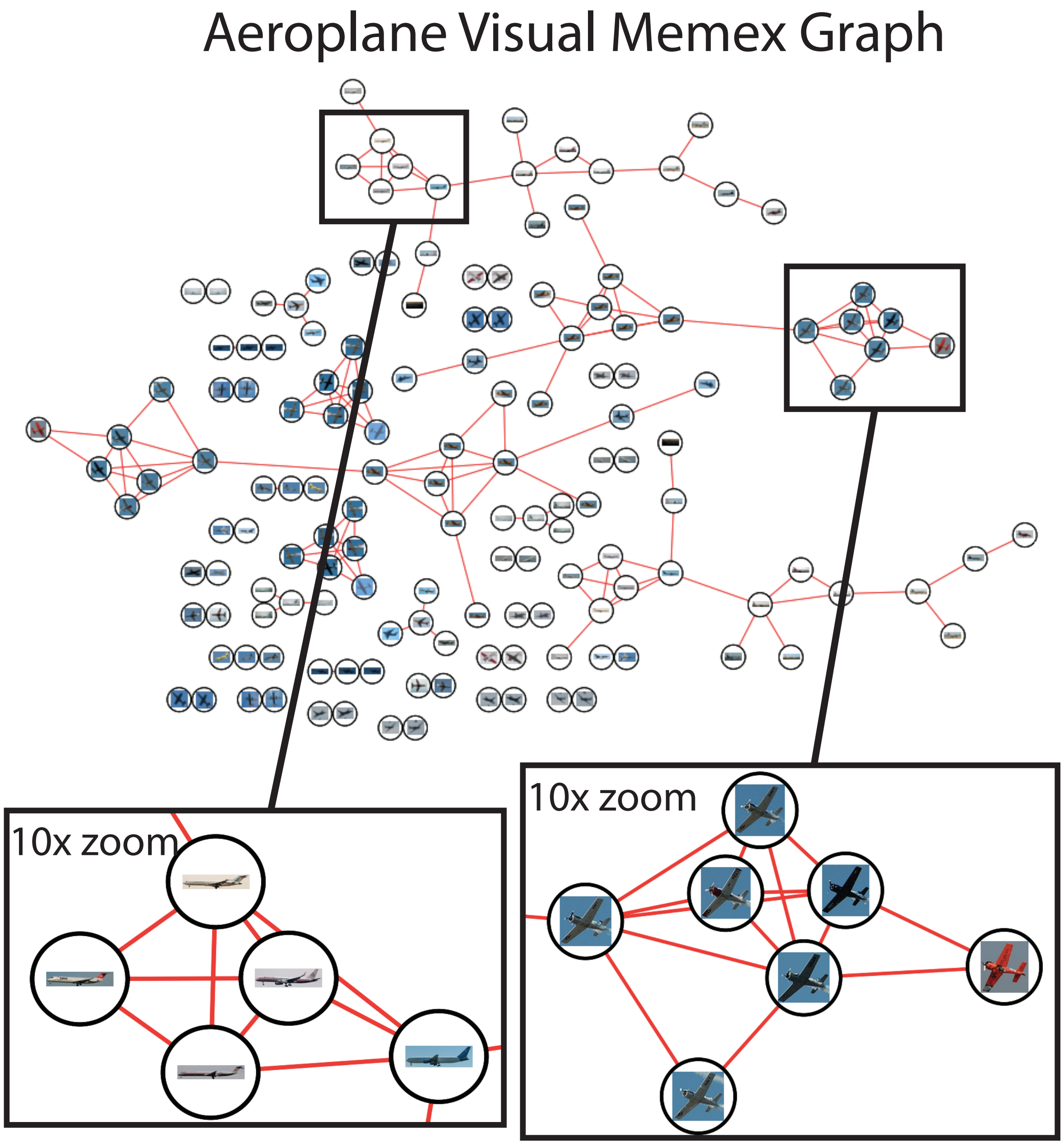


Figure 4.11: **Aeroplane Concept Visual Memex.** The similarity graph is created by applying each Exemplar-SVM to in-class images.

Chapter 5

Contextual Object Prediction

How far can you go before running an object detector?

Antonio Torralba

In real scenes composed of many different objects, the spatial configuration of one object can facilitate recognition of related objects [Bar and Ullman, 1996], and quite often ambiguities in recognition cannot be resolved without looking beyond the spatial extent of the object in question. Thus, algorithms which jointly recognize many objects at once by taking account of contextual relationships have been quite popular. While early systems relied on hand-coded rules for inter-object context (e.g., [Hanson and Riseman, 1978, Strat and Fischler, 1991]), more modern approaches typically perform inference in a probabilistic graphical model with respect to categories where object interactions are modeled as higher order potentials [He et al., 2004, Kumar and Hebert, 2005, Shotton et al., 2006, Rabinovich et al., 2007, Galleguillos et al., 2008, Parikh et al., 2008, Russell et al., 2007]. One important implicit assumption made by all such models is that interactions between object *instances* can be adequately modeled as relationships between human-defined object *categories*.

In this chapter we challenge this "category assumption" for object-object interactions and propose a novel category-free approach for modeling object relationships. We propose a new framework, the *Visual Memex Model*, for representing and reasoning about object identities and their contextual relationships in an exemplar-based, non-parametric way. We evaluate our model on Antonio Torralba's proposed Context Challenge [Torralba, 2003a] against a baseline category-based system.

Our starting point is Vannevar Bush's observation that strict categorical indexing of concepts has severe limitations [Bush, 1945] (see discussion in Chapter 1). Another motivation is Moshe Bar who believes that prediction plays a key role in the human brain [Bar, 2009]. Abandoning rigid object categories, we embrace Bush's and Bar's belief in the primary role of associations, but unlike Bush, we aim to discover these associations automatically from the data. At the core of our model is an exemplar-based representation of objects [Nosofsky, 1986, Malisiewicz and Efros, 2008]. The Visual Memex can then be thought of as a vast graph, with nodes representing all the object instances in the dataset, and arcs representing the different types of associations between them (Figure 5.1). There are two types of arcs in our model, encoding two different relationships between objects: 1) visual similarity (e.g. this car looks like that car), and 2) contextual associations (e.g. this car is next to this building).

Once the graph is built, it can be used to interpret a novel image (Figure 5.1, left) by first connecting segments within the image with similar exemplars, and then propagating contextual information between these exemplars through the graph. When an exemplar gets activated, visually similar exemplars as well as other contextually relevant objects get activated as well. This way, exemplar-to-exemplar similarity in the Memex graph can serve as Bush's "trails" to link concepts together in a non-parametric, query-dependent way, without the use of predefined categories. For example, in Figure 5.1, we should be able to infer that a car seen from the rear often co-occurs with an oblique building wall (but not a frontal wall) – something which category-based models would be hard-pressed

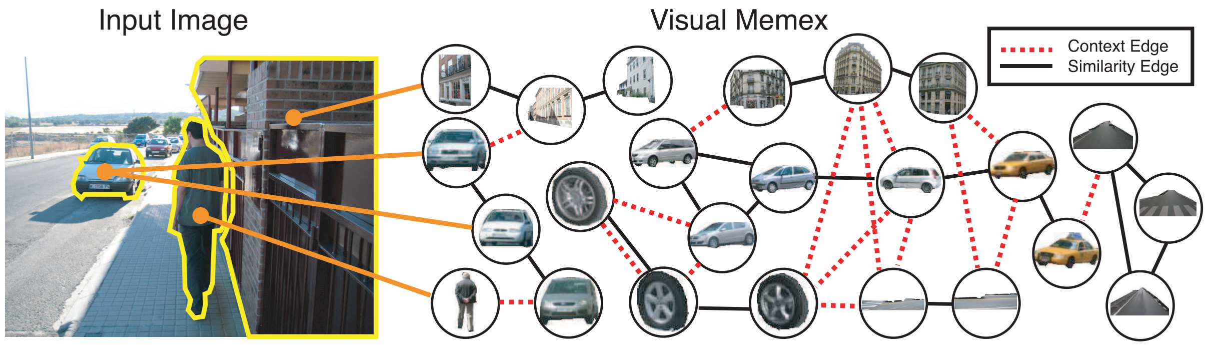


Figure 5.1: The **Visual Memex** graph encodes object similarity (solid black edge) and spatial context (dotted red edge) between pairs of object exemplars. A spatial context feature is stored for each context edge. The Memex graph can be used to interpret a new image by associating image segments with exemplars in the graph (orange edges) and propagating the information. *Figure best viewed in color.*

to achieve.

Formally, we define the Visual Memex Model as a graph $G = (V, E_S, E_C, \{D\}, \{\mathbf{f}\})$ consisting of N object exemplar nodes V, similarity edges E_S, context edges E_C, N per-exemplar similarity functions $\{D\}$, and the spatial features $\{\mathbf{f}\}$ associated with each context edge.

5.1 Building a LabelMe Visual Memex Graph

We extract a large database of exemplar objects and their ground-truth segmentation masks from the LabelMe [Russell et al., 2008] dataset and learn the structure of the Visual Memex in an offline setting. We use objects from the 30 most frequently occurring categories in LabelMe. Similarity edges are created using the per-exemplar distance function learning framework of [Malisiewicz and Efros, 2008], and context edges are created each time two exemplars are observed in the same image. We have a total of $N = 87,802$ exemplars in the Visual Memex, $|E_S| = 276,782$ similarity edges, and $|E_C| = 989,106$ context edges.

We use the per-exemplar distance-function learning algorithm of [Malisiewicz and

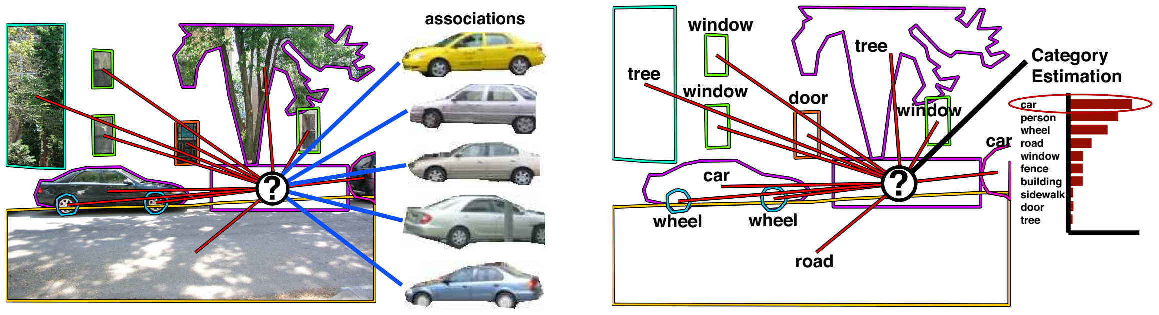


Figure 5.2: **Torralba's Context Challenge: "How far can you go without running a local object detector?"** The task is to reason about the identity of the hidden object (denoted by a "?") without local information. In our category-free Visual Memex model, object predictions are generated in the form of exemplar associations for the hidden object. In a category-based model, the category of the hidden object is directly estimated.

Efros, 2008], as described in Chapter 2, to learn the object similarity edges. We create a similarity edge between two exemplars if they are deemed similar by each others' distance functions. We use a fixed $\lambda = .00001$ and $\sigma = 100$ for all exemplars (see Equation 2.6).

When two objects occur inside a single image, we encode their 2-D spatial relationship into a context feature vector $\mathbf{f} \in \Re^{10}$ (visualized as red dotted edges in Figure 5.1). The context feature vector encodes relative overlap, relative displacement, relative scale, and relative height of the bottom-most pixel between two exemplar regions in a single image. This feature captures the spatial relationship between two regions and *does not take into account any appearance information* — it is a generalization of the spatial features used in [Galleguillos et al., 2008]. We measure the similarity between two context features using a Gaussian kernel: $K(\mathbf{f}, \mathbf{f}') = e^{-\alpha_1 ||\mathbf{f} - \mathbf{f}'||^2}$ with $\alpha_1 = 1.0$.

5.2 Torralba's *Context Challenge*

The intuition that we would like to evaluate is that many useful regularities of the visual world are lost when dealing solely with categories (e.g., the side view of a building should associate more with a side view of a car than a frontal view of a car). The key motivation behind the Visual Memex is that context should depend on the appearance of an object and not just the category it belongs to. In order to test this hypothesis against the commonly held practice of abstracting away appearance into categories, we need a rich evaluation dataset as well as a meaningful evaluation task.

We found that the *Context Challenge* [Torralba, 2003a] recently proposed by Antonio Torralba fits our needs perfectly. The evaluation task is inspired by the question: "How far can you go without running an object detector?" The goal is to recognize a single object in the image without peeking at pixels belonging to that object. Torralba presented an algorithm for predicting the category and scale of an object using only contextual information [Torralba, 2003b], but his notion of context is *scene-centered* (i.e, the appearance of the entire image is used for prediction). Since the context we wish study is object-centered, we use an object-centered formulation of the Context Challenge. While it is not clear if the absolute performance numbers on the Context Challenge are very meaningful in themselves, we feel that it is an ideal task for studying object-centered context and the role of categorization assumptions in such models.

In our variant of the Context Challenge, the goal is to predict the category of a hidden object y_i solely based on its spatial relationships to some provided objects — without using the pixels belonging to the hidden object at all. For our study, we use manually provided regions and category labels of K supporting objects inside a single image. We refer to the identities of the K supporting objects in the image as $\{y_1, \ldots, y_K\}$ (where $y \in \{1, \ldots, |C|\}$) and the set of K $2D$ spatial relationship features between each supporting object and the hidden object as $\{\mathbf{f}_{i1}, \ldots, \mathbf{f}_{iK}\}$.

5.3 Inference in the Visual Memex Model

In this section, we explain how to use the Visual Memex graph (automatically constructed from data) to perform inference for the Context Challenge hidden-object prediction task. Not making the "category assumption," the model is defined with respect to exemplar associations for the hidden object. Inference in the model returns a compatibility score between every exemplar and the hidden object, and can be though of as returning an ordered list of exemplar associations. Due to the nature of exemplar associations as opposed to category assignments, a supporting object can be associated with *multiple* exemplars as opposed to a *single* category. We create soft exemplar associations between each of the supporting objects and the exemplars in the Visual Memex using the similarity functions $\{D\}$ (see Section 5.1).

$\{S_1, \ldots, S_K\}$ are the appearance features for the K supporting objects. A_j^a is the affinity between exemplar a in the Visual Memex and the j-th supporting object and is created by evaluating S_j under a's distance function $A_j^a = e^{-D_a(S_j)}$. $\Psi(e_i, e_j, \mathbf{f}_{ij})$ is the pairwise compatibility between exemplar e_i and e_j under the spatial feature $\mathbf{f}_{ij}$. Let W_{ab} be the adjacency matrix representation of the similarity edges ($W_{uv} = [(u,v) \in E_S]$). Inference in the Visual Memex Model is done by optimizing the following conditional distribution which scores the assignment of an arbitrary exemplar e_i to the hidden object based on contextual relations:

$$p(e_i | A_1, \ldots, A_K, \mathbf{f}_{i1}, \ldots, \mathbf{f}_{iK}) \propto \prod_{j=1}^{K} \sum_{a=1}^{N} A_j^a \Psi(e_i, e_a, \mathbf{f}_{ij}) \tag{5.1}$$

$$\log \Psi(e_i, e_j, \mathbf{f}_{ij}) = \frac{\sum_{(u,v) \in E_C} W_{iu} W_{jv} K(\mathbf{f}_{ij}, \mathbf{f}_{uv})}{\sum_{(u,v) \in E_C} W_{iu} W_{jv}} \tag{5.2}$$

The reason for the summation inside Equation 5.2 is that it aggregates contextual interactions from similar exemplars. By doing this, we effectively "densify" the contextual interactions in the Visual Memex. An interpretation of this densification procedure is that we are creating a kernel density estimator for an arbitrary pair of exemplars (e_i, e_j) via a weighted sum of kernels placed at context features in the data set $\{\mathbf{f}_{uv}\} : (u, v) \in E_C$ where the weights $W_{iu}W_{jv}$ measure visual similarity between pairs (e_i, e_j) and (e_u, e_v) .

We experimented with using a single kernel, $\Psi(e_i, e_j | \mathbf{f}_{ij}) = K(\mathbf{f}_{ij}, \mathbf{f}_{e_i,e_j})$, and found that the integration of multiple features via the densification described above is a key ingredient for successful Visual Memex inference.

Finally, after performing inference in the Visual Memex Model, we are left with a score for each exemplar. At this stage, as far as our model is concerned, the recognition has already been performed. However, since the task we are evaluated on is category-based, we combine the returned exemplars into a vote for categories using Luce's Axiom of Choice [Medin and Schaffer, 1978] which averages the exemplar responses per-category.

CoLA-based Parametric Model

We would like to evaluate the Visual Memex model against a more traditional, category-based framework with parametric inter-category relationships. One of the most recent and successful approaches is the CoLA model [Galleguillos et al., 2008]. CoLA learns a set of parameters for each pair of categories which correspond to relative strengths of the four different top,above,below,inside spatial relationships. In the case of dealing with categories directly we consider a conditional distribution over the category of the hidden object y_i that factors as a star graph with K leaves (with the hidden object being connected to all the supporting objects). $\boldsymbol{\theta}$ are model parameters, Ψ is a pairwise potential that measures the compatibility of two categories with a specified spatial relationship, and Z is a normalization constant such that the conditional distribution sums to 1.

$$p(y_i|y_1,\ldots,y_K,\mathbf{f}_{i1},\ldots,\mathbf{f}_{iK},\boldsymbol{\theta}) = \frac{1}{Z}\prod_{j=1}^{K}\Psi(y_i,y_j,\mathbf{f}_{ij},\boldsymbol{\theta}) \tag{5.3}$$

Following [Galleguillos et al., 2008], we use a feature function $h(\mathbf{f})$ that computes the affinity between feature $\mathbf{f}$ and a set of prototypical spatial relationships. We automatically find P prototypical spatial relationships by clustering all spatial feature vectors $\{\mathbf{f}\}$ in the training set via the popular K-means algorithm. Let $h(\mathbf{f}) \in \Re^P$ be the normalized vector of affinities to cluster centers $\{\mathbf{c}_1,\ldots,\mathbf{c}_P\}$. $\boldsymbol{\theta}$ is the set of all parameters in this model, with $\boldsymbol{\theta}(y_i,y_j) \in \Re^P$ being the parameters associated with the pair of categories (y_i,y_j).

$$\log\Psi(y_i,y_j,\mathbf{f}_{ij},\boldsymbol{\theta}) = [h(\mathbf{f}_{ij})^T]\,\boldsymbol{\theta}(y_i,y_j) \tag{5.4}$$

$$h_i(\mathbf{f}) \propto e^{-\alpha||\,\mathbf{f}\,-\mathbf{c}_i||^2} \tag{5.5}$$

We tried using the four prototypical relationships corresponding to above, below, inside, and outside as in [Galleguillos et al., 2008], but found that using K-means with significantly larger number of prototypes $P = 30$ produced superior results. For learning $\boldsymbol{\theta}$, we found the maximum likelihood $\boldsymbol{\theta}$ using gradient descent. The training objective function was optimized to mimic what happens during testing on the Context Challenge task. Since the distributions for the Context Challenge task are defined with respect to a single category variable (see Equation 5.3), we could compute the partition function directly and did not require any approximations as in [Galleguillos et al., 2008] (which required training in a loopy graph).

Reduced KDE Memex Model

Since the Visual Memex Model and the CoLA-inspired model make different assumptions with respect to objects (category-based vs. exemplar-based) and context (parametric vs. nonparametric), we feel it would also be useful to examine a hybrid model, dubbed the Reduced KDE Memex Model, which uses a nonparametric model of context based on Kernel Density Estimation(KDE) but operates on object categories. The Reduced KDE Memex Model is created by collapsing all exemplars belonging to a single category into fully-connected components which can be thought of as adding categories into the Visual Memex graph. Identities between individual exemplars are lost, and thus we lose the fine details of a spatial context. By forming categories, we can no longer say a particular spatial relationship is between a blue side view of a car and an oblique brick building, we can only say it is a relationship between a car and a building. Now that we are left with an unordered bag of spatial relationships $\{\mathbf{f}\}$ between two categories, we need a way to measure compatibility between a newly observed $\mathbf{f}$ and the stored relationships.

We use the same form of the Context Challenge conditional distribution as in Equation 5.3. We use a Kernel Density Estimator(KDE) for every pair of categories, and the potential Ψ can be thought of as a matrix of such estimators. The use of nonparametric potentials in graphical models has been already explored in the domain of texture analysis [Paget and Longstaff, 1998]. δ_{ij} is the Kronecker delta function.

$$\log \Psi(y_i, y_j, \mathbf{f}_{ij}) = \frac{\sum_{(u,v)\in E_C} \delta_{y_i y_u} \delta_{y_j y_v} K(\mathbf{f}_{ij}, \mathbf{f}_{uv})}{\sum_{(u,v)\in E_C} \delta_{y_i y_u} \delta_{y_j y_v}} \tag{5.6}$$

The Reduced Memex model, being category-based and nonparametric, aggregates the spatial relationships across many different pairs of exemplars from two categories. While we used a fixed kernel K which measures distance isotropically across the dimensions

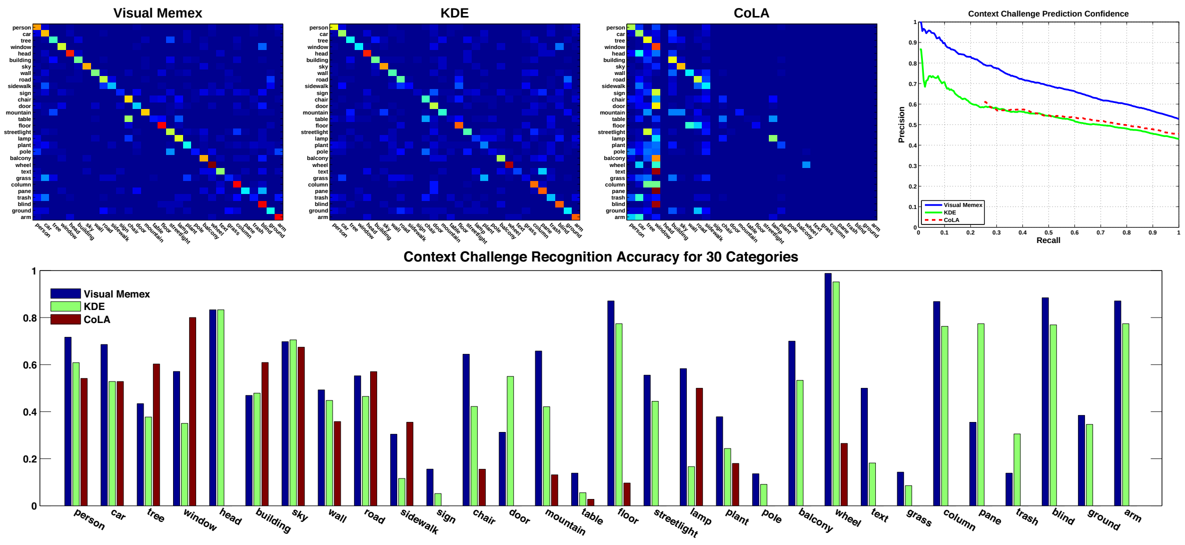


Figure 5.3: **Context Challenge Evaluation** a.) Context Challenge confusion matrices for the 3 methods: Visual Memex, KDE, and CoLA. b.) Recognition Precision versus Recall when thresholding output based on confidence. c) Side by side comparison of the 3 methods' accuracies for 30 categories.

of $\mathbf{f}$, the advantage of such a nonparametric approach is that with enough data the particularities of K do not matter. We also experimented with a Nearest Neighbor based model, but found the Kernel Density Estimation approach to be superior.

5.4 *Context Challenge* Evaluation: Visual Memex versus Categories

For the Context Challenge evaluation, we use 200 randomly selected densely labeled images from LabelMe [Russell et al., 2008]. Our testset contains 3048 total objects from 30 different categories. For an image with K objects, we solve K Context Challenge problems with one hidden object and K-1 supporting objects. Qualitative results on this prediction task can be seen in Figure 5.4.

We evaluate the performance of our Visual Memex model, the Reduced Memex KDE model, and the CoLA-inspired model with respect to categorization performance (confusion matrices can be seen in top left of Figure 5.3). The overall recognition

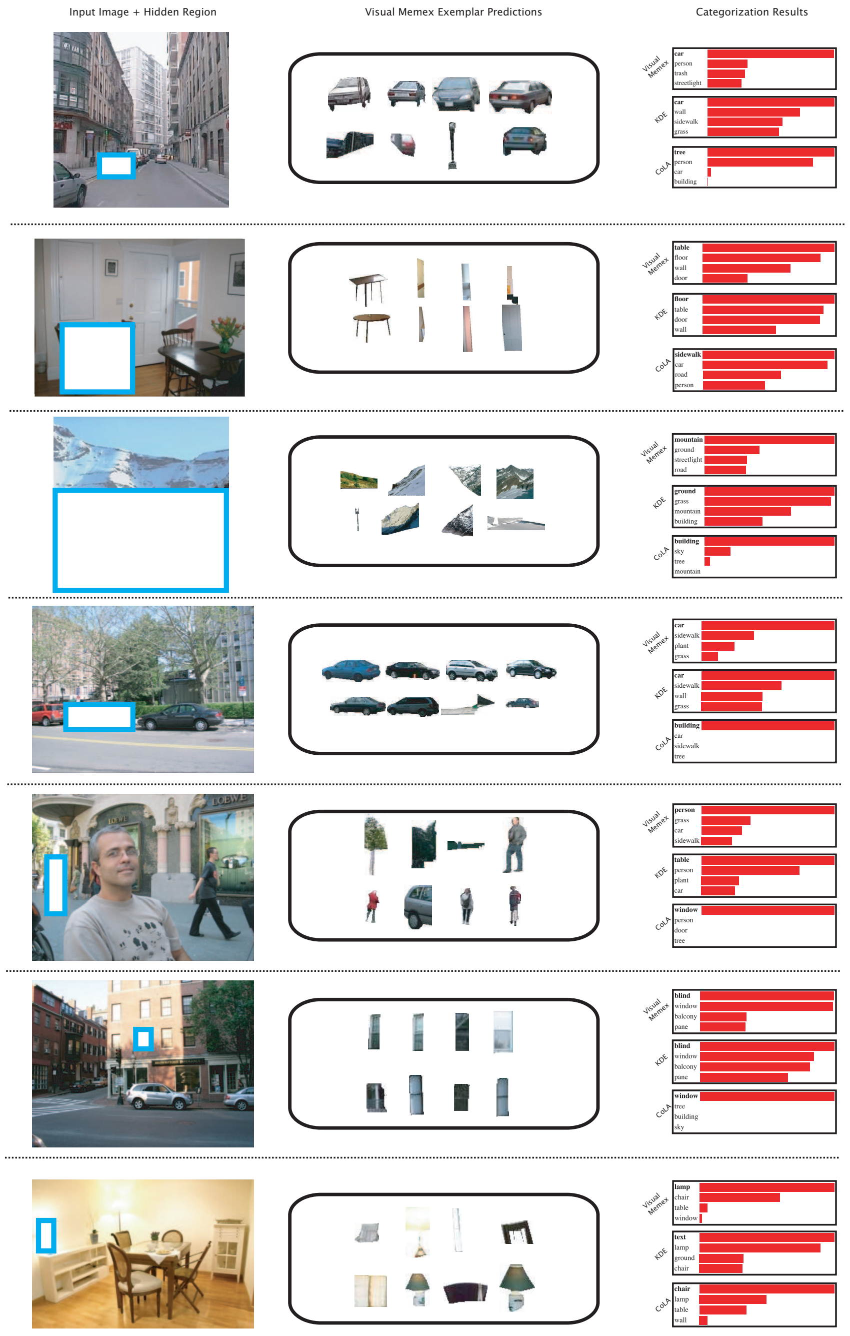


Figure 5.4: Qualitative Results on the Context Challenge. Exemplar predictions are from the Visual Memex model and categorization results are from the Visual Memex model, the KDE Model, and CoLA [Galleguillos et al., 2008].

accuracy of the Visual Memex Model, Reduced Memex Model, and CoLA are $.527$, $.430$, and $.457$ respectively. Note that the Visual Memex Model performs significantly better than the baselines. Taking a closer look at the per-category accuracies of the three methods (see bottom of Figure 5.3), we see that the CoLA-based method fails on many categories. The average per-category recognition accuracies of the three methods are: $.534$, $.454$, and $.213$. The Visual Memex Model still performs the best, but we see a significant drop in performance for the category-based CoLA method. CoLA is biased towards the popular categories, returning the most frequently occurring category (window) quite often. Overall, the Visual Memex Model achieves the best performance for 21 out of the 30 categories.

In addition, we plot precision recall curves for each of the three methods to determine if high confidence returned by each model is correlated with high recognition rates (top right of Figure 5.3). The Visual Memex model has the most significant high-precision low-recall regime, suggesting that its confidence is a good measure of success. The relatively flat curve for the CoLA method is related to the problem of overcompensation for popular classes as mentioned above. The distributions returned by CoLA tend to degenerate to a single non-zero value (most often on one of the popular categories such as window). This is why the maximum probability returned by CoLA is not a good measure of confidence.

We also demonstrate the power of the Visual Memex to predict *appearance* solely based on contextual interactions with other objects and their visual appearance. The middle row of Figure 5.4 demonstrates some of these associations. Note how in row 1, a plausible viewpoint is selected rather than just a random car. In row 3 we see that the appearance of snow on one mountain suggests that the other portion of the image also contains a snowy mountain. In summary, we presented a category-free Visual Memex Model and applied it to the task of contextual object recognition within the experimental framework of the *Context Challenge*. Our experiments confirm our intuition that moving beyond categories is beneficial for improved modeling of relationships between objects.

5.4.1 Follow-up Work

Since its initial publication in 2009, several interesting approaches have been published which are either influenced by our Visual Memex work, or are very relevant. Pietro Perona's effort to organize visual knowledge in the form of a *Visipedia* [Perona, 2010] is more akin to a visual encyclopedia than a visual web of concepts; however, it is nevertheless motivated by Vannevar Bush's ideas regarding the memex and thus highly relevant to our approach.

Regarding the use of object-graphs in vision, the work of [Fisher and Hanrahan, 2010] uses a similar category-free model but applies it to the problem of 3D model search when given the appearances of a small set of supporting 3D objects. [Lee and Grauman, 2010] have successfully used object graphs and graph-based descriptors for unsupervised object category discovery.

Chapter 6

Conclusion: Toward the Visual Memex

In his dissertation, we demonstrated the many advantages of posing the recognition problem as object *association* as opposed to the typical category-based object naming paradigm. We presented an exemplar-based object recognition framework, showcased the power of per-exemplar similarity measures for creating high-quality visual associations, as well as evaluated our ensemble of Exemplar-SVM approach on the popular PASCAL VOC object detection benchmark task. Not only does our method perform competitively with state-of-the-art methods, but it can be applied to meta-data transfer tasks which are hard to tackle with category-based models. Below we outline several key advantages of our framework over category-based approaches:

1. **Conceptual simplicity:** Instead of learning one complex category-specific model, we learn a separate simple template-based detector for each exemplar in our dataset. Each of our learning problems involves a single positive instance and is convex, thus allowing us to use efficient optimization libraries during learning.

2. **Meta-data transfer:** Our detections produce high quality alignments which allow us to transfer any meta-data associated with the exemplar directly onto the detection window, without an additional iterative alignment step. This shows

that once we cast the recognition problem in terms of association, we are able to produce much more than just a category-labeled bounding box. Problems such as segmentation and geometry transfer can be easily handled by our exemplar-based approach.

3. **Parallelizability/Easily extendable:** Because our exemplars are trained independently, it is possible to distribute learning across a cluster and then simply concatenate the resulting Exemplar-SVMs to create one large ensemble. Because of the minimal interaction across exemplars, we can “inject knowledge” into our ensemble of Exemplar-SVMs by augmenting it with additional exemplars, without having to re-train every exemplar.

4. **Image/object matching:** We have shown our framework can be used for matching objects as well as cross-domain image matching, suggesting that other applications, requiring their own notion of visual similarity, could benefit from our exemplar-based approach.

Given the power of exemplar-based visual associations, coupled with both increased computing speeds and improved access to large-scale computational resources, it is perhaps the right time for vision researchers to seriously consider Vannevar Bush’s ideas regarding a memory-based model of thought: “It operates by association. With one item in its grasp, it snaps instantly to the next that is suggested by the association of thoughts, in accordance with some intricate web of trails carried by the cells of the brain.” [Bush, 1945] Our work on the exemplar-based Visual Memex Model suggests that there is merit to the idea of an intricate web of visual trails. As we have shown in Chapter 5, it is possible to create such a visual network, with exemplars as nodes and exemplar relationships as edges. By applying our Visual Memex Model to a contextual object prediction task, we were able to predict the appearance of a hidden object solely based on its spatial relationships to a set of observed objects in the image. Thus embracing a memex-based way of thinking about the visual world entails thinking about visual knowledge as hyper-linked visual knowledge rather than as category-based knowledge. Such a category-free view of the world has the chance to change the way we address the problem of image understanding, much like the way Google and Wikipedia changed the way we think about knowledge.

Finally, there is growing evidence that the ideas regarding visual associations are here to stay. Given the role which analogies, associations and predictions play in cognitive neuroscience [Bar, 2009], the recent scientific findings which suggest that humans' visual memory is larger than previously thought [Brady et al., 2009], as well as the success of knowledge going digital (e.g., Wikipedia, Facebook, Google), we feel that the time is right to start embracing the power of visual associations. We hope that our thesis makes a strong case for exemplar-based representations and the strength of per-exemplar similarity functions — we hope that the success of our Exemplar-SVM framework and Visual Memex Model will motivate the next generation of researchers to question the necessity of relying on object categories for object/scene interpretation.

Appendix A

Cross-Domain Image Matching

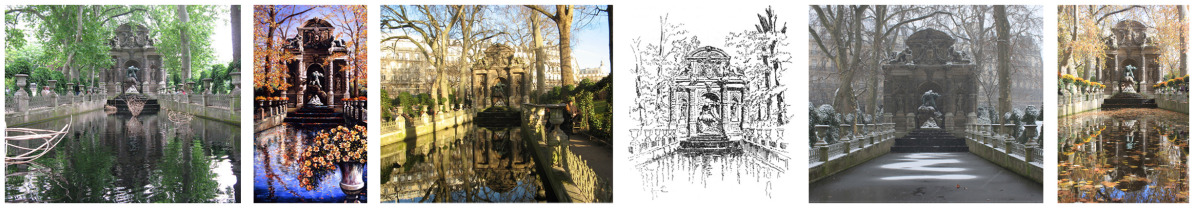


Figure A.1: **Cross-Domain Image Matching:** We are interested in defining visual similarity between images across different domains, such as photos taken over different seasons and lighting, paintings, sketches, etc. What makes this challenging is that the visual content is only similar on the higher scene level, but quite dissimilar on the pixel level.

In this thesis, we summarize the results of an application of the Exemplar-SVM framework to the problem of cross-domain matching [Shrivastava et al., 2011]. We briefly discuss the problem and only show a subset of our results for several different cross-domain applications.

The central element common to many image-matching applications in computer graphics is searching a large dataset to find visually-similar matches to a given query — be it an image patch, a full image, or a spatio-temporal block. However, defining a good visual similarity metric to use for matching can often be surprisingly difficult. Granted, in many situations where the data is reasonably homogeneous (e.g., different patches within the same texture image [Efros and Freeman, 2001], or different frames within the same video [Schodl et al., 2000]), a simple pixel-wise, sum-of-squared-differences matching works quite well. But what about the cases when the visual content is only similar on the higher scene level, but quite dissimilar on the pixel level, such as in Figure A.1. For instance, methods that use scene matching (e.g., [Hays and Efros, 2007, Dale et al., 2009]) often need to match images across different illuminations, different seasons, different cameras, etc. Likewise, re-texturing an image in the style of a painting [Hertzmann et al., 2001, Efros and Freeman, 2001] requires making visual correspondence between two very different domains — photos and paintings. The cross-domain matching is even more critical for applications such as Sketch2Photo [Chen et al., 2009] and CG2Real [Johnson et al., 2010], which aim to bring domains as different as sketches and CG renderings into correspondence with natural photographs. In all of these cases, pixel-wise matching fares extremely poorly, because perceptually small differences can result in arbitrary large pixel-wise differences. What is needed is a visual metric that can show some robustness to small, unimportant visual differences, yet still capture the important visual structures that make two images appear similar. This is precisely what makes this problem so difficult — the visual similarity algorithm somehow needs to know which visual structures are important for a human observer and which are not.

We show that a novel version of the Exemplar-SVM algorithm provides a very simple, yet surprisingly effective approach to visual matching which is particularly well-suited for matching images across different domains. Given an image represented by some features the aim is to focus the matching on the features that are the most visually important *for this particular image*. The central idea is use the notion of "data-driven uniqueness". We hypothesize that the important parts of the images are those that are more unique or rare within the visual world (represented here by a large dataset). Note that since the same local features could represent very different visual content

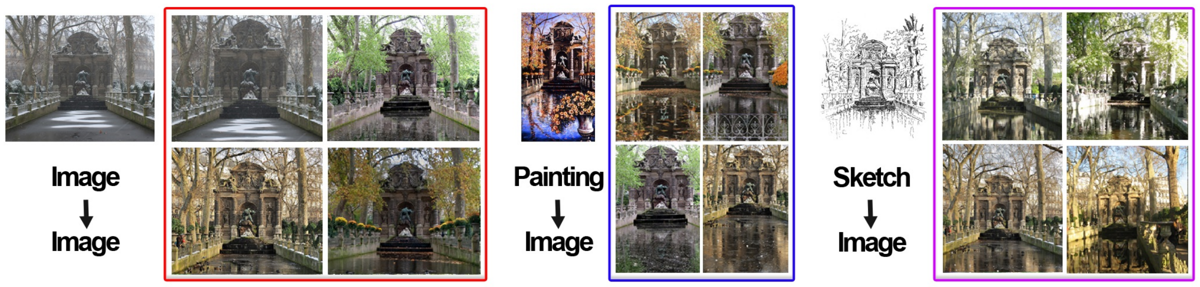


Figure A.2: **Sketch and Painting Matching** Our approach works well across different visual domains including sketches and paintings

depending of context, our notion of uniqueness has to be *scene-dependent*, i.e., each query scene decides what's the best way to weight its constituent parts. By focusing on the globally salient parts of the image, the approach can be successfully used for generic cross-domain matching, without making any domain-specific changes, as shown on Figure A.2.

We operationalize this data-driven uniqueness by using the Exemplar-SVM framework, but in a purely category-free manner. For a query image, we train an Exemplar-SVM to discover which parts of an image are most discriminative *in relationship to the rest of the dataset*. Thus in this scenario, the Exemplar-SVM's negative set does not use any class-information (which was required in [Malisiewicz et al., 2011]).

Figure A.3 show some example sketch queries and the corresponding top 6 retrieval results for our approach and the baselines — it can be seen that our approach not only outperforms all of the baselines, but returns images showing the target object in a very similar pose and viewpoint as the query sketch. Qualitative examples of painting-to-image matching can be seen in Figure A.4. Qualitative painting2GPS examples can be seen in Figure A.5.

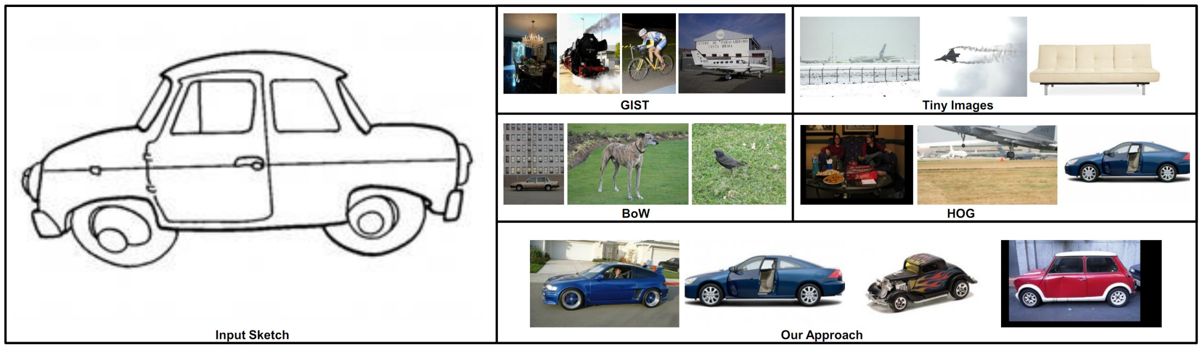


Figure A.3: **Sketch2Image Qualitative Example.** Our approach using the Exemplar-SVM algorithm is compared to several popular techniques for matching images.

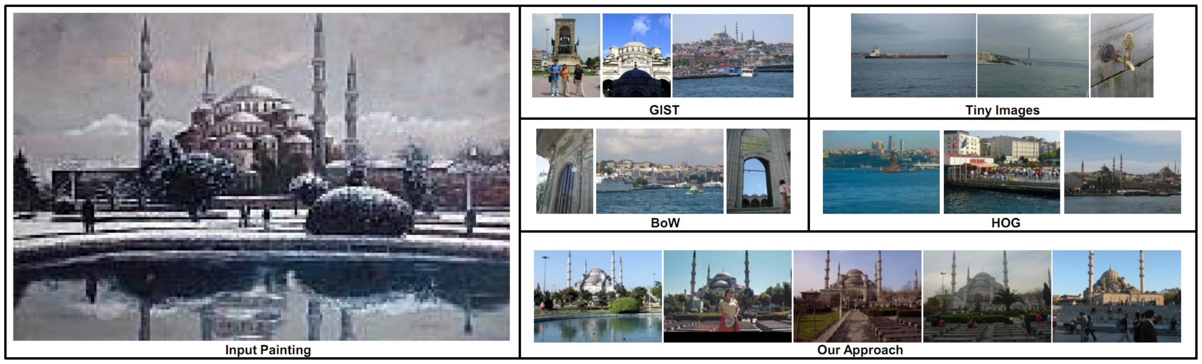


Figure A.4: **Painting2Image Qualitative Example.** Our approach using the Exemplar-SVM algorithm is compared to several popular techniques for matching images.

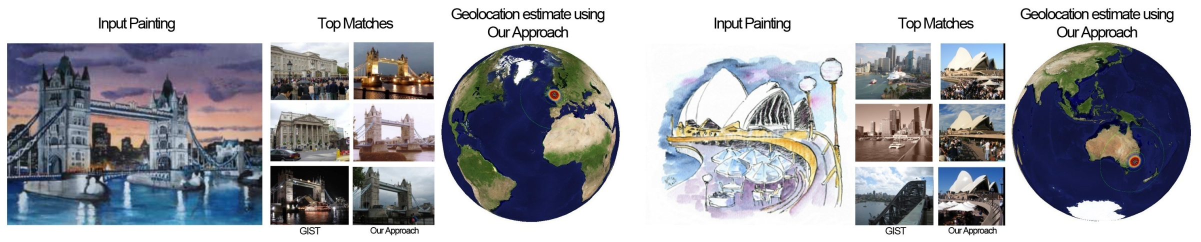


Figure A.5: **Painting2GPS Qualitative Examples.** In these two examples, we show the input painting as well as the top 3 matches using GIST and our approach. We display GPS localization results as a density map overlaid on the globe.

Appendix B

Exemplar-SVMs: An Open Source Implementation

We briefly describe the implementation of the Exemplar-SVM algorithm used to train thousands of exemplars across a cluster of $200+$ machines, in a MapReduce-like fashion. While we learn $\mathbf{w}$'s for a large number of exemplars, each exemplar's learning problem and calibration can be solved *independently* allowing for easy parallelization. The Matlab source code of the Exemplar-SVM framework (Chapter 3 and Chapter 4) is available as a public, open-source project on GitHub. GitHub is a code sharing website with many modern social-networking features built-in. The philosophy behind hosting open-source projects on GitHub is that it makes collaboration and interaction much easier than uploading a tarball on a website. GitHub makes it easy to use a project, track its progress, and also contribute any bug-fixes and enhancements for the computer vision community to enjoy.

1. Exemplar-SVM Project: `http://www.cs.cmu.edu/~tmalisie/projects/iccv11/`

2. Exemplar-SVM Code: `https://github.com/quantombone/exemplarsvm`

The Exemplar-SVM system requires access to a cluster with a shared network drive. To be able to run on a variety of clusters, the basic principle which we employ is that of a single process which could be replicated N times. There is one main driver program, which when looping over exemplars, uses lockfiles to see if an exemplar has already been trained. If an exemplar has not been trained, the current process grabs the exemplars and writes a lockfile to disk — any other process will simply skip over an already finished exemplar or a locked exemplar.

B.1 Exemplar-SVM Training Parameters

We use `libsvm` [Chang and Lin, 2001] to train each exemplar's $\mathbf{w}$. We alternate between learning the weights given an active set of negative windows, and mining additional negative windows using the current $\mathbf{w}$ as in [Felzenszwalb et al., 2010]. We use the same regularization parameter $C_1 = 0.5$ and $C_2 = .01$ for all exemplars, but found our weight vectors to be robust to a wide range of Cs, especially since they are re-scaled during calibration. We use 2500 as the maximum number of negatives to mine, which corresponds to a single pass over the negatives. We consider images as well as their left-right flipped counterparts for both training and testing.

B.2 Sliding Windows and Feature Pyramids

One of the biggest drawbacks of segmentation-based "segment-then-recognize" techniques is that they require significant computational resources to generate candidate segments — once a segmentation is computed for an image and feature computation has been applied to the resulting segments, it is typically wise to store the results on hard-disk to avoid unnecessarily running the segmentation algorithm a second time. Segmentation-based approach preclude certain applications such as real-time / interactive-time applications (due to speed of computing a segmentation), as well as applying them to datasets of millions of images (due to storage space requirements).

The second shortcoming of using image segmentation for object recognition is that it introduces a second place where object recognition could fail. By failing to group pixels into reasonable chunks with sufficient spatial support for recognition, segmentation-based approaches can simply fail at producing *any* reasonable segments for some difficult-to-segment objects. What this means is that even if we were able to obtain *ideal* per-exemplar similarity measures, we might still not be able to recognize every single object depicted in a scene. This is a severe limitation, especially because applying per-exemplar similarity measures (which we have already spent a considerable time training) is significantly faster than generating the actual segments. What this suggests is that we should generate *more segments in less time* — if the per-exemplar similarity functions are good enough, then they could decide which segments to keep.

As will be seen in this Appendix, HOG-based sliding window approaches offer a significant speedup over their segmentation-based counterparts. The speedup is so significant that we can process many more regions per image, thus alleviating our worry that segmentation might completely miss some objects. We first outline the multiscale HOG feature-pyramid, then present the two different ways in which we can perform localization within the pyramid. The blazing-fast detection strategy based on convolution requires very little memory overhead and is suitable for dealing with a small number of exemplars (e.g., the mining stage in Exemplar-SVM learning). Unfortunately, using sliding-window convolution for a large number of exemplars can still be slow. Our contribution in this chapter is the **block-matching** method of detection which requires a single matrix-multiplication and is much faster when localizing thousands of exemplars inside a single image (e.g., applying an ensemble of Exemplar-SVMs during testing).

Multiscale HOG Pyramid

Histograms of Oriented Gradients (HOG) are some of the most widely used descriptors due to their speed of computations, inherent robustness to slight object variations/deformations, and ability to capture a coarse (yet rigid) spatial layout of features. While initially introduced for pedestrian detection [Dalal and Triggs, 2005] using a linear SVM, HOG is also used by the state-of-the-art deformable part model of [Felzenszwalb et al., 2010].

Rather than independently computing HOG descriptors for different regions in an image, HOG is most often computed in a pyramid-like fashion. Image pyramids [Adelson et al., 1991] are a classical techniques in vision, and are critical in making HOG computations fast. One of the hallmarks of using feature pyramids for localization is that we do not require explicitly looping over regions in the image and independently calling a `features(region)` function. Since HOG features are a concatenation of smaller HOG cells, two highly overlapping candidate regions from the same level of the pyramid will share many of the same blocks. In other words, it suffices to compute the HOG descriptor for each cell in an image only once, then concatenating different cells when we want descriptors computed for different regions. We use 10 levels per octave when creating the pyramid and use 1.0 as the finest image scale.

Representing Objects as HOG Templates

We use the HOG descriptor from [Felzenszwalb et al., 2010] which uses $F = 31$ numbers to represent each cell, and a block size of 8 pixels ($\texttt{sbin} = 8$). When initializing exemplars, instead of warping each exemplar to a canonical frame (as is necessary in monolithic methods [Dalal and Triggs, 2005]), we let each exemplar define its own HOG dimensions from the ground-truth bounding box's aspect ratio. Using a single positive instance means that we do not need to worry about the alignment of multiple instances (which must be addressed in monolithic classifiers by an alignment strategy akin to the latent root-filter update step of Latent SVMs [Felzenszwalb et al., 2008]).

When it is necessary to extract features over a target region, provided as a bounding box, we simply compute the image-wide multiscale HOG pyramid and search for a slice of the pyramid whose spatial extent best matches the target bounding box (according to overlap score). Since we could extract multiple rectangular regions (from different levels of the pyramid) to represent the exemplar, it is also necessary to provide additional constraints regarding the size of the desired HOG dimensions. We use two simple heuristics for choosing how to frame an exemplar: finding a template which comprises roughly `goalsize` cells, and a maximum cell dimension, `maxcelldimension`. In the HOG experiments used throughout this thesis, we use `goalsize`$=100$ and `maxcelldimension`$=12$. This means that we look for a pyramid slice which has high overlap with the ground-truth bounding box, comprises roughly 100 cells and has a maximum cell dimension of 12. For example, if an exemplar is represented with a 10×10 HOG descriptor, then the total number of dimensions in the descriptor will be 3100 (assuming $F = 31$).

Localizing a single exemplar via convolution

For the purpose of localizing a single exemplar, represented by a template $\mathbf{w}$, within the HOG pyramid, we apply a convolution across each level of the pyramid independently. Given a level of the pyramid, represented as a $M \times N \times F$ matrix of numbers (F being the dimensionality of the descriptor), and $\mathbf{w}$ being of size $M' \times N' \times F$, we perform a convolution which results in a $(M - M') \times (N - N')$ matrix of scores. This method of localization is very fast since it does not require the creation of an explicit feature matrix — the overlapping HOG features only need to be stored in pyramid form. We use this convolution-based strategy during the hard negative mining step of the Exemplar-SVM algorithm because training is performed independently for each exemplar.

Applying Distance Functions via Two Convolutions

Convolution can also be used for localizing exemplars represented by distance functions, although the complexity is more than twice that of linear templates (because the convolution must be applied twice and a squared HOG pyramid must be computed). A squared HOG pyramid is obtained by squaring all of the elements in the original HOG pyramid. To see which two convolutions must be performed, consider the diagonal Mahalanobis distance function form (same as as defined in Equation 2.1):

$$D_e(\mathbf{x}_i) = (\mathbf{x}_e - \mathbf{x}_i)^T W_e (\mathbf{x}_e - \mathbf{x}_i) \tag{B.1}$$

$$\mathbf{w}_e = diag(W_e) \tag{B.2}$$

We can perform some manipulations (which exploit the fact that W_e is diagonal) to write the distance function as follows:

$$D_e(\mathbf{x}_i) = \mathbf{x}_e^T W_e \mathbf{x}_e + \mathbf{x}_i^T W_e \mathbf{x}_i - 2\mathbf{x}_e^T W_e \mathbf{x}_i \tag{B.3}$$

$$D_e(\mathbf{x}_i) = K + \widehat{\mathbf{w}}_e^T \mathbf{x}_i + \mathbf{w}_e^T \widehat{\mathbf{x}}_i \tag{B.4}$$

$$K = \mathbf{x}_e^T W_e \mathbf{x}_e \tag{B.5}$$

$$\mathbf{x}_i^T W_e \mathbf{x}_i = \mathbf{w}_e^T \widehat{\mathbf{x}}_i \tag{B.6}$$

$$\widehat{\mathbf{x}}_i[k] = (\mathbf{x}_i[k])^2 \tag{B.7}$$

$$\widehat{\mathbf{w}}_e = -2W_e \mathbf{x}_e \tag{B.8}$$

As can be seen in Equation B.4, the first part of the distance function evaluation, K, is independent of the test window $\mathbf{x}_i$ and needs to be computed only once, and can be cached for future use. The part of the distance function evaluation which depends on the test-window's features can be broken down into two convolutions: the first using the weight vector $\mathbf{w}_e$, but applied to the squared HOG pyramid $\widehat{\mathbf{x}}_i$, and the second using $\widehat{\mathbf{w}}_e$ and applied to the raw HOG pyramid.

Localizing many exemplars via a single matrix multiplication

In the case of applying thousands of exemplars to the HOG pyramid extracted from a single image, it is beneficial to explicitly construct the $D \times N$ feature-matrix, where D is the total dimensionality of the HOG template, and N is the total number of candidate bounding boxes extracted from a single image (across all pyramid levels). We call this approach the **block-matching method** for sliding-window localization. The key insight which allows us to do this is the realization that the result of applying an $M \times N$ HOG template is equivalent to applying an enlarged, 0-padded, $(M+\Delta_M)\times(N+\Delta_N)$ template where Δ_M and Δ_N indicate how much to pad. Because application of a template is a simple dot-product, adding zeros will not change the result (as long as we also pad the HOG pyramid with 0s to account for boundary effects). We exploit the fact that we used `maxcelldimension`$= 12$ during exemplar initialization (see Appendix B.2). All of our exemplars can then be 0-padded into a common 12×12 frame (i.e., $D = 4464$).

Given a new image, we first extract all 12×12 windows from each level of its 0-padded HOG pyramid. In practice we get $N \approx 20,000$ distinct windows when using `sbin`$= 8$ and using images which comprise approximately 300×400 pixels. All of our exemplars can now be applied to this image because they have also been 0-padded to the canonical 12×12 frame. In the case of using linear templates, as is the case with the Exemplar-SVM algorithm, all of our exemplars can be applied via a single matrix multiplication (even if they were initially of different dimensions). When applying distance functions (i.e., diagonal Mahalanobis metrics) in a similar block-fashion (as was necessary for the experiments in Section 3.4), we need to both 0-pad the exemplar's features *and* learned distance function weights to fit the common 12×12 frame.

While the block-matching methods takes more computer memory to store the highly-redundant feature-matrix and takes approximately 4 seconds to compute per image, this computation is independent of the number of exemplars. The single matrix-multiplication is quite fast and provides a significant speedup over convolution when the number of exemplars is moderately large. We found that for up to 50 exemplars, the convolution-based method is faster, while for more than 50 exemplars, the single-matrix-multiplication method is faster.

B.3 Exemplar-SVM run-time complexity

The run-time complexity of our approach at test time scales linearly with the number of positive instances (but unlike kernel-SVM methods, not the negatives). However, in practice, the bottleneck appears to be per-image tasks (loading, computing HOG pyramid etc.) – the actual per-instance computation is just a single dot-product, which can be done extremely fast. For an average PASCAL class (~ 300 training examples yielding ~ 300 separate classifiers) our method is only 6 times slower than a category-based method such as [Felzenszwalb et al., 2010]. More generally, because of the long-tailed distribution of objects in the world (10% of objects own 90% of exemplars [Torralba et al., 2008]), the extra cost of using exemplars vs. categories will greatly diminish as the number of categories increases.

Bibliography

[Adelson, 2001] Adelson, E. (2001). On seeing stuff: the perception of materials by humans and machines. In *Proc. SPIE*.

[Adelson et al., 1991] Adelson, E. H., Simoncelli, E. P., and Freeman, W. T. (1991). Pyramids and multiscale representations. *Proc. 13th European Conference on Visual Perception*.

[Aha et al., 1991] Aha, D. W., Kibler, D., and Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, 6.

[Aristotle,] Aristotle. *Categories*.

[Atkeson et al., 1997] Atkeson, C., Moore, A., and Schaal, S. (1997). Locally weighted learning. *Artificial Intelligence Review*.

[Bar, 2007] Bar, M. (2007). The proactive brain: Using analogies and associations to generate predictions. *Trends in Cognitive Science*.

[Bar, 2009] Bar, M. (2009). The proactive brain: memory for predictions. *Philosophical Transactions of the Royal Society B*, 364:1235–1243.

[Bar and Ullman, 1996] Bar, M. and Ullman, S. (1996). Spatial context in recognition. *Perception*, 25:343–352.

[Basri, 1992] Basri, R. (1992). Recognition by prototypes. *International Journal of Computer Vision*, 19:19–147.

[Belongie et al., 2002] Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *PAMI*.

[Berg et al., 2005] Berg, A. C., Berg, T. L., and Malik, J. (2005). Shape matching and object recognition using low distortion correspondence. *CVPR*.

[Berkeley, 1710] Berkeley, G. (1710). *Treatise Concerning the Principles of Human Knowledge*.

[Bourdev et al., 2010] Bourdev, L., Maji, S., Brox, T., and Malik, J. (2010). Detecting people using mutually consistent poselet activations. *ECCV*.

[Brady et al., 2009] Brady, T., Konkle, T., Alvarez, G., and Oliva, A. (2009). Visual long-term memory has a massive storage capacity for object details. *Proceedings of the National Academy of Sciences, USA*.

[Bush, 1945] Bush, V. (1945). As we may think. *The Atlantic Monthly*.

[Chang and Lin, 2001] Chang, C.-C. and Lin, C.-J. (2001). *LIBSVM: a library for support vector machines*. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[Chapelle, 2007] Chapelle, O. (2007). Training a support vector machine in the primal. *Neural Computation*, 19:1155–1178.

[Chen et al., 2009] Chen, T., Cheng, M.-M., Tan, P., Shamir, A., and Hu, S.-M. (2009). Sketch2photo: internet image montage. *ACM Trans. Graph.*, 28.

[Chen et al., 2001] Chen, Y., Zhou, X., , and Huang, T. S. (2001). One-class svm for learning in image retrieval. *ICIP*.

[Chum and Zisserman, 2007] Chum, O. and Zisserman, A. (2007). An exemplar model for learning object classes. *CVPR*.

[Comaniciu and Meer, 2002] Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *PAMI*, 24(5):603–619.

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. *CVPR*.

[Dale et al., 2009] Dale, K., Johnson, M. K., Sunkavalli, K., Matusik, W., and Pfister, H. (2009). Image restoration using online photo collections. In *International Conference on Computer Vision*.

[Edelman, 1995] Edelman, S. (1995). Representation, similarity and the chorus of prototypes. *Minds and Machines*.

[Efros and Freeman, 2001] Efros, A. A. and Freeman, W. T. (2001). Image quilting for texture synthesis and transfer. *Proceedings of SIGGRAPH 2001*.

[Ellson et al., 2001] Ellson, J., Gansner, E. R., Koutsofios, E., North, S. C., and Woodhull, G. (2001). Graphviz - open source graph drawing tools. In *Graph Drawing*, pages 483–484.

[Everingham et al., 2010] Everingham, M., Gool, L. V., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *IJCV*.

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

[Felzenszwalb et al., 2010] Felzenszwalb, P. F., Girschick, R. B., McCallester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part based models. *PAMI*.

[Fisher and Hanrahan, 2010] Fisher, M. and Hanrahan, P. (2010). Context-based search for 3d models. In *ACM SIGGRAPH Asia 2010 papers*, SIGGRAPH ASIA '10, pages 182:1–182:10. ACM.

[Frome and Malik, 2006] Frome, A. and Malik, J. (2006). Image retrieval and recognition using local distance functions. *NIPS*.

[Frome et al., 2007] Frome, A., Singer, Y., Sha, F., and Malik, J. (2007). Learning globally-consistent local distance functions for shape-based image retrieval and classification. *ICCV*.

[Frome, 2007] Frome, A. L. (2007). *Learning Local Distance Functions for Exemplar-Based Object Recognition*. PhD thesis, EECS Department, University of California, Berkeley.

[Galleguillos et al., 2008] Galleguillos, C., Rabinovich, A., and Belongie, S. (2008). Object categorization using co-occurrence, location and appearance. *ECCV*.

[Gu and Ren, 2010] Gu, C. and Ren, X. (2010). Discriminative mixture-of-templates for viewpoint classification. *ECCV*.

[Halevy et al., 2009] Halevy, A., Norvig, P., and Pereira, F. (2009). The unreasonable effectiveness of data. *Intelligent Systems*.

[Hanson and Riseman, 1978] Hanson, A. and Riseman, E. (1978). Visions: A computer system for interpreting scenes. *Computer Vision Systems*, pages 303–333.

[Hays and Efros, 2007] Hays, J. and Efros, A. A. (2007). Scene completion using millions of photographs. *ACM Transactions on Graphics (SIGGRAPH 2007)*, 26.

[He et al., 2004] He, X., Zemel, R. S., and Carreira-Perpiñán, M. Á. (2004). Multiscale conditional random fields for image labeling. *CVPR*, pages 695–702.

[Hertzmann et al., 2001] Hertzmann, A., Jacobs, C., Oliver, N., Curless, B., and Salesin, D. (2001). Image analogies. In *Proceedings of SIGGRAPH 2001*.

[Hoiem et al., 2005] Hoiem, D., Efros, A. A., and Hebert, M. (2005). Automatic photo pop-up. *ACM Transactions on Graphics (SIGGRAPH 2005)*, 24(3).

[Hume, 1739] Hume, D. (1739). *A Treatise of Human Nature*.

[Johnson et al., 2010] Johnson, M. K., Dale, K., Avidan, S., Pfister, H., Freeman, W. T., and Matusik, W. (2010). CG2real: Improving the realism of computer generated images using a large collection of photographs. *IEEE Transactions on Visualization and Computer Graphics*.

[Kemelmacher-Shlizerman et al., 2011] Kemelmacher-Shlizerman, I., Shechtman, E., Garg, R., and Seitz, S. M. (2011). Exploring photobios. *SIGGRAPH*.

[Kumar and Hebert, 2005] Kumar, S. and Hebert, M. (2005). A hierarchical field framework for unified context-based classification. *ICCV*.

[Lai et al., 2011] Lai, K., Bo, L., Ren, X., and Fox, D. (2011). Sparse distance learning for object recognition combining rgb and depth information.

[Lakoff, 1987] Lakoff, G. (1987). *Women, Fire, and Dangerous Things: What Categories Reveal About the Mind*.

[Lee and Grauman, 2010] Lee, Y. J. and Grauman, K. (2010). Object-graphs for context-aware category discovery. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–8.

[Leibe et al., 2004] Leibe, B., Leonardis, A., and Schiele, B. (2004). Combined object categorization and segmentation with an implicit shape model. *Workshop on Statistical Learning in Computer Vision, ECCV*.

[Li et al., 2010] Li, F., Carreira, J., and Sminchisescu, C. (2010). Object recognition as ranking holistic figure-ground hypotheses. *CVPR*.

[Li et al., 2011] Li, Y., Gu, L., and Kanade, T. (2011). Robustly aligning a shape model and its application to car alignment of unknown pose. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PP(99):1.

[Liu et al., 2009] Liu, C., Yuen, J., and Torralba, A. (2009). Nonparametric scene parsing: Label transfer via dense scene alignment. *CVPR*.

[Locke, 1689] Locke, J. (1689). *An Essay Concerning Human Understanding*.

[Malisiewicz and Efros, 2007] Malisiewicz, T. and Efros, A. A. (2007). Improving spatial support for objects via multiple segmentations. *BMVC*.

[Malisiewicz and Efros, 2008] Malisiewicz, T. and Efros, A. A. (2008). Recognition by association via learning per-exemplar distances. *CVPR*.

[Malisiewicz and Efros, 2009] Malisiewicz, T. and Efros, A. A. (2009). Beyond categories: The visual memex model for reasoning about object relationships. *NIPS*.

[Malisiewicz et al., 2011] Malisiewicz, T., Gupta, A., and Efros, A. A. (2011). Ensemble of exemplar-svms for object detection and beyond. *ICCV*.

[Medin and Schaffer, 1978] Medin, D. L. and Schaffer, M. (1978). Context theory of classification learning. *Psychological Review*, 85:207–238.

[Murphy, 2002] Murphy, G. L. (2002). *The Big Book of Concepts*. The MIT Press.

[Nosofsky, 1986] Nosofsky, R. M. (1986). Attention, similarity, and the identification-categorization relationship. *Journal of Experimental Psychology: General*, 115(1):39–57.

[Paget and Longstaff, 1998] Paget, R. and Longstaff, I. D. (1998). Texture synthesis via a noncausal nonparametric multiscale markov random field. *IEEE Transactions on Image Processing*.

[Parikh et al., 2008] Parikh, D., Zitnick, C. L., and Chen, T. (2008). From appearance to context-based recognition: Dense labeling in small images. *CVPR*.

[Park et al., 2010] Park, D., Ramanan, D., and Fowlkes, C. (2010). Multiresolution models for object detection. *ECCV*.

[Perona, 2010] Perona, P. (2010). Vision of a visipedia. *Proceedings of the IEEE*.

[Platt, 1999] Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*.

[Rabinovich et al., 2007] Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewora, E., and Belongie, S. (2007). Objects in context. *ICCV*.

[Ramanan and Baker, 2011] Ramanan, D. and Baker, S. (2011). Local distance functions: A taxonomy, new aglorithms, and an evaluation.

[Ren et al., 2005] Ren, L., Patrick, A., Efros, A. A., Hodgins, J. K., and Rehg, J. M. (2005). A data-driven approach to quantifying natural human motion. *ACM Transactions on Graphics (SIGGRAPH 2005)*, 24(3):1090–1097.

[Rosch, 1978] Rosch, E. (1978). Principles of categorization. *Cognition and Categorization*, pages 27–48.

[Rosch and Mervis, 1975] Rosch, E. and Mervis, C. B. (1975). Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology*, 7:573–605.

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

[Russell et al., 2009] Russell, B., Efros, A. A., Sivic, J., Freeman, W. T., and Zisserman, A. (2009). Segmenting scenes by matching image composites. *NIPS*.

[Russell et al., 2008] Russell, B., Torralba, A., Murphy, K., , and Freeman, W. T. (2008). Labelme: a database and web-based tool for image annotation. *IJCV*.

[Russell et al., 2007] Russell, B. C., Torralba, A., Liu, C., Fergus, R., and Freeman, W. T. (2007). Object recognition by scene alignment. *NIPS*.

[Savarese and Fei-Fei, 2007] Savarese, S. and Fei-Fei, L. (2007). 3d generic object categorization, localization and pose estimation. *ICCV*.

[Schank, 1983] Schank, R. C. (1983). *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*. Cambridge University Press.

[Schodl et al., 2000] Schodl, A., Szeliski, R., Salesin, D. H., and Essa, I. (2000). Video textures. In *Proceedings of SIGGRAPH '00*.

[Schlkopf et al., 2001] Schlkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation*.

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

[Shirky, 2005] Shirky, C. (2005). Ontology is overrated: Categories, links, and tags.

[Shotton et al., 2006] Shotton, J., Winn, J. M., Rother, C., and Criminisi, A. (2006). Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. *ECCV*.

[Shrivastava et al., 2011] Shrivastava, A., Malisiewicz, T., Gupta, A., and Efros, A. A. (2011). Data-driven visual similarity for cross-domain image matching. *Submitted to SIGGRAPH Asia*.

[Sivic and Zisserman, 2003] Sivic, J. and Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In *ICCV*, pages 1470–, Washington, DC, USA. IEEE Computer Society.

[Skagestad, 1996] Skagestad, P. (1996). The mind's machines: The turing machine, the memex, and the personal computer. *SEMIOTICA*.

[Strat and Fischler, 1991] Strat, T. and Fischler, M. (1991). Context-based vision: Recognizing objects using information from both 2-d and 3-d imagery. *PAMI*, 13:1050–1065.

[Thomas et al., 2009] Thomas, A., Ferrari, V., Leibe, B., Tuytelaars, T., and Gool, L. J. V. (2009). Shape-from-recognition: Recognition enables meta-data transfer. *CVIU*.

[Torralba, 2003a] Torralba, A. (2003a). The context challenge. *http://web.mit.edu/torralba/www/carsAndFacesInContext.html*.

[Torralba, 2003b] Torralba, A. (2003b). Contextual priming for object detection. *International Journal of Computer Vision*, 53:169–191.

[Torralba et al., 2008] Torralba, A., Fergus, R., and Freeman, W. T. (2008). 80 million tiny images: a large dataset for non-parametric object and scene recognition. *PAMI*, 30.

[Vedaldi et al., 2009] Vedaldi, A., Gulshan, V., Varma, M., and Zisserman, A. (2009). Multiple kernels for object detection. *ICCV*.

[Wahba, 1973] Wahba, G. (1973). A class of approximate solutions to linear operator equations. *Journal of Approximation Theory*, 9(1):61 – 77.

[Weinberger, 2007] Weinberger, D. (2007). *Everything is miscellaneous : the power of the new digital disorder*. New York : Times Books.

[Wittgenstein, 1953] Wittgenstein, L. (1953). *Philosophical Investigations*. Blackwell Publishing.

[Xiao et al., 2010] Xiao, J., Hays, J., Ehinger, K., Oliva, A., and Torralba, A. (2010). Sun database: Large-scale scene recognition from abbey to zoo. *CVPR*.

[Xu et al., 2006] Xu, L., Crammer, K., and Schuurmans, D. (2006). Robust support vector machine training via convex outlier ablation. *AAAI*.

[Zhang et al., 2006] Zhang, H., Berg, A., Maire, M., and Malik, J. (2006). SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. *CVPR*, pages 2126–2136.