Object Class Detection: A Survey

XIN ZHANG, National University of Defense Technology
YEE-HONG YANG, University of Alberta
ZHIGUANG HAN, HUI WANG, and CHAO GAO, National University of Defense Technology

Object class detection, also known as category-level object detection, has become one of the most focused areas in computer vision in the new century. This article attempts to provide a comprehensive survey of the recent technical achievements in this area of research. More than 270 major publications are included in this survey covering different aspects of the research, which include: (i) problem description: key tasks and challenges; (ii) core techniques: appearance modeling, localization strategies, and supervised classification methods; (iii) evaluation issues: approaches, metrics, standard datasets, and state-of-the-art results; and (iv) new development: particularly new approaches and applications motivated by the recent boom of social images. Finally, in retrospect of what has been achieved so far, the survey also discusses what the future may hold for object class detection research.

Categories and Subject Descriptors: I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis; I.4.9 [**Image Processing and Computer Vision**]: Applications; I.5.4 [**Pattern Recognition**]: Applications—*Computer vision*

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Object class detection, categorization, segmentation, intra-class appearance variation, appearance model, evaluation, social images

ACM Reference Format:
Zhang, X., Yang, Y.-H., Han, Z., Wang, H., and Gao, C. 2013. Object class detection: A survey. ACM Comput. Surv. 46, 1, Article 10 (October 2013), 53 pages.
DOI: http://dx.doi.org/10.1145/2522968.2522978

1. INTRODUCTION

Object recognition is one of the fundamental challenges in computer vision, which generally consists of two different types of tasks: *object instance recognition* and *object class recognition*. The first type aims at identifying previously seen object instances such as a specific car, and is largely a matching problem in which the differences between the stored exemplars and the objects to be reidentified in an input image are mainly caused by imaging condition changes, and hence can be effectively handled by some alignment process. The second type, also known as *category-level* or *generic object recognition*, focuses on recognizing (always unseen-before) instances of some predefined categories. It is far more challenging than the first type because of: (i) the existence

This work was funded by the National Natural Science Foundation of China (NSFC) under grant no. 60902091, the Natural Sciences and Engineering Research Council of Canada (NSERC), and the China Scholarship Council (CSC).
Authors' addresses: X. Zhang (corresponding author), College of Information Systems and Management, National University of Defense Technology, China; email: ijunzhanggm@gmail.com; Y.-H. Yang, Department of Computing Science, University of Alberta, Canada; Z. Han, H. Wang, and C. Gao, College of Information Systems and Management, National University of Defense Technology, China.
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2013 ACM 0360-0300/2013/10-ART10 $15.00
DOI: http://dx.doi.org/10.1145/2522968.2522978

of a sheer number of different categories in the real world among which the inter-category visual differences sometimes may be very small, and (ii) large intra-category appearance variations caused by different object colors, textures, shapes, as well as varying imaging conditions. Additionally, an object in a real-world scene often occupies just a (small) portion of the scene and is occluded by others or accompanied by similar-looking background structures. Thus, object class recognition in practice needs not only to determine whether or not any instances of categories of interest are present in an input image, but also to locate them accurately in the image to separate them from the background. The more complex task performed then is called *object class detection*, which aims at recognizing as well as locating instances of categories of interest in input images.

Besides being critical to recognizing object classes in real scenes, object class detection is also a prerequisite step in a wide range of higher-level vision tasks, including activity or event recognition, full scene content understanding, etc. In addition, it can support a long list of practical applications such as intelligent video surveillance [Aggarwal and Ryoo 2011], robot navigation [Krüger et al. 2007], Content-Based Image Retrieval (CBIR) [Datta et al. 2008], Image-Based Rendering (IBR) [Snavely et al. 2010], photo manipulation [Chen et al. 2009], and Augmented Reality (AR) [Palmese and Trucco 2008].

The significance of object class detection to computer vision and to practical applications has motivated researchers to focus intensively in this area in the last couple of decades. In addition, the emergence of many powerful machine learning classifiers and feature analysis techniques has given rise to a further increase in research activities in object class detection in the new century as evidenced by a large number of diverse approaches reported each year. Despite all these efforts, currently achieved accuracy (see Section 6.2) is still too low to apply object class detection methods in general-purpose practical applications. Hence, this problem is still far from being solved.

A thorough review and summarization of existing work is certainly helpful and essential for further progress in object class detection. Several researchers have already made some efforts to summarize research related to this field. For example, Ponce et al. [2006b] edited a book, titled *Toward Category-Level Object Recognition*, which collects a series of representative papers on object categorization, detection, and segmentation. Additionally, in the short courses, *Recognizing and Learning Object Categories*, given in ICCV'2005, CVPR'2007, and ICCV'2009, Fei-Fei et al. [2005, 2007, 2009] summarized existing related work from different angles. Dickinson [2009], in his book chapter, traced the evolution of object categorization in the last four decades. In addition, Szeliski set aside a chapter (Chapter 14) in his textbook [Szeliski 2010] to introduce object recognition, detection, and scene understanding. However, none of these related works focuses directly on the topic of object class detection, nor are they surveys. Furthermore, the contributions they covered are mostly published before 2008.

The preceding discussions motivate this survey, in which we comprehensively study and analyze the current progress and discuss future prospects of object class detection from different viewpoints, including the definition of the problem, its core tasks and key challenges, existing core techniques, quantitative evaluation, as well as newly emerging approaches and applications motivated by the proliferation of the Internet, in particular, social networking. A possible organization of the various issues of object class detection is given in Figure 1, which is largely inspired by a similar figure in the survey on Content-Based Image Retrieval (CBIR) [Datta et al. 2008]. The rest of this article is organized accordingly. Specifically, we describe the problem of object class detection in Section 2; summarize existing core techniques in categorical appearance modeling, localization, and supervised classification, in Sections 3, 4, and 5,

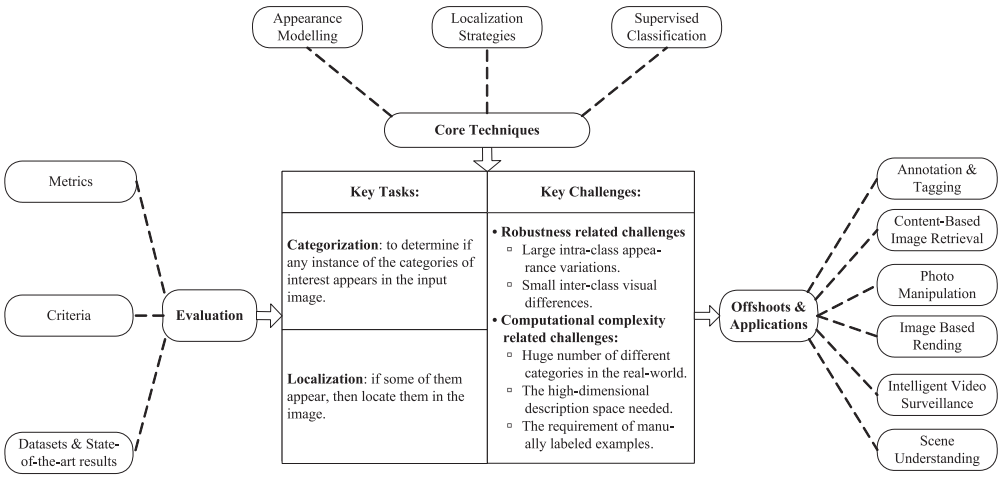


Fig. 1. Different facets related to object class detection. The arrows in the figure mean "support," for example, the core techniques including appearance modeling, localization strategies, and supervised classification support carrying out the key tasks and addressing the key challenges.

respectively; and discuss evaluation issues in Section 6. Offshoots motivated by social images are discussed in Section 7. This survey concludes in Section 8. Notably, the recent tutorial of Grauman and Leibe [2011] on visual object recognition covers very similar topics to what are presented here. It surveys both instance and category recognition techniques, but focuses mainly on the recognition and detection of objects having relatively fixed shapes and sizes, that is, structured objects, and does not cover the models and algorithms of amorphous ones. Moreover, the quantitative evaluation metrics and the offshoots motivated by social images discussed in this survey have not been included in that tutorial.

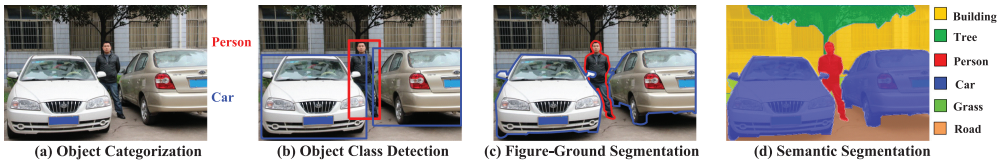
This survey mainly focuses on the major progress made in the last five years; but for completeness and better readability, some early related works are also included. Before proceeding, we first clarify several potentially confusing terms (see Appendix A for more related concepts). The term *class*, or its equivalent *category*, represent a set of objects with some common semantic features, such as person. In contrast, the term *object*, or its equivalent, *instance*, refers to a specific individual in a class, for example, Jack. Note that we use each pair of terms interchangeably in this survey.

2. THE PROBLEM OF OBJECT CLASS DETECTION

2.1. The Tasks

Object class detection, also called *category-level object detection* [Chia et al. 2009] or *object category detection* [Aytar and Zisserman 2011], consists of two key tasks, or rather, two goals (see the middle of Figure 1): (i) *object categorization* which determines whether or not any instance of the categories of interest is present in a given image, and (ii) *object localization* which determines the positions and extents of all the objects that are found present. If only one class is of interest, say the pedestrian or the face, then it is a special case of *class-specific detection* [Gall and Lempitsky 2009; Knopp et al. 2011]. A more general case is *multiclass detection* [Salakhutdinov et al. 2011; Torralba et al. 2006], by which all instances of multiple predefined categories are detected.

To better understand object class detection and its subtasks, we first consider their outputs. Typically, the output of object categorization is a list of categorical labels indicating which categories of interest have instances present in the input image (see



(a) Object Categorization (b) Object Class Detection (c) Figure-Ground Segmentation (d) Semantic Segmentation

Fig. 2. Different vision tasks related to object class detection. Best viewed in color. In (a) object categorization; (b) object class detection; and (c) figure-ground segmentation, two foreground classes of interest are *person* and *car*, which are labeled in red and blue, respectively. Semantic segmentation of the input image, shown in (d), considers four additional background classes, that is, *building*, *tree*, *grass*, and *road*.

Figure 2(a) for example). Differently, object localization results often consist of multiple choices such as bounding boxes [Dalal and Triggs 2005], the objects' centers [Shotton et al. 2005, 2008a], matched contour templates [Heitz et al. 2009; Opelt et al. 2006], or the closed boundaries of objects [Zhang et al. 2010]. Among them, rectangles tightly bounding the detected objects are most commonly used in the literature. The results of categorization and localization can be combined and visualized in a more compact way, for example, using rectangular bounding boxes in different colors to indicate objects of different categories as shown in Figure 2(b).

From the implementation point of view, if object class detection is interpreted as the task of searching in an input image for regions corresponding to objects of predefined categories, then we can perform it in the following steps: (i) enumerate all possible regions in the input image I, (ii) decide whether or not each of them corresponds to any of the predefined categories, and (iii) evaluate all regional responses to obtain the overall detection results. The key step is in object categorization of a given region R, that is, to determine whether R corresponds to any of the predefined categories, which can be implemented by matching R's visual representation with the categorical representation of each candidate category c, and then by evaluating all matching scores $S(c, R)$. The appearance model of a category is often trained using some labeled examples of that category provided by human annotators. Thus, categorization of a region R is essentially a supervised classification process, in which the categorical appearance representations are critical for discriminating different categories successfully. In addition, since there are always a huge number of possible regions in a typical image, it is crucial to efficiently locate among them candidates for which the categorization classifier responds positively. Viewing all of these, we identify representations of instances and categories, localization of object regions, and supervised classification methods as the core techniques of object class detection, which are detailed in Sections 3, 4, and 5, respectively.

The concept of object class detection can be further dissected through analyzing the strong ties it has with several related vision tasks. One of the most closely related tasks is object categorization. It can be performed at three different levels. The first level is image-level categorization (see Figure 2(a)), which determines whether or not any instance of the categories of interest is present in a testing image and hence is a subtask of object class detection as discussed at the beginning of this section. The second level is region-level categorization, which can be used in practice to implement object class detection as discussed earlier. The third level is pixel-level categorization that determines to which one of the predefined categories a given pixel belongs. It can also be applied to implement detection, in particular, to locate unstructured objects such as grass and road (see Sections 3.3 and 4.3 for more details).

Another closely related task is *figure-ground segmentation*, which aims at accurately separating foreground objects from their background in a given image [Kuettel and Ferrari 2012], and thus is essentially a kind of *category-level object segmentation*

[Larlus et al. 2010]. This is evidenced by the facts that: (i) purely bottom-up[1], category-independent methods commonly fail to locate semantic objects in a single static image, and that (ii) successful interactive figure-ground segmentation approaches, such as Lempitsky et al. [2009] and Rother et al. [2004], typically require the user to (roughly) specify the objects' locations in the input image (often in the form of a rectangle around each object just like that given by an ideal detector). The close relationship between figure-ground segmentation and object class detection is vividly displayed in Figure 2(c): when considering only the foreground objects and representing their detected locations by their closed boundaries, object class detection is equivalent to figure-ground segmentation. Moreover, since pixel-wise object segmentation provides the most accurate specification of objects' locations and extents, we argue that figure-ground segmentation can be viewed as a special case of object class detection which targets only on the foreground objects and restricts their location representations to be their closed boundaries. However, it should be noted that segmenters often focus only on the visible parts of objects in a given image, and thus their results are inaccurate for specifying the real extents and even locations of occluded objects, as exemplified by the left car in Figure 2(c). In contrast, object detectors usually attempt to localize the whole object even if it is partially occluded[2]. As a result, the objects specified by such a detector may overlap with each other. For example, the blue bounding boxes in Figure 2(b) overlap with the red bounding box surrounding the person.

In addition, since a successful segmentation (of unoccluded objects) automatically leads to a perfect localization, object class detection is also closely related with *semantic image segmentation* (see Figure 2(d)). Unlike figure-ground segmentation, it aims at segmenting all objects in a given image irrespective of whether they are the foreground or the background ones. Thus, it is a more general vision task compared with figure-ground segmentation, which can be considered as one of its subtasks. In fact, the strong ties between detection and segmentation have motivated many researchers to integrate them synergistically. We discuss these works, in particular those on how to use segmentation to aid localization, in Section 4.3.

The aforesaid relational analysis motivates us to include categorization, detection, localization, as well as segmentation methods in this survey.

2.2. Key Challenges

Generic object class detection imposes a number of difficult constraints and boundary conditions on established pattern recognition techniques. Inspired by Grauman and Leibe [2011], we categorize these challenges into two groups, that is, the *robustness-related* and *computational-complexity- and scalability-related*. The challenges of the first group consist mainly of the usually very large intra-class appearance variations and potentially small inter-class appearance differences. The challenges of the second group include the existence of a large number (or rather thousands) of different categories and their high-dimensional appearance descriptions, the difficulty in obtaining sufficient number of unambiguously labeled training samples, etc.

[1]*Bottom-up segmentation* considers primarily the visual cues inside the input image without using learned categorical representations, and thus can only partition the image into a set of regions homogeneous in terms of low-level features. In contrast, *top-down segmentation* is a process guided by stored categorical representations. It often performs recognition and segmentation in an interleaving manner, and thus can separate semantic objects from the background.

[2]We note that some datasets including the PASCAL VOC dataset described in Section 6.2 require human annotators to put the bounding box around the unoccluded portion of each object. However, this paradigm is debatable, in our point of view, as it encourages competing detectors to locate only the visible part of the object of interest.

Fig. 3. Large appearance variations exist among different instances of the category "horse." Images are selected from the INRIA Horse Dataset (http://www.vision.ee.ethz.ch/∼calvin/datasets.html) with permission from Vittorio Ferrari.

2.2.1. Robustness-Related Challenges. Intra-class appearance variations refer to the appearance differences among different objects of the same class. As shown in Figure 3, different instances of the same category, "horse" here, are always in different body shapes, poses, colors, textures, and backgrounds. In fact, due to variations in lighting, background, posture, and viewpoint and the influence of occlusions and background clutter, even the same horse may look very different in different photos. This makes it very difficult to build a categorical appearance model which can accommodate all possible intra-class variations as well as to match a trained model with a previously unseen instance.

Following the work of Schroff [2009], we subdivide typical intra-class variations into two types, namely *object variations* and *image variations*. The former are commonly caused by the different individualities of different instances, for example, they always appear in different colors, textures, shapes, poses, and sizes. The latter are often caused by different imaging or observation conditions. That is, images of different instances, or even the same instance, are usually captured at different time and locations, in different weather conditions, with different cameras, under different illuminations, and from different viewpoints.

Identical image variations, such as lighting changes, may occur in different categories, while object variations can be different for distinct categories. In particular, different categories usually have different degrees of deformation, which means a change in the shape or size of an object. According to their deformability, we can group familiar object categories into two superclasses: (i) *deformable* or *structured* classes, such as bottle, pot, building, human body, and horse, which have relatively constant shape and size features and thus their deformations can be modeled reasonably well, and (ii) *amorphous* or *unstructured classes*, such as sky, grass, and cloud, which have no constant shape or size, and hence their deformations are extremely difficult, if not impossible, to be accommodated by a structured model. Note that, even though some objects such as bottles and pots are individually rigid, they are still regarded as deformable because of their notable intra-class shape variations. Table I summarizes different object classes and their corresponding connotations of intra-class appearance variations in terms of typical visual features and imaging conditions. With respect to object variations, deformable objects of the same class, such as dogs, often differ from each other notably in color, texture, and size, while less in shape. In contrast, for amorphous objects such as sky and grass, the defining cues are commonly color or texture, while the (global) shape and size features are often ill-defined. As for image variations, we are only concerned with those that may cause the object appearances to change notably in the resulting images, and thus should be taken into account when building the categorical representations for recognition. Such image variations consist mainly of lighting,

Table I. Decomposition of Intra-Class Appearance Variations

Object Classes	Object Variations				Image Variations				
	color	texture	shape	size	lighting	viewpoint	scale	occlusion	clutter
Deformable classes	✓	✓	−✓	✓	✓	✓	✓	✓	✓
Amorphous classes	−✓	−✓	NA	NA	✓	NA	NA	NA	NA

"✓" means that the instances of the corresponding object class (row) always vary notably in this feature/condition dimension (column); "−✓" means that the class varies less, that is, it is somewhat invariant, in this feature/condition dimension; and "NA" indicates that the feature/condition is not meaningful for the categorical representation of the object class.

Fig. 4. Very small inter-class visual differences exist between the categories *dog* and *wolf*. Can you tell which one is a dog and which one is a wolf? The correct answer is that those in the left two images are dogs and those in the right two images are wolves. Image courtesy of the following Flickr users: Erkhemchukhal Dorj, Viktoriia Vitkovskaya, Mike Seamons, and Kevin Dempsey.

viewpoint, and scale changes, occlusions, and background clutter. Notably, due to having no global shapes or structures, amorphous categories, such as sky and sea, are often described in terms of local appearances based on color or texture (refer to Section 3.3), which are less sensitive to the viewpoint/scale changes, occlusions, or background clutter. Hence, these image variations are often not considered when developing their representations.

In general, more invariant features are helpful in defining or identifying the common and representative aspects of the object class of interest, and thus are more effective for recognition. Therefore, according to the results given in Table I, different classes should be described with different features, for example, color, texture, shape, and structural information can all be used to characterize deformable classes, and, in most cases, only color and texture are effective for amorphous ones. In fact, current detectors focus mainly on structured classes. For example, the 20 object classes that are defined in the popular dataset PASCAL VOC [Everingham et al. 2010], are all deformable categories. They are commonly described with texture or shape features in the literature. Since these two features are quite independent of color and lighting, the issues of intra-class variations left to be addressed when detecting structured objects are mainly occlusions, background clutter, and changes in size/scale, pose, or viewpoint.

Inter-class appearance differences refer to the visual differences between objects of different categories. As shown in Figure 4, such differences (e.g., those between *dog* and *wolf* in this figure) can be very small and even indiscernible to us. To address this problem, the categorical appearance model must be powerful enough in discriminating different classes. At the same time, the ubiquitous existence of (often very large) intra-class variations imposes stringent requirements on the model's robustness. Due to the inherent trade-offs between discriminative power and robustness [Varma and Ray 2007], these two requirements make the object class detection problem strikingly challenging.

2.2.2. Computational-Complexity- and Scalability-Related Challenges. We can roughly estimate the computational requirement needed for object class detection based on the

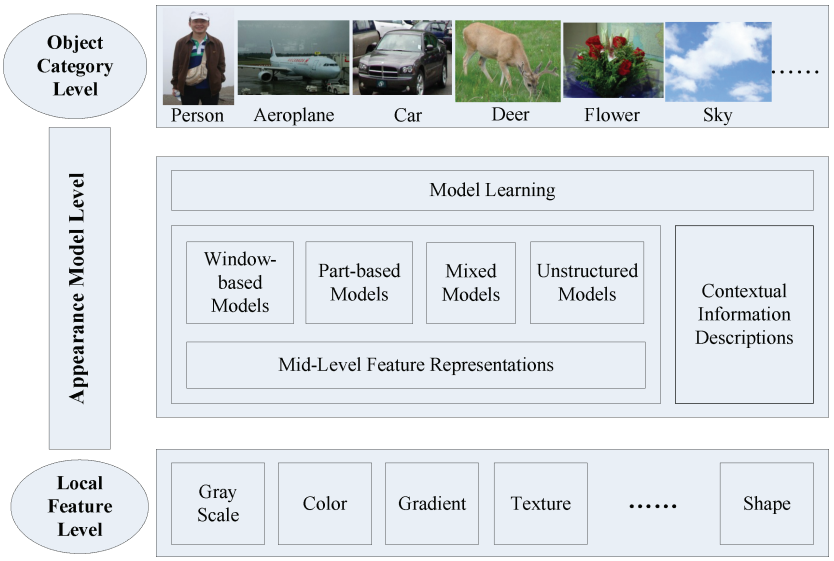


Fig. 5. The bridging role of categorical appearance models in object class detection. The appearance model level, that is, the central box, can be subdivided into two layers: the lower one consists of two boxes corresponding to the description of the object's appearances and their context, and the upper one is model learning which abstracts the categorical models from the representations of some given training examples.

fact that about half of the cerebral cortex in primates is devoted to processing visual information [Felleman and Van Essen 1991]. This is because: (i) there are a huge number of categories, each of which has a large number of images, and (ii) many visual features are required to identify each category, which results in a high-dimensional but very sparsely occupied description space [Pinz 2005]. In addition, the category models are often learned from labeled examples. While accurately labeled image examples tend to be the most informative, they are also the most labor intensive to obtain. Thus, detection systems must consider methods that can reduce human supervision by lowering the requirements on label accuracy or by learning using small training sets. Furthermore, as previously unseen instances frequently emerge (e.g., a car is newly published) and category appearances often keep evolving (e.g., the appearances of cars have changed a lot over the last five decades), it is essential to develop suitable mechanisms to update models continuously or to learn incrementally.

3. APPEARANCE MODELING TECHNIQUES

Review of the input and output of object class detection can help to further understand the importance of appearance models: the input is an image from which we can extract some low-level visual features, while the output is a set of objects which belong to high-level semantic concepts. Thus, the process of object class detection is to map low-level features to some high-level semantics. However, it is well-known that there is the so-called *semantic gap*, that is, the gap between data and semantics, between these two levels. The role of appearance models is to bridge this gap, as illustrated in Figure 5. Therefore, they are a critical component for any object class detection system.

In general, low-level features are exploited to establish appearance models, and it is the latter that characterizes different object categories. Hence, the term *appearance modeling* used in this survey refers to the process of developing representations for individual objects and abstracting from these representations the appearance model,

that is, a common representation, of each category of interest. Notably, in such a process, machine learning is often indispensable.

As discussed in Section 2.2.1, deformable classes possess relatively fixed shapes and geometric structures which could be exploited to characterize them, while amorphous ones have no such defining cues. Accordingly, *structured* and *unstructured models* are developed to represent them, which are discussed in Section 3.2 and 3.3, respectively. Note that, though these two families of models appear very different when viewed from the angle of geometrics, they have a significant overlap in terms of the low- or mid-level features commonly used, which are overviewed in Section 3.1. It is also noteworthy that all of these models focus mainly on the appearances of objects themselves. However, in the real world, objects never occur in isolation: Instances of each class tend to appear in specific environments and always covary with other objects and backgrounds in the scene. This kind of object-environment interactions is usually called the *context*. Since there is a general consensus that contextual information is an important complement to objects' appearances, it is regarded, in this survey, as a part of the *generalized* concept of appearance and its description methods are included as well into the body of appearance modeling techniques (Section 3.4).

3.1. Description of Relevant Visual Cues

Visual features commonly used for building categorical representations for recognition can be subdivided into three groups according to their different levels of locality, that is, pixel-, patch- and region-level feature descriptions.

3.1.1. Pixel-Level Feature Description. Pixel-level features are features that can be calculated at each pixel separately. Popular descriptions include gray-scale value indicating a pixel's intensity, and its color vector defined in some perceptually uniform or nonuniform color space.

3.1.2. Patch-Level Feature Description. The term "patch" often refers to a small local subwindow surrounding some point of interest in the image plane or scale pyramid, which can be sparsely sampled by a key-point detector, for example, the Difference of Gaussian (DoG) detector in Lowe [2004], or densely sampled on a regular grid as done in Dalal and Triggs [2005]. Given a point, a large number of possible descriptors can be computed over the patch centered on it. The patch is also called the *support region* or the *neighborhood* of the point. Since it is usually quite small relative to the image size, patch-level descriptors are called *local feature descriptors* as well. Different descriptors often emphasize different image properties like pixel intensities, colors, textures, edges, etc. A comprehensive survey of them is beyond the scope of this article. Hence, we only succinctly discuss in the following a very small portion of them that are commonly used in the field of object class detection. The interested reader may refer to the evaluations presented in the work of Mikolajczyk and Schmid [2005] and Van de Sande et al. [2008, 2010] for more details and discussions on gray-scale and color descriptors, respectively. Note that these two evaluations are performed in very different contexts: the first aims at specific image or object recognition, while the second is in image category recognition which is more closely related to the topics covered in this survey.

The SIFT Descriptor and its Variants. The SIFT descriptor [Lowe 2004] is undoubtedly the most famous patch-level feature description, which encodes the local gradient information within a given patch in a manner invariant to lighting changes, planar rotations, and small position shifts.

Though originally proposed as a combination of a DoG key-point detector, the SIFT descriptor has been demonstrated by later studies, such as Mikolajczyk and Schmid [2005] and Van de Sande et al. [2010], to be applied to any patches and that it achieves

generally good performance for categorization. Thus, it has become more and more popular and various modifications have been presented. Notable variants include PCA-SIFT [Ke and Sukthankar 2004], Gradient Location-Orientation Histogram (GLOH) [Mikolajczyk and Schmid 2005], etc. The idea underlying PCA-SIFT is to use Principal Component Analysis (PCA), a dimensionality reduction technique, to make the descriptor more compact and hence more efficient for matching, while GLOH extends SIFT by computing the descriptor on a log-polar location grid.

An efficient alterative to SIFT is Speeded Up Robust Features (SURF) proposed by Bay et al. [2008, 2006], which makes use of integral images [Viola and Jones 2001] to speed up the computation of key-point detection as well as descriptor evaluation.

Filter-Bank Responses. A filter bank is a set of band-pass filters that separates the input signal into multiple components, each of which carries a single frequency subband of the original signal. It can be convolved with a given image patch to capture certain spatial frequency content within the patch. The most commonly used filter banks include Gaussian functions and their derivatives, Gabor functions, wavelets, etc. In the early years, filter-bank responses were often employed for texture description and classification. Later, they were introduced into the field of object categorization by Winn et al. [2005] to encode local appearances of objects. Their proposed filter bank consists of 3 Gaussians, 4 Laplacian of Gaussians (LoG), and 4 first-order derivatives of Gaussians. A given patch is hence described using a 17-dimensional feature vector. This filter bank has been widely used in detection and in segmentation methods, for example, Shotton et al. [2008b, 2006, 2009] and Wang and Grimson [2007]. In addition, as they are shown to provide a good model of cortical simple cell receptive fields [Jones and Palmer 1987], Gabor filter banks are widely used in biologically motivated representations [Huang et al. 2008; Mutch and Lowe 2008, 2006; Serre et al. 2007b, 2005], which are also discussed in the next section.

3.1.3. Region-Level Feature Description. An image patch is often too small to accommodate a whole object or even a part of it. Therefore, to describe objects' or parts' appearances, we need to further summarize the relevant visual cues at a larger scale, that is, the region level. Note that "region" is a general concept. It is by definition a set of connected pixels in an image. Thus, a region can be a regularly or irregularly shaped segment in an image, or can even be the whole image.

A region-level description is commonly developed with the goals of: (i) capturing the (most) discriminating visual properties of the target categories or their components while (ii) keeping sufficient robustness against possible intra-class variations (see the definition given in Section 2.2.1). Toward these two goals, modern detection or categorization systems often adopt some histogram-based region-level representations such as Bag-of-Features (BoF) and Histograms of Oriented Gradients (HOG). They are usually built upon contrast-based local features such as gradients, which are invariant to lighting or color variations. Moreover, spatial binning involved in histogram computation provides robustness to in-plane rotations (provided some rotation-invariant local feature descriptor such as the SIFT descriptor is used) and shifts.

Aside from contrast-based descriptions mentioned before, shape cues are also frequently captured and described at the region level for object class recognition or detection using contour or boundary fragments, shapelets, etc. In addition, color features are sometimes used as defining cues for categories (e.g., bare body parts such as the hand and the face) having relatively constant colors.

Bag-of-Features. Bag-of-Features (BoF) is also called *Bag-of-Words* (BoW) or *Bag-of-Visual-Words* (BoVW) in the literature. It is originally proposed by Joachims [1997] to describe textual contents for text categorization. Sivic and Zisserman [2003] first

adapted it to retrieve user-specified scenes or objects from video sequences, which is essentially an instance recognition problem. Later, Csurka et al. [2004, 2006] adapted it to visual categorization, and called it *bag-of-key-points*. The ideas underlying BoF are: (i) to represent visual contents using a set of representative primitives, and (ii) to summarize the appearance of a given region by counting the occurrence frequencies of representative primitives within it.

To select representative primitives, a process called *visual vocabulary* or *codebook construction* is performed, which often requires a large collection of images containing objects of different categories and various backgrounds. Local patches are extracted from each image and the (modified) SIFT descriptors or filter-bank responses are computed on them. Hence, a large number of patch-level descriptors are produced, which are further clustered into a number of groups commonly using K-means clustering. The centroid of each cluster is then selected as a representative and added to the vocabulary. Such vocabulary entries are usually called *visual words* if they are selected from (modified) SIFT descriptors, or *textons* if they are from filter-bank responses. Thus, for terminological generality, we use in this article the acronym BoF rather than BoW. Note that the rationale of using local feature descriptors lies in their good repeatability among different images, which makes the resulting representatives an analogy of words in textual contents.

To compute the BoF vector of a given region, the local descriptor extraction process mentioned earlier is conducted on the region first. Then, each extracted local descriptor is transformed to some entries in the vocabulary, typically through Vector Quantization (VQ). This transformation is motivated by the attempt to represent the visual content in a way similar to those of the textual contents, that is, using visual words or textons instead of the original local descriptors. Then, the occurrence frequency of each vocabulary entry within the given region is counted and concatenated into a vector, which is further normalized to unity to form the final BoF vector. The resulting BoF vector is of a fixed size, which is equal to the length of the visual vocabulary.

As for implementation, there are several choices, which include the local patch detection and description methods, the clustering techniques for vocabulary construction, the schemes for transforming the extracted local descriptors to vocabulary entries, and the quantitative measure of the similarity between BoF vectors needed by both local descriptor clustering and transforming. The influence of these choices on the final categorization performance can be investigated through empirical studies. Nowak et al. [2006] concentrate mainly on the sampling scheme of local feature points. Their experimental results show that dense sampling is more preferable to sparse key-point detectors, because it can provide more regular and complete coverage of objects, and hence a higher categorization accuracy can be obtained. Zhang et al. [2007] provide a more comprehensive study that covers the available choices for key-point detector type, the level of geometric invariance, feature descriptors, the kernels of Support Vector Machines (SVM) classifiers [Vapnik 1998], and the influence of unwanted background feature points. They give a set of useful practical guidelines, from which we list the most relevant ones and give our own understanding as follows: (i) Since different features are often complementary to each other, combinations of multiple feature detectors and descriptors usually achieve better results than using a single detector and descriptor. (ii) Due to the intrinsic trade-offs between invariance and discriminativeness [Varma and Ray 2007], though local invariance properties of features are preferable, they should not exceed the level required for a given application or they may degrade the discriminative power of the features. (iii) As an orderless representation, a suitable distance metric is quite important for a high categorization performance. For an SVM-based BoF classification, nonlinear kernels such as the Gaussian kernel using the Earth Mover's Distance (EMD) metric [Rubner et al. 2000], called the *EMD kernel*,

the chi-squared kernel, and the Histogram Intersection Kernel (HIK) [Maji et al. 2008] always achieve better performances than that of using the linear ones. (iv) Training data containing background points, that is, harder training samples, are helpful for achieving better generalization ability, and hence can improve the categorization accuracy. Actually, the benefit of background pixels can also be understood with respect to *context*, which is detailed in Section 3.4.

The original BoF discussed earlier is a simple and effective region-level description. However, it has several known limitations: (i) As an unstructured representation, it contains no geometric information, which is widely accepted to be important for recognition. (ii) K-means clustering is commonly used to construct the visual vocabulary, which tends to overadapt to dense regions in the feature space. Consequently, points in sparse regions are very likely to be ignored, which could be useful for discriminating different categories. (iii) Vector quantization is involved to compute the BoF vector, which is intrinsically a lossy process. In addition, quantization is computationally expensive as it needs to search through the vocabulary for each detected feature point, especially when the vocabulary length and the number of detected feature points are large.

The goal of incorporating geometric information into BoF is to overcome its first limitation. Two different methods are commonly employed. The first is to divide the input region into a set of subregions and concatenate their BoF vectors to implicitly encode the layout information, as typified by the *spatial pyramid* BoF [Lazebnik et al. 2006]. The second is to augment BoF with the geometric relations between local features [Cao et al. 2010; Savarese et al. 2006; Wu et al. 2009].

To overcome the second limitation of the original BoF approach, that is, to improve the discriminative power of the visual vocabulary, Jurie and Triggs [2005] propose to exploit the acceptance radius to ensure that the codewords are more evenly distributed in the descriptor space. Their method is based on two interesting findings: First, densely sampled local patches always outperform sparsely detected key-points. Second, for natural images, the feature distribution of densely sampled local patches in the descriptor space is generally nonuniform but power-law like. Later, Yang et al. [2008b] proposed to unify the codebook generation with classifier training into a single optimization framework to enhance the pertinence of the codebook. Recently, Cai et al. [2010] proposed to learn a weighted similarity metric which makes the measured similarity between images with the same label larger than those with different labels. In this way, the codebook generation is also correlated with classifier training to enhance the categorization performance, which is shown to be particularly effective when the number of training samples is not sufficient for constructing a large-size codebook.

Several lines of research have been conducted to address the third limitation from different perspectives. The first is to design a hierarchical vocabulary, or vocabulary tree [Nister and Stewenius 2006] to speed up quantization. The second is to replace the *hard assignment* scheme (i.e., assigning a single vocabulary entry to each local descriptor) used in the original BoF framework with soft assignment (i.e., multiple entries may be assigned to the same local descriptor) to reduce information loss [van Gemert et al. 2010]. The third is to replace vector quantization with *sparse coding* [Wright et al. 2010], which models local descriptors as a linear combination of a few elements from a basis dictionary. Sparse coding is a plausible model of the visual cortex [Lee et al. 2006] and can often help to capture more salient properties of visual patterns [Yang et al. 2009]. As a result, using a sparse coded descriptor together with a linear classifier, for example, linear SVM, can produce very good results for image classification as shown in several recent works [Wang et al. 2010c; Yang et al. 2009; Zhang et al. 2011a]. The fourth is to develop methods to avoid using vector quantization. Notably, Boiman et al. [2008] propose to use the image-to-class distance for nonparametric image-level

object categorization. The image-to-class distance is defined as the sum of distances between the local descriptors detected in an input image and their nearest neighbors in the categorical exemplar images. Later, Wang et al. [2010c] proposed to learn a category-specific Mahalanobis distance measure to enhance the discriminativeness of the image-to-class distance and to improve its computational efficiency.

In summary, BoF is very simple while effective, and hence attracts extensive attention. However, the original BoF suffers several limitations discussed earlier. Recent work focuses on overcoming these limitations from different perspectives, from which we can summarize several widely accepted and instructive guidelines: (i) dense sampling plus invariant descriptors is the best choice for feature detection/description; (ii) to improve efficiency, quantization always needs to be speeded up; (iii) incorporating geometric information into BoF can enhance its discriminative power, and hence improve the categorization performance; (iv) training the vocabulary, the classifier, and even the similarity metrics in a unified way will also help to improve the final performance. However, how to incorporate the aforementioned guidelines in a system effectively and efficiently remains an open problem. For more on BoF, the interested reader may refer to the recent survey of Jiang et al. [2010].

HOG and its Variants. HOG was first proposed by Dalal and Triggs [2005]. It is essentially a dense version of SIFT descriptors. Specifically, it divides the input region into a set of uniformly spaced *cells* of 8×8 pixels using a dense grid and groups each set of 2×2 adjacent cells into a *block* to obtain a set of densely sampled, overlapping blocks, over each of which all the cell-wise histograms of oriented gradients are concatenated and normalized to form the block descriptor. Then, all block-wise descriptors are concatenated to form the final HOG descriptor of the input region.

HOG is a very powerful representation for structured objects. In particular, no single feature has been shown to outperform HOG for pedestrian detection up to now [Dollar et al. 2011]. Thus, it has received extensive attention after its birth, and various modifications have been proposed. Notably, inspired by the spatial pyramid BoF model [Lazebnik et al. 2006], Bosch et al. [2007a] propose the *Pyramid HOG* (PHOG) representation, which is essentially a multiresolution pyramid of HOGs. To compute the PHOG representation, the input region is divided into a sequence of increasingly finer spatial grids by repeatedly doubling the number of divisions in each axis direction (like a quadtree); and then, at each resolution level, the HOG descriptors of all subregions are computed and concatenated to be the level description. In the end, all level descriptors are concatenated and normalized to form the final PHOG vector. Another notable modification of HOG is the Co-occurrence Histograms of Oriented Gradients (CoHOG) descriptor proposed by Watanabe et al. [2009], which uses discretized gradient orientation pairs, instead of discrete orientations, to form the histogram bins. CoHOG enhances the discriminative power but pushes the feature space to a much higher-dimensional space.

GIST. The capacity of humans to perform scene categorization in as little as 100 milliseconds in the near absence of attention has been attributed to our ability to rapidly extract the "gist" of a scene [Fei-Fei et al. 2002]. In other words, the quintessential characteristics discriminating different scene categories can often be rapidly summarized by our human visual systems. Inspired by this fact, researchers, for example, Oliva and Torralba [2001] and Torralba et al. [2003], have developed a set of holistic representations, often called GIST. Similar to HOG, the GIST descriptor of a given region is the concatenation of gradient orientation histograms collected from some uniformly spaced subregions. However, these subregions are often larger than the blocks in HOG and do not overlap with each other. For example, Torralba [2003] uses a 4×4 grid to subdivide

the input image. Moreover, the gradients are often captured using the wavelet or the windowed Fourier transform.

Bio-Motivated Representations. Since humans and primates, even some lower animals, outperform the best machine vision systems in terms of any measures, it is natural for us to consider designing biologically plausible representations. Some researchers have pursued along this direction and proposed a family of bio-motivated representations, which commonly follow a theory that accounts for the first 100–200 milliseconds of processing along the feedforward ventral stream in the primate visual cortex [Riesenhuber and Poggio 1999].

The facts related to the ventral stream in the visual cortex are succinctly stated by Serre et al. [2007b]:

> "(i) Visual processing is hierarchical, aiming to build invariance to position and scale first and then to viewpoint and other transformations. (ii) Along the hierarchy, the receptive fields of the neurons (i.e., the part of the visual field that could potentially elicit a response from the neuron) as well as the complexity of their optimal stimuli (i.e., the set of stimuli that elicit a response of the neuron) increases. (iii) The initial processing of information is feedforward (for the immediate recognition tasks in which the image presentation is rapid and there is no time for eye movements or shifts of attention). (iv) Plasticity and learning probably occurs at all stages and certainly at the level of InferoTemporal (IT) cortex and PreFrontal Cortex (PFC), the topmost layers of the hierarchy."

Based on the preceding facts, various bio-motivated representations are proposed. Among them, the hierarchical model for the feedforward process proposed by Riesenhuber and Poggio [1999] is regarded as the seminal work, which summarizes the previous facts quantitatively and conforms well to the anatomical and physiological results. In its simplest form, the model consists of four layers of computational units, that is, S_1, C_1, S_2, and C_2, where simple S units alternate with complex C units. The S units combine their inputs with a bell-shaped tuning function to increase selectivity. The C units pool their inputs through a maximum (MAX) operation, thereby increasing invariance. To compute the descriptor, an input region is fed into this hierarchy to undergo feature computations at all layers. It is suggested that features from the intermediate and higher layers in the model should be learned from visual experience. However, the proposed model uses a very simple static dictionary of handcrafted features. Serre et al. [2007b, 2005] extend this model by incorporating a dictionary learned from training images and verify the effectiveness of their improved model for object categorization and detection. Later, they showed that their recognition system based on this bio-motivated representation can predict the level and the pattern of performance achieved by humans on a rapid masked animal versus nonanimal categorization task [Serre et al. 2007a]. Mutch and Lowe [2008, 2006] further extend the representation of Serre et al. by: (i) sparsifying features using only dominant orientations, (ii) performing lateral inhibition through response suppression on both spatial and scale neighborhoods, (iii) adding location and scale information into the feature above the S_2 level, and (iv) selecting features highly weighted by SVM. Some other notable extensions include Huang et al. [2008], Jhuang et al. [2007], Schindler et al. [2008], and Song and Tao [2010].

Shape Features. Shape features are also a frequent cue for object categorization and detection. Commonly used region-level shape representations include contour or boundary fragment (i.e., local edge template) [Opelt et al. 2006; Shotton et al. 2005], edgelets (i.e., short segments of lines or curves) [Wu and Nevatia 2005, 2007c], shapelets (i.e., weighted combinations of low-level features including locations, directions, and strengths of gradients) [Sabzmeydani and Mori 2007], shape context (i.e., a

Fig. 6. The self-similarity properties between images containing pentagrams. See text for more details. Best viewed in color.

3D histogram of edge point locations and orientations in a log-polar coordinate system) [Belongie et al. 2001, 2002], etc.

Self-Similarity Features. All the aforesaid region-level representations make use of some low-level visual property such as pixel colors, intensities, edges, gradients, or other filter responses and summarize their (distribution) pattern to form the final region description. Different from them, the spirit of *self-similarity* is that image regions are similar if they contain similar layout of local patterns within them but not because they have similar low-level visual properties such as color or texture. For example, though the images in Figure 6 differ from each other notably in terms of color and texture, it is easy to discern that they contain instances of the same category, that is, pentagram. What makes these images (semantically) similar is the fact that they share a similar relative geometric layout of local color or texture patterns, that is, the self-similarity property.

Generally, there are two types of self-similarity descriptors, namely *global* and *local*. Shechtman and Irani [2007] first propose a Local Self-Similarity (LSS) descriptor which captures the internal geometric layout of local self-similarities within a given subwindow R_p surrounding a pixel p. Specifically, the LSS descriptor L_p for p measures the similarity using the simple *Sum of Square* Differences (SSD) of a small patch t_p (typically 5×5) centered at p with the larger subwindow R_p (typically 40×40) also centered at p. The LSS descriptor defined in this way is a patch-level, local descriptor. Two different methods have been presented to extend it to region level. The first is to use an ensemble of patch-level LSS descriptors computed over a region as its description [Shechtman and Irani 2007]. The second is to summarize LSS descriptors with the BoF framework to obtain the BoLSS (Bag-of-LSS) vector of a region [Vedaldi et al. 2009].

The second type of self-similarity descriptors, that is, the Global Self-Similarity (GSS) descriptor, was first proposed by Deselaers and Ferrari [2010], which is an extension of the preceding LSS descriptor. Specifically, the subwindow R_p is replaced by the whole input region. Then, region-level GSS descriptors can be computed in any of the ways mentioned before for LSS descriptors.

Summary. The region-level descriptions discussed in this section are often built upon low-level features such as colors, gradients, or self-similarities. Typically, they remain close to image-level information without attempts at capturing the high-level geometric structure of objects (in terms of parts, for example) even if it is available (i.e., the target objects belong to structured categories). Thus, they are called *mid-level representations* [Boureau et al. 2010], which can be used directly for category-level image classification if the region of interest is the input image itself.

Computing these mid-level representations commonly involves two main steps [Boureau et al. 2010]: (i) the *coding* step which transforms local descriptors into codes (e.g., visual words or textons) having some desirable properties such as compactness, sparseness (i.e., most components are 0), or statistical independence, and (ii) the *spatial pooling* step which pools the local codes over some spatial areas (e.g., the whole

region for BoF, a coarse grid of cells for HOG, or coarse hierarchy of cells for spatial pyramid BoF, PHOG, and bio-motivated representations).

Popular coding methods include hard quantization, soft quantization, and sparse coding, which have been briefly discussed in this section. The original BoF [Csurka et al. 2004], the spatial pyramid BoF [Lazebnik et al. 2006], the bio-motivated features of Serre et al. [2007b, 2005], and the ensemble of LSS [Shechtman and Irani 2007] all adopt hard quantization when coding, but the first three require a learned visual vocabulary while the last one does not. By trilinear interpolation, HOG [Dalal and Triggs 2005] performs soft quantization without a vocabulary. Notably, the modified spatial pyramid BoF proposed by Yang et al. [2009] adopts sparse coding.

As for pooling, commonly used schemes include MAX and average pooling. Note that the weighted sum can be viewed as a type of averaging. Hence, among the mid-level representations discussed before, BoF, HOG, and GIST adopt average pooling, while bio-motivated representations LSS and GSS use MAX pooling. In addition, the layout of local features can be preserved if pooling is performed locally, that is, in subregions, as is done in HOG, the spatial pyramid BoF, and GIST. On the contrary, globally pooled representations (e.g., BoF) discard all such layout information.

3.2. Structured Models

Currently, three different types of models are commonly used in the literature to describe structured objects, namely *window-based*, *part-based*, and *mixed models*. They belong to a large family called *structured models*, which also includes some more generalized ones such as grammar-based models [Zhu and Mumford 2006]. In this section, we discuss the basic components of different structured models as well as how to learn them from training examples.

3.2.1. Window-Based Models. The basic idea underlying *window-based* or *global models* is to describe the appearance of an object in an image using a (mid-level) descriptor computed in a window surrounding the object. The window is usually prespecified in a fixed shape such as a rectangle or a polygon. To compute such a window-based representation, the key steps include: (i) specifying the window shape in advance, (ii) selecting appropriate features and the corresponding descriptions, and (iii) concatenating feature descriptors computed in the given image window to form a single window-based appearance descriptor.

Different window-based models differ from each other mainly in their shapes and the features used. As for the window shapes, the most popular one is undoubtedly the rectangle, evidenced by a large body of modern detectors adopting it, such as Dollar et al. [2010, 2009], Vedaldi et al. [2009], and Walk et al. [2010]. Rectangular window-based models are particularly effective for categories such as monitor and pedestrian in which instances are innately rectangular or commonly appear in canonical poses and hence can be bounded by rectangular boxes fairly tightly (see Figure 7(a) for an example). Significant progress has been achieved by rectangular window-based models in the detection of these categories. In particular, both the seminal works of Viola and Jones [2001] in face detection and of Dalal et al. [2006] and Dalal and Triggs [2005] in pedestrian detection have adopted rectangular windows and established significantly higher performance records. However, because the shape of most real objects is not rectangular (see Figure 7(b) for an example), if a rectangular window were to be used, background pixels would be included as well. These pixels may act as noise and make the resulting representation inaccurate. An alternative is to use polygonal windows to bound the objects more tightly as proposed by Yeh et al. [2009]. However, polygonal windows can only alleviate (but not overcome) the limitation of rectangular ones because in most cases they still cannot conform well to the object boundaries (as shown

(a) (b)

Fig. 7. Rectangular and polygonal bounding boxes. A monitor (in the left image) can often be bounded tightly by a rectangular box, while a boat (in the right image) cannot, in which case a polygonal box is a more appropriate choice as shown in the figure. Best viewed in color.

in Figure 7(b)). In addition, due to having more degrees of freedom than a rectangle, a polygonal window often increases the computational burden substantially. A more desirable approach in terms of window shapes is to use a window that can align adaptively with the closed boundary of the target object. However, the shape of such a window could not be prespecified as we had no knowledge about the shape of the target object to be detected before accurately localizing it. To overcome this problem, Zhang et al. [2010] propose to use *free-shape windows* instead of rectangular or polygonal ones and exploit the edge detection results in the input image to approximate the shape of such a window for each target object.

As for the appearance descriptions, HOG proposed by Dalal et al. [2006] and Dalal and Triggs [2005] is the most popular and powerful one. Since its introduction, several variants of HOG, for example, those in Laptev [2006, 2009], Vedaldi et al. [2009], Watanabe et al. [2009], and Zhu et al. [2006], have been proposed all with outstanding performance. Sometimes, window appearances are also described in terms of BoF [Vedaldi et al. 2009], bio-motivated features [Serre et al. 2007b, 2005], or shape features such as shape context [Belongie et al. 2002] or shapelets [Sabzmeydani and Mori 2007]. Though no single feature has been shown more powerful than HOG in detection tasks, it has been extensively verified and widely accepted that additional features can provide complementary information and thus help to improve, often notably, the discriminative power of window-based representations. Wojek and Schiele [2008] demonstrate experimentally that the combination of HOG, densely sampled shape context, shapelets, and Harr-like features outperforms any individual ones. Later, they further incorporated motion cues, encoded using the Internal Motion Histogram wavelet difference (IMHwd) [Dalal et al. 2006], and improved the performance of onboard pedestrian detection [Wojek et al. 2009]. Walk et al. [2010] extend the feature combination by additionally proposing and integrating the local Color Self-Similarity (CSS) features. Some other authors, such as Mu et al. [2008] and Wang et al. [2009b], explore the applicability of Local Binary Patterns (LBP) [Ojala et al. 2002], a famous texture descriptor that has yielded good results for texture classification, to window-based representation for object detection.

It is noteworthy that different feature representations, in particular the spatial pooling schemes they adopt (global or local), do bring different properties to the resulting window-based models. Specifically, if a model uses a globally pooled representation, such as BoF, which contains no layout information, it is insensitive to in-plane rotations and pose changes provided that an invariant local feature descriptor, such as the SIFT descriptor, is used. In contrast, if a locally pooled representation such as HOG is used, then the model is always sensitive to in-plane variations. Based on these differences, we subdivide window-based models into two groups, namely

globally and *locally pooled window-based models*, which are further compared in Section 3.2.5.

A window-based model is usually learned from a set of training samples consisting of both positive and negative ones. The positive samples are descriptors computed over some manually annotated bounding boxes, each of which bounds tightly to a single object of interest. To ensure the final descriptors have identical lengths, all the image windows specified by these bounding boxes are first rescaled to a predefined size before the descriptor is computed. The negative samples are computed from windows containing no object. Discriminative classifiers (refer to Section 5.3) such as the SVM and variants of boosting are often used to train window-based models. Take the model learned using the linear SVM, for example, where the training set is fed into the classifier to start the learning process, which produces as the results a set of Support Vectors (SVs) together with their weights and a constant bias. Interestingly, the weighted sum of the SVs is just the learned categorical model, which can be matched to the descriptor computed from an input window using a simple dot product operation to determine whether or not the input window contains the object of interest.

3.2.2. Part-Based Models. A typical part-based model consists of two components: a set of *parts* and the geometric relations among them, called part *topology*. Parts commonly refer to some (relatively) rigid components of an object, for example, the head or forearm of a human body. They are often described in terms of their appearance properties such as colors, textures, or gradients, as well as their geometric properties such as height, width, and shape. Part topology is usually described using the relative locations of parts and the (possible) connections among them, for example, an upper leg is connected with a lower leg.

The earliest model of this family may be the Pictorial Structures (PS) model proposed by Fischler and Elschlager [1973], which consists of spring-like part connections. A natural way to express the PS model is in terms of an undirected graph $G = (V, E)$, where the vertices $V = \{v_1, \ldots, v_n\}$ correspond to the n parts (v_i is the i-th part, $i = 1, \ldots, n$), and E is the set of edges, in which an edge $(v_i, v_j) \in E$ corresponds to a pair of connected parts v_i and v_j. An object is given by a particular part configuration $L = \{l_1, \ldots, l_n\}$. An energy function (to be minimized) is defined to fit a PS model to an image as $Energ = \sum_i m_i(l_i) + \sum_{(v_i, v_j) \in E} d_{ij}(l_i, l_j)$, where $m_i(l_i)$ denotes the matching cost of part v_i when it is placed at location l_i in the input image and is determined mainly by its appearance dissimilarity from the region in the image located in l_i. The second term penalizes the deformation of the model compared with the current part layout hypothesis, in which $d_{ij}(l_i, l_j)$ measures the model deformation when v_i and v_j are connected and placed at l_i and l_j, respectively. Since this energy function is very similar to that used in Markov Random Fields (MRF), it inspires other researchers, such as Felzenszwalb and Huttenlocher [2005] and Kumar et al. [2005], to embed the PS model into an MRF framework to simplify parameter estimation.

Depicting parts with nodes and connections between them with edges or arcs, we obtain the graphical representations of part topologies as shown in Figure 8, which is based on a similar figure in Carneiro and Lowe [2006]. Typical part topologies include: (i) The star structure (see Figure 8(a)) [Felzenszwalb et al. 2008, 2010b; Leibe et al. 2004] consists of a central reference part and a set of secondary parts connected to it. (ii) The tree structure (see Figure 8(b)) [Andriluka et al. 2009; Felzenszwalb and Huttenlocher 2005] restricts the location of each nonroot part to depend only on that of its parent. (iii) The k-fan structure (see Figure 8(c)) [Crandall et al. 2005] consists of a fully connected set of k reference parts and some secondary parts, each of which is connected to every reference part (and nothing else). (iv) The fully connected constellation structure (see Figure 8(d)) [Fergus et al. 2003, 2007] contains connections

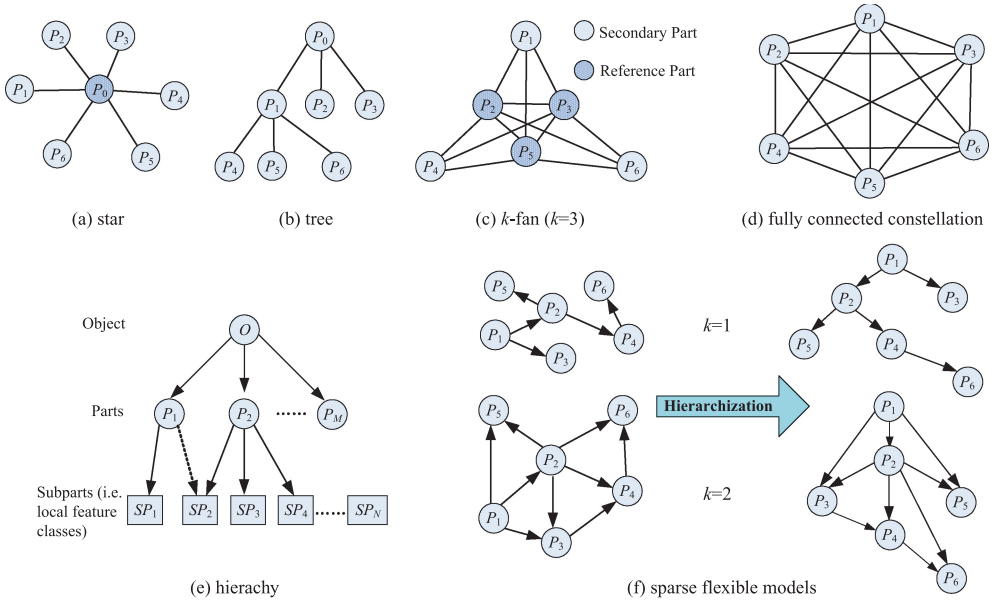


Fig. 8. Graphical representations of typical part topologies, adapted from Carneiro and Lowe [2006] with permission.

between any pair of parts. (v) Directed Acyclic Graphs (DAG), including the *hierarchy* (see Figure 8(e)) [Bouchard and Triggs 2005] and *Sparse Flexible Model* (SFM) (see Figure 8(f)) [Carneiro and Lowe 2006], all have some kind of (roughly) hierarchical structures (as revealed by hierarchizing the original SFM graphs; see Figure 8(f)). Note that the model parameter k in the k-fan depicts the number of reference parts the model contains, while in an SFM [Carneiro and Lowe 2006] it defines the maximum number of closest neighbors a part may geometrically depend on.

Generally, complex models are more flexible, and hence are capable of dealing with large object deformations. However, they have more free parameters, which make model learning and inference more difficult or even intractable. Therefore, some trade-offs between complexity and flexibility must be made. Recent works, such as Andriluka et al. [2009] and Felzenszwalb et al. [2010b], have shown that the simplest star and tree structures are quite effective even for some complex articulated classes such as human body and horse. Hence, we mainly focus on these two groups of part-based models in the following. In addition, we discuss the most general part-based models, namely *grammar-based* models, which represent objects using variable hierarchical structures.

Star-Structured Models. The key assumption underlying star-structured models is that the location of each part depends only on a central reference part, that is, parts' locations are mutually independent. Thus, given the object center, we can learn from a set of annotated objects a separate relative location distribution for each part. Conversely, when the observed location of a part is given in a novel test image, its learned relative location distribution can be inverted to provide a probability distribution of the object center location. Then, the probability of each point in the input image being an object center can be computed by summing up all parts' prediction probabilities for that point. Finally, the maximally probable point is taken as the final prediction of the object's center.

Early star-structured models, such as Fergus et al. [2005] and Leibe et al. [2004, 2006, 2005], commonly use small image blocks or patches as part appearance descriptions and the relative locations between parts and the object centroid to encode the star topology. In the training stage, a large number of candidate patches are first selected from a large set of object images, mainly picked from the object regions in these images, to construct a visual codebook for each target category, and the relative location of each patch to the object centroid is also encoded. These patches are then matched to a validation set that consists of both object and background images. Those mostly matched with background regions are discarded. In the testing stage, all codebook patches are matched to the input image to locate potential object centroids in a *voting* manner, which is detailed in Section 4.2.

To improve the invariance of part descriptions to lighting, color, and texture changes, shape cues are employed. Popular shape-based part descriptions include contour or boundary fragments [Opelt et al. 2006; Shotton et al. 2005, 2008a], *k Adjacent Segments* (*k*AS) [Ferrari et al. 2008; Ravishankar et al. 2008], edgelets [Wu and Nevatia 2007c], etc. The use of shape cues often makes these models perform better on some shape-based categories such as human body, horse, and cow. However, their part topology representations, which are similar to those of their predecessors, are still too rigid. For instance, Opelt et al. [2006] use the vectors from the boundary fragments to the object centroid to encode their relative locations. Then, they take a small circular region around the end of each vector as a possible region of its predicted object centroid to allow for part deformations. This implicitly restricts the displacement of each part relative to the object centroid to be nearly identical. However, different object parts, such as the head and leg of a human body, always have different ranges of relative displacements with respect to the object centroid.

The recent star-structured model of Felzenszwalb et al. [2008, 2010b], based on the PS framework, relaxes the aforementioned restriction by defining the cost of the part's relative displacement to be a parametric quadratic function, and by learning distinct parameters for each part. Specifically, the model combines a window-based global object template, which is called the *root filter* and is equivalent to the window-based model of Dalal and Triggs [2005], matched at a coarse image scale with a set of star-structured local part templates (each is called a *part filter*) matched at a finer scale. Thus, it can be regarded as a mixture of window-based and part-based models and hence will be revisited in Section 3.2.3. Each part in this model is described by a 5-tuple, $P_i = (F_i, v_i, s_i, a_i, b_i)$, where F_i is the HOG feature filter, v_i is a 2D vector specifying the center location of the bounding box of part P_i relative to the root position, s_i is the size/scale of this box, while a_i and b_i are the parameters of its quadratic deformation cost function. Since the HOG-based part description is usually more robust to part deformations than patch- or shape-based ones, this model needs fewer parts (fixed to 6 for each 2D model by the authors) than its predecessors. In addition, it is learned with less human supervision using the Latent SVM (LSVM) framework, in which part locations are defined as latent variables (refer to Section 5.4) to avoid manual labeling of parts in the training stage. The deformability of the star structure, the discriminative power of HOG, and the less supervision needed by LSVM have led the resulting model to receive much attention from researchers and to achieve much success in the recent PASCAL VOC contest [Everingham et al. 2010]. However, the model has several notable shortcomings: (i) Its 5-tuple part description contains no parameter to describe the orientation changes of the parts relative to the root, even though these changes are very common. (ii) Because of the star structure used, the proposed model can only describe part-root dependencies; but, for articulated classes such as the *human body*, some parts are more dependent on their neighbors, for example, the lower arm depends more on the connected upper arm than on the root. (iii) To support multiview detection,

the model is further extended to a mixture of 2D models, each of which is trained independently under a distinct viewpoint. However, since an object part is often visible under different viewpoints, its description should be shared among these viewpoints to make the mixture model more concise and the detection at novel viewpoints easier.

Another interesting star-structured model is that proposed by Gu et al. [2009], which uses regions generated by a bottom-up segmentation system [Arbelaez et al. 2009] as parts and describes their appearances using some HOG- or GIST-like locally pooled histograms capturing the regional shape, color, and texture features. The location of a region in an exemplar object is encoded in terms of the bounding box of that object. Thus, parts are geometrically independent of each other and the model is essentially a star-structured one. This is also evidenced by the fact that, with this model, objects in an input image are located using a voting scheme (refer to Section 4.2).

Tree-Structured Models. Compared with star-structured models, the tree-structured ones can represent richer dependencies including connections between any non-leaf node and its children. In particular, since a tree structure provides a natural way to capture the kinematic constraints between parts of a human or animal body, for example, the lower leg is connected only to the corresponding upper leg, tree-structured models are widely used to model articulated categories such as the human or animal body. However, the increase in flexibility is accompanied with an unavoidable increase in the number of free parameters, and hence an increase in fitting cost. Thus, research on using tree-structured models should focus more on improving the computational efficiency.

The PS model is always used with tree structures as typified by the model proposed by Felzenszwalb and Huttenlocher [2005], in which the energy function is embedded into an MRF-like generative framework to ease the parameter estimation process. It uses 10 parts to model the human body. Each part is represented by a rectangle parameterized by its center position, size, and orientation parameters. To enhance computational efficiency, they further propose using a spatial transformation to capture the location of each part and restrict the pairwise connection term in the energy function (given at the beginning of Section 3.2.2) of PS, that is, $d_{ij}(l_i, l_j)$, to be the Mahalanobis distance of the functions of the transformed part locations. Similar tree-structured part models are adopted in some class-specific segmentation methods, such as OBJ CUT [Kumar et al. 2005], and some 3D human body segmentation and pose estimation methods, such as PoseCut [Bray et al. 2006]. The recent method of Andriluka et al. [2009], which also uses a tree-structured PS model, is quite good for modeling the human body in different poses, including some extremely complex postures such as those in acrobatic exercises. The proposed model adopts a different number of parts to support the detection of the upper body, the full body, and pedestrian. Parts are defined as rectangles similar to those in the aforementioned PS model of Felzenszwalb and Huttenlocher [2005]; but their appearances are described using densely sampled *shape context descriptors* [Belongie et al. 2001] instead. Another recent example is the part-template tree of Lin et al. [2010, 2007] for human detection and segmentation, which consists of only six parts organized in a four-layer template tree. The first layer in the tree is empty, and the others correspond to the head-torso, upper legs, and lower legs, respectively. However, since the arms are ignored and the inter-part connections are fairly rigid, this simple model is relatively weak in dealing with pose changes.

Note that, to be tractable and physically plausible, tree-structured models often use a relatively small number (commonly less than 10) of *semantically meaningful* parts to represent an object, that is, each part in a tree-structured model often corresponds to a physical part of the target object. This kind of part definition is different from star-structured models described earlier.

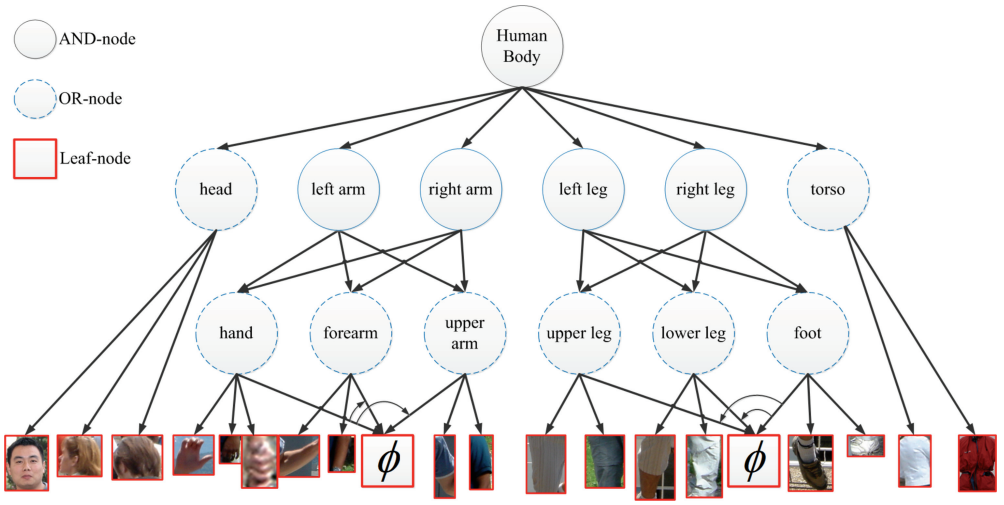


Fig. 9. The AND-OR graph for the *human body* category. ϕ represents an empty leaf node. See text for more details.

With respect to training, tree-structured models commonly need images with elaborate labeling of object parts, which is quite time consuming and labor intensive. A notable exception is the three-layer tree-structured model proposed by Zhu et al. [2010] recently, which is an extension of the two-layer star-structured model of Felzenszwalb et al. [2008, 2010b]. The authors adopt the latent structural SVM algorithm [Yu and Joachims 2009], a framework very similar to the LSVM, to learn the model from labeled training examples with objects but not their parts covered by bounding boxes. The model demonstrates superior performance and has been adopted in some newly developed detectors, such as Pedersoli et al. [2011].

Before closing this section, we summarize the distinction between the relative cost and the representation power of star- and tree-structured models. A star-structured model can be viewed as a special tree-structured one with only two layers. The shallower structure makes it simpler, that is, it needs fewer parameters to fully specify the model, and hence is more efficient for learning. However, the complexities of matching a star- and tree-structured model both have an order of $O(N^2P)$, where N is the number of locations in the input image and P the number of parts in the model. As for the representation power, since tree-structured models can capture richer inter-part dependencies, they are usually more powerful than star-structured ones, in particular for describing and handling pose changes. Presumably this claim can be verified by two interesting phenomena in recent research: (i) most methods aiming at capturing pose changes use tree- rather than star-structured models; and (ii) deep structured tree models such as those in Pedersoli et al. [2011] and Zhu et al. [2010] often outperform their star-structured counterparts, such as the model of Felzenszwalb et al. [2008, 2010b], on datasets rich in intra-class pose variations, for example, PASCAL VOC [Everingham et al. 2010].

Grammar-Based Models. Grammar-based models [Zhu and Mumford 2006], also called *compositional models* in Girshick et al. [2011] and Xu et al. [2008], generalize part-based models by representing objects using variable hierarchical structures.

The grammar is usually embodied in a hierarchical AND-OR graph to represent object categories (as illustrated in Figure 9), which consists of three types of nodes, that is, AND nodes, OR nodes, and leaf nodes. An AND node represents a composition of

its children (i.e., subparts). An OR node is a "switch node" for possible choices of the components (i.e., only one of its children will be selected for each object instance) and hence allows the graph to adapt to different objects of the same class. Leaf nodes relate directly to some low-level feature units such as image patches or visual descriptors extracted from the input image. Thus, grammar-based models allow for explicit structural variations. They also provide a natural framework for sharing information and computation between different object classes. For example, different nodes in a model or even in different models might share reusable parts.

However, the representation power of grammar-based models (embodied in AND-OR graphs) comes at a price of increased computational demands for performing inference and learning. It is still not clear how to rapidly search over the large space of possible configurations of an AND-OR graph. Hence, efficient learning and inference algorithms are in urgent need to enable grammar models in practical use. Some recent works, such as Chen et al. [2007] and Zhu et al. [2011], have taken steps toward this direction.

3.2.3. Mixed Models. An important finding of past research is that powerful part-based models like tree-structured ones, which can adapt to variations in object pose, only help in the presence of strong pose variations, such as persons in sport scenes, while the most successful model to date for some less deformable categories such as pedestrians, who are usually standing or walking upright, is still a window-based one using locally pooled features such as HOG [Walk et al. 2010]. In other words, these two types of models are suited in different application scenarios. Therefore, instead of separating into window- and part-based representations, a more appropriate way may be to combine them to exploit their relative merits. We call the resulting approach a *mixed model*. A prominent example is the star-structured model of Felzenszwalb et al. [2008, 2010b]. As aforementioned, the root filter in that model is a window-based, global template and the other parts form a part-based model. However, this combination scheme is quite straightforward and rigid as: (i) the part filters are placed at twice the spatial resolution of the placement of the root, and (ii) every filter in the mixed model needs to be matched when detecting an object. Actually, when an object is very small in the image, we may not locate its parts even with our eyes. Based on this observation, Park et al. [2010] propose a resolution-dependent combination of window- and part-based models. The structure of their proposed model can vary adaptively according to the size of the object to be detected. Specifically, the model degenerates into the window-based one of Dalal and Triggs [2005] when detecting small objects, such as pedestrians shorter than 50 pixels; and turns into the part-based model of Felzenszwalb et al. [2008, 2010b] when detecting larger objects, such as persons taller than 100 pixels. The adaptively variable structure, which is in agreement with the spirit of grammar-based models, makes the mixed model very effective for describing articulated objects and helps the corresponding detector to achieve first ranking in the challenging Caltech pedestrian detection evaluation [Dollar et al. 2011].

3.2.4. Models for Multiview Detection. As discussed in Section 2.2.1, objects are often photographed under different viewpoints. This motivates researchers to explore the problem of multiview object detection [Christoudias et al. 2008; Hoiem et al. 2007; Razavi et al. 2010; Seemann et al. 2006; Thomas et al. 2006; Zhang and Chang 2006], which is further generalized to 3D object detection [Liebelt and Schmid 2010; Liebelt et al. 2008; Savarese and Li 2007; Yan et al. 2007].

To support multiview or 3D detection, a common idea is to make use of multiview or 3D categorical appearance models by matching their 2D projections in different viewpoints with the input image to predict the optimal viewing angle for detection. Among the structured models discussed before, the star-structured part models are widely used and have been extended to 3D, for example, Razavi et al. [2010], Seemann

et al. [2006], and Thomas et al. [2006]. Another notable example is the multiview model of Felzenszwalb et al. [2008, 2010b], which is composed of a few 2D star-structured models, called *components*, trained under different viewpoints. Several other detectors, such as Savarese and Li [2007] and Zhang and Chang [2006], build more complex 3D part models containing connections between neighboring parts and overlapping viewpoints. In addition, graphical or synthetic 3D models have also been employed to support multiview detection, as done by Hoiem et al. [2007a], Yan et al. [2007], Liebelt and Schmid [2010], and Liebelt et al. [2008].

3.2.5. Comparison of Window- and Part-Based Models. We compare globally and locally pooled window-based models with part-based ones in terms of their capabilities of dealing with intra-class variations when detecting structured object classes. As mentioned in Section 2.2, for these classes, occlusions, background clutter, and changes in size/scale, pose, or viewpoints are the main issues of intra-class variations to be considered. (i) All of these models can easily handle size changes by multiscale matching. (ii) Since no layout information is included, globally pooled window-based models are invariant to planar rotations and pose changes if invariant local features are used, while locally pooled ones are sensitive to them. Albeit part-based models are also not invariant to pose changes, they always can tolerate these changes to a larger extent compared with locally pooled window-based models due to their inherent deformability. In addition, the relative locations of parts of an object detected with part-based models can be used to describe the object pose, and to support pose estimation, which is confirmed by the state-of-the-art pose estimator proposed recently by Yang and Ramanan [2011]. Hence, part-based models should be regarded as *pose aware* rather than pose invariant. (iii) The adoption of invariant local features leads all these models to be robust to small viewpoint changes. Moreover, they can be extended to support multiview detection by combining multiple 2D models trained under different viewpoints. (iv) Since spatial pooling over larger extents can help to achieve higher levels of invariance, globally pooled window-based models, compared with locally pooled ones, are usually more robust to partial occlusions. In addition, since a portion of the parts in a part-based model is often sufficient to support a valid object hypothesis, part-based models usually can tolerate partial occlusions better than, at least locally pooled window-based ones. Moreover, by training different decision thresholds for different combinations of model parts, their robustness against occlusions can be notably improved and the occluded parts can be located as done in the recent work of Girshick et al. [2011].

3.3. Unstructured Models

The models discussed in the previous section capture the internal geometric structures of objects of interest and use them together with appearance information to discriminate different categories. However, there are also many categories, such as grass, water, and cloud, with no obvious inter-object part-wise correspondence, which renders structured models ineffective. Instead, they are often described using mid-level representations computed over pixel neighborhoods or (object) component regions. Since this kind of categorical representations does not encode structural information explicitly, we call them *unstructured models*.

In general, there are two different ways to model unstructured objects. The first is to encode appearance information at the pixel level, that is, to compute a visual descriptor for each pixel over some local neighborhood surrounding it. The resulting descriptors can be used for pixel-wise categorization. For computational efficiency, input images are usually down-sampled using some regular grid prior to pixel-level descriptor computation. This representation is simple and has proven effective in many state-of-the-art systems, such as Larlus et al. [2010], Serre et al. [2007b], and Shotton

et al. [2008b, 2006, 2009]. However, fixed-size neighborhoods may span beyond object boundaries and hence make the resulting descriptors unreliable, which often causes inaccurate boundaries and "holes" to the detected objects. Moreover, the descriptor computed over a pixel neighborhood always can capture only the appearance of a (small) portion of the object of interest. Thus, it is often too local to distinguish between adjacent objects of the same class as well as to ensure the (categorical) label consistency among neighboring pixels.

The second way is to encode object appearances based on its component regions. To that end, some bottom-up oversegmentation process, for example, superpixel extraction [Ren and Malik 2003] or region decomposition [Gould et al. 2009a], is first conducted on the input image to produce a set of small, coherent segments, which are expected not to be larger than any object part. Then, the appearance of each region is summarized to form the region-level representation used for region-wise categorization. Using regions as the elementary units is (partly) helpful for ensuring label consistency between neighboring pixels and for avoiding the inclusion of background pixels. However, the basic assumption underlying these methods is that all the pixels of a particular segment belong to the same object [He et al. 2006]. This always does not conform to experimental observations, particularly when the segments are relatively large [Kohli et al. 2008]. Albeit combining multiple different oversegmentations can help to reduce errors caused by any single segmentation [Gould et al. 2009b], appropriate region selection remains an open problem.

Since an object often contains multiple dissimilar regions, pixel- and even region-level representations can capture only a part of its appearance. Thus, they are insufficient for reliably discriminating different categories. To alleviate this problem, a more complex, usually a parametric energy (or matching cost) function, may be defined as a sum of unary, pairwise [Shotton et al. 2006], and higher-order potentials [Kohli et al. 2008]. The unary potentials are often defined based on the pixel- or region-level representation to capture the likelihood of a pixel or region belonging to a certain category. The pairwise and high-order potentials are defined to ensure the label consistency between neighboring pixels (or regions) as well as to encode contextual information at different scales (see Section 3.4). It is noteworthy that the training of this energy function often needs more human supervision because each pixel in the training images must be labeled as one of the categories of interest or as the background, which is different from the training of structured models requiring object or part annotations with bounding boxes only.

Albeit unstructured models can also be used to recognize structured objects, they are widely accepted to be more appropriate for amorphous ones, while structured models are more powerful for shape-based ones. Hence, many recent works, for example, Gould et al. [2009b], Ladicky et al. [2010], Maire et al. [2011], and Serre et al. [2007b], combine them together to enhance localization accuracy and even to implement a *complete scene understanding* system, which requires not only the pixel-wise segmentation of an image, but also labeling of object instances of a particular class [Ladicky et al. 2010].

3.4. Contextual Information Description

The earliest work in employing context in computer vision can be traced back to the work by Strat [1993], who states context as "any or all information that may influence the way a scene and the objects within it are perceived." Since then, it has been widely accepted that context is important to many machine vision tasks. As well, psychophysical research has also shown the importance of context in the human visual system [Oliva and Torralba 2007]. Despite the wide use of "context" in much computer vision research, Divvala et al. [2009] were the first to identify the lack of a clear

definition or meaning of it. They try to provide their interpretations of context based on the early psychophysical findings of Biederman et al. [1982].

To study the specific benefits of context to object class detection, let us first briefly revisit the famous hypothesis of Biederman et al. [1982], which states that there are five categories of object-environment dependencies that can characterize the organization of objects in real-world scenes. The five categories include "(i) *interposition*: objects interrupt their background, (ii) *support*: objects often rest on surfaces, (iii) *probability*: objects tend to be found in some environments but not others, (iv) *position*: given an object in a scene, it is often found in some positions but not others, and (v) *familiar size*: objects have a limited set of sizes relative to other objects." From this hypothesis, we can easily discover the benefits of context to object detection. First, according to the *probability* constraint, context can be helpful for object categorization, especially when object appearance alone is insufficient for decision. For example, yellow and fist-sized spherical things on a tennis court are more likely to be tennis balls, while those with a similar appearance in a tree are more likely to be lemons. Second, according to the *interposition, support*, *familiar size*, and particular *position* dependencies, context can provide useful cues for object localization. As an example, a keyboard is more likely to be near a monitor, but not a bathtub. Third, according to the *interposition*, *support*, and *familiar size* dependencies, context can also help to delineate the spatial extent of objects of interest. In short, contextual information, which is an important complement to objects' appearances, is very useful for object class detection.

Based on the preceding analysis, the concept of *contextual interaction* can be further understood as the probabilistic dependency of the presence of objects, their appearances, and locations on their environments. Note the word "probabilistic" here emphasizes that the context may be ambiguous and unreliable, and thus may not work for every scene and for every object [Zheng et al. 2009].

Studies in cognitive neuroscience suggest that our perceptual processes are organized hierarchically so they proceed from global structures to more and more detailed analysis. Specifically, our initial conscious percept follows a feedforward path to recognize the scene category at a glance, identifying "forest before trees," which then initializes the feedback processes that focus attention to perceive details of the scene [Hochstein and Ahissar 2002]. In the feedback processes, different contextual cues, including global scene information as well as the inter-part interactions inside the scene, always direct the visual search for objects of interest in the scene [Oliva and Torralba 2007]. Hence, the two levels of context that are of particular importance for object class detection are the *global* and *local context*. Therefore, similar to the survey of Galleguillos and Belongie [2010], we organize commonly used contextual cues into these two levels. To give a more comprehensive overview, we further subsume different sources of context summarized in the survey of Divvala et al. [2009] into the corresponding levels. In addition, we include in the overview recent works reported after 2009, which are not covered by the aforementioned two surveys.

3.4.1. Global Context. Global context refers to global information about the whole scene, for example, the scene category, which can serve as cues for object categorization and detection. For example, a street view image always implies that possible objects in it mainly include cars, pedestrians, buildings, traffic signs, etc.

Typical global context includes: (i) *scene context* includes the image-level statistics capturing the "gist" of a scene [Oliva and Torralba 2001, 2006], the scene category identified through BoF classification [Shotton et al. 2008b; Verbeek and Triggs 2007b], etc. (ii) *Imaging condition context* includes the intrinsic [Luo et al. 2006] and extrinsic camera parameters [Hoiem et al. 2008], the various parameters of scene illumination (e.g., sun direction [Lalonde et al. 2010], cloud cover, shadow), weather conditions (e.g.,

temperature, season, rain, snow, fog, and haze [Narasimhan and Nayar 2002]), and geographic parameters such as the actual location of the image (e.g., GPS), the terrain type (e.g., tundra, dessert, ocean), and the land use category (e.g., urban, farm) [Hays and Efros 2008], etc. (iii) *Temporal context* contains temporal proximity information [Divvala et al. 2009], which is often more relevant to images in a video and typically can be described using the time of capture [Gallagher et al. 2008], nearby frames in the same video, or videos captured from similar scenes [Liu et al. 2008], or contextualized histograms [Ni et al. 2009]. (vi) *Cultural context*, as defined by Divvala et al. [2009], refers to the biases of how we take pictures (framing [Simon and Seitz 2008], focus, subject matter), how we select datasets [Ponce et al. 2006a], and even how we name persons [Gallagher and Chen 2008]!

3.4.2. Local Context. Local context denotes the interactions among different parts of the same scenes, in particular, those between an object and its surrounding pixels, regions, or objects.

Generally, there are three different types of local context. (i) *Semantic context* refers mainly to the co-occurrence pattern of different objects [Rabinovich et al. 2007] or object parts [Karlinsky et al. 2010] in the same scene, or even the social relations among them [Wang et al. 2010a], which can be captured using the object-graph descriptor [Lee and Grauman 2010], pairwise or high-order potentials in CRF [Xiang et al. 2010], or the textual tag list of the image (if available) [Hwang and Grauman 2010]. However, note that if semantic context indicates the kind of event, activity, or scene category being depicted [Li and Fei-Fei 2007], then it is essentially a type of global context. (ii) *Scene structure context* captures the 3D geometric structure of a scene [Hoiem et al. 2008], which can be used to reason about occlusions [Hoiem et al. 2007b]. (iii) *Local surroundings context* refers to the interactions between the region of interest and its surrounding pixels/patches. The trick used by Dalal and Triggs [2005] to include surrounding pixels through increasing the detection window size is essentially to capture this type of context. Recently, the polar geometric histogram [Zheng et al. 2009] has been proposed to encode the local surroundings context explicitly. More commonly, local surroundings context is captured using pairwise potentials in MRF/CRF-based methods, for example, Shotton et al. [2006]. Furthermore, the shape and boundary of an object is also defined using its surroundings, namely in terms of local surroundings context.

3.4.3. Summary. As a kind of probabilistic dependencies, both global and local context are more suitable to be described by probabilistic models. In addition, making combinatory use of multitype, multilevel, and even multidomain (e.g., cultural, geographic, meteorological) contextual information has been shown promising to enhance detection performance in some recent works, such as Galleguillos et al. [2010], Gonfaus et al. [2010], and Lin and Davis [2010]. However, one of the main problems troubling computational recognition approaches using contextual information is the lack of simple representations of context and efficient algorithms for the extraction of such information from the visual input. Hence, more efforts toward this direction are still much needed.

4. LOCALIZATION STRATEGIES

The task of localization is to search in the input image for regions that best match the learned appearance models of the categories of interest. In the simplest case of class-specific detection, object localization can be interpreted as the process of searching for the global maximum or local maxima of the matching score $S(c, R)$ with respect to the region of interest R. However, a region of arbitrary shape may have too many free parameters, which make the search process very time consuming and even intractable. To address this problem, a number of approximation schemes have been proposed.

Among them the most popular methods include the *subwindow search*, *voting*, and *localization through segmentation* as described in the following. Note that the first two are *exclusive* for structured models, while the last one can be applied to both structured and unstructured models. Moreover, the subwindow search is frequently integrated into a segmentation-based localization process to further improve the localization accuracy of structured objects.

4.1. Localization through Subwindow Search

The subwindow search strategy, also called *sliding window*, performs object categorization over all possible subwindows in the input image to locate potential objects. The key idea is to replace R in the matching score $S(c, R)$ with a subwindow W of a prespecified simple shape to reduce the number of free parameters. Hence, the matching score should be denoted as $S(c, W)$.

In the simplest case, for example, in Dalal and Triggs [2005], a subwindow W is a rectangle of a given size. Then, two basic variables to search are the 2D coordinates of the center or a corner of W. In addition, to detect objects of the same class but with different sizes, the scale factor of W is required. Then, the subwindow search is often performed in a multiscale pyramid of the input image. The size and the aspect ratio (defined as the ratio of width to height) of W depend on the dataset used during (window-based) model development and are often chosen after training. In addition, since the positions surrounding the real object locations always respond positively to the model as well, some postprocessing step is needed to suppress these nonmaximal detections to reduce false positives. Two dominant NonMaximal Suppression (NMS) methods are the mean-shift mode estimation [Dalal 2006] and the pairwise MAX scheme [Felzenszwalb et al. 2008] which discards the less confident of every pair of detections that overlap sufficiently.

The classical subwindow search strategy described before is simple, intuitive, and proved effective. However, it constantly suffers two severe limitations. The first is its low computational efficiency. For an $n \times n$ image, if three variables are searched, that is, the 2D center/corner coordinates plus a scale factor, then the number of subwindows needed to be categorized amounts to n^4! The second limitation is because of the difference between the predefined shape of the subwindow and that of the detected object (which has been discussed previously in Section 3.2.1). Consequently, background pixels are often included in the subwindow as well, which affects the accuracy of window-based categorization, and hence the accuracy of localization. Later work mainly focuses on overcoming these two limitations.

Several approaches to speeding up the subwindow search have been presented in the literature. The first is to reduce the search range using the branch-and-bound algorithm, as typified by the Efficient Subwindow Search (ESS) method proposed by Lampert et al. [2008], which shows that, by dividing the search space into disjoint subspaces and by designing an easily computable bound of the matching score, a priority queue can be used to notably reduce the search time. However, ESS restricts the subwindow to be rectangular and the matching score to be the sum of entries such as the contributions of extracted visual descriptors or pyramid levels. Moreover, its performance highly depends on the presence of objects in the input image and will degrade to an exhaustive search when no object is present. Hence, An et al. [2009] propose the improved ESS, which is based on the linear-time Kadane's algorithm for 1D maximum subarray search [Bentley 1984]. These ESS methods have a complexity linear in the histogram dimension, and thus still run slowly when high-dimensional histograms are involved, which are very common for detectors based on BoF or HOG features. To solve this problem, Wei and Tao [2010] present an efficient method that

has a constant complexity in the histogram dimension by harnessing the spatial coherence of natural images and by computing the matching score in an incremental manner.

The second way of improving efficiency is to reduce the search range using contextual information [Torralba et al. 2003]. For example, the recent work of Hwang and Grauman [2010], mentioned in Section 3.4, mines semantic context from the accompanied textual tags of Web images and applies them to speed up the window search process. Another notable example is the work of Alexe et al. [2010], who propose a category-independent *objectness* measure based on the general characteristics of objects including their local surroundings context (see Section 3.4.2) captured by edges and superpixels, and use this measure as a location prior to provide a smaller search range.

The third way of improving efficiency is to reduce the search range using some object-hypothesis-generating approaches, namely to perform a *selective search* [Van de Sande et al. 2011] instead of an exhaustive search. A representative is the Constrained Parametric Min Cuts (CPMC) algorithm developed by Carreira and Sminchisescu [2010]. It can generate a set of figure-ground segmentations, among which foreground segments covering full objects are extracted with high probability. Similarly, Endres and Hoiem [2010] propose a method to produce a set of ranked regions such that the top-ranked ones are likely to be good segmentations of different objects. Both methods can generate a small number (typically 10–100) of highly probable object locations, which can then be used as object hypotheses and thus notably reduce the search space of the subwindow search, as confirmed by Carreira et al. [2011] using the CPMC algorithm. However, both of them use a powerful but computationally expensive contour detector [Arbelaez et al. 2009], which makes the corresponding object hypothesis generation process very time consuming, and hence limits their applicability with respect to large datasets. Instead, Van de Sande et al. [2011] propose a fast hierarchical grouping algorithm to form a region tree, in which all segments or their tight bounding boxes are considered as potential object locations. Hence, a larger number (1000–10000) of hypotheses are generated for the subwindow search, which is still far less than those that would be evaluated in a brute-force search process.

An additional way of improving efficiency is to use the cascaded search, which removes a large amount of relatively easy, nonobject regions as soon as possible using a binary decision tree. Early approaches in this category include the cascaded boosting classifier of Viola and Jones [2001] and the coarse-to-fine face detector of Fleuret and Geman [2001]. Recent examples include the cascaded detector based on part-based models [Felzenszwalb et al. 2010a] and the three-stage classifier proposed by Vedaldi et al. [2009].

Besides reducing the search space, the integral image [Viola and Jones 2001] or the integral histogram [Porikli 2005] can be further used to significantly speed up the feature computation over *rectangular* subwindows, for example, in Lampert et al. [2008], Sabzmeydani and Mori [2007], Yan [2009], Zhu et al. [2006], and Wang et al. [2009b].

With respect to the second limitation of the window search strategy, polygonal [Yeh et al. 2009] and even arbitrarily shaped subwindows [Zhang et al. 2010] have been proposed as mentioned in Section 3.2.1. However, a more complex window shape requires more parameters, and hence speedup is of particular importance. Recently, a branch-and-cut algorithm [Vijayanarasimhan and Grauman 2011] has been proposed to support efficient localization using irregularly shaped subwindows and additive matching score functions.

Fig. 10. An illustration of the voting strategy for object localization. Best viewed in color. The images are selected from the Caltech Motorcycles dataset with permission from Pietro Perona and Rob Fergus. The dataset can be found at http://www.vision.caltech.edu/html-files/archive.html.

Note that the subwindow search is a fairly generic localization strategy which can be used with not only window-based models but also part-based ones, for example, Felzenszwalb et al. [2010b].

4.2. The Voting Strategy

Different from the subwindow search, the voting strategy is designed exclusively for part-based models, particularly star-structured ones. Since such a model consists of two key components, namely a set of parts and their topological relations, its matching score $S(c, R)$ is equal to the sum of the matching scores of the parts and the topological conformity measure. The voting strategy maximizes $S(c, R)$ in two stages: The first stage finds the best matching locations in the input image for all model parts, and hence maximizes the sum of the matching scores of the parts. The second stage maximizes the topological conformity measure by searching for the best topological hypotheses in the Hough voting space cast by the parts detected in the first stage. Thus, the voting strategy is essentially a greedy approach.

Object localization using the voting strategy is illustrated in Figure 10. As described previously in Section 3.2.2, each part in a star-structured model is often encoded using multiple patches or fragments selected from training examples, by which the locations relative to the object center are also provided (see Figure 10(a) for example). To localize potential objects in an input image, all part patches are matched to it. As shown in Figure 10(b), each matched patch then gives a prediction of the possible locations of the object center in the input image. The final prediction of the object center is determined by the location with the maximum number of votes (see Figure 10(c)). Because each model patch takes its own matching location in the input image as the reference point to locate the potential object centers, the voting strategy is robust to translation and in-plane rotation of objects in the testing images. In addition, scale invariance can be achieved by employing scale-normalized part descriptions, for example, the scale-normalized contour fragments in Shotton et al. [2008a] or the scale-invariant keypoints in Leibe et al. [2008], and by explicitly performing a multiscale hypothesis search. Sometimes, a figure-ground mask corresponding to each model patch is projected to the input image to get top-down segmentation results as in Leibe et al. [2004, 2005] and Marszałek and Schmid [2007].

As mentioned before, the voting strategy is a greedy approach. Thus, it may not give the real optimal solution because the best part matching results may not give rise to the best topological hypothesis. Moreover, part matching is commonly implemented using an exhaustive search, which is computationally expensive. To circumvent the first problem, a relaxed part matching is more preferable, by which a single part can be matched to multiple locations in the input image to enlarge the resulting topological voting space [Leibe et al. 2004, 2006]. To enhance the part matching speed, the generalized distance transform and the fast chamfer match can be used [Opelt et al. 2006; Shotton et al. 2005].

4.3. Localization through Segmentation

As discussed in Section 2.1, a successful segmentation automatically leads to a perfect localization. This fact has motivated many researchers to develop localization methods based on segmentation.

The most straightforward way is to localize objects through a semantic segmentation process, in which a pixel-level unstructured representation (refer to Section 3.3) is used to perform pixel-wise categorization. However, as discussed in Section 3.3, due to the locality of the adopted representation, the segmentation results often suffer "holes" inside objects and very inaccurate object boundaries. Hence, additional constraints are imposed under a probabilistic framework such as the CRF [Shotton et al. 2006] or the MRF [Verbeek and Triggs 2007a] to ensure the label consistency between neighboring pixels as well as to incorporate contextual cues. Though these methods can localize both structured and unstructured objects, they are less effective for structured objects than the subwindow search or the voting strategy because of the unstructured models. Moreover, also due to the locality of pixel-level representation, these localization methods often cannot distinguish between adjacent objects of the same class.

Another method is to perform semantic segmentation and object localization simultaneously through region-wise categorization. Specifically, the input image is first oversegmented into a set of pieces, for example, superpixels [Fulkerson et al. 2009; He et al. 2006], or small coherent regions [Gould et al. 2009b]. The appearance of each of these regions is summarized using some mid-level representation (refer to Section 3.1.3). Then, a region-level categorization is performed using the SVM with an RBF-χ^2 kernel [Fulkerson et al. 2009] or the multilayer perceptron [He et al. 2006]. Similar to localization based on pixel-wise categorization, region-wise categorization results commonly need to be further refined using some probabilistic framework such as the CRF to enhance the localization accuracy. Also note that initialization using single oversegmentation may make the recovery from errors introduced in the initial set of segments difficult. To address this problem, multiple segmentations can be used instead [Gould et al. 2009b], which reduces the risk of poor initialization.

A third method is to combine detectors with segmenters, which can not only improve the localization quality (from bounding boxes to pixel-wise labels), but also distinguish between adjacent instances of the same category. Earlier methods in this group, such as Borenstein et al. [2004] and Borenstein and Ullman [2008], focus mainly on combining bottom-up with top-down segmentation to enhance the localization accuracy. Recently, researchers have tried to combine detectors with *semantic* segmenters. Bottom-up segmentation of the input image is still used in the initial phase. A window-based detector is then employed to perform the subwindow search in the input image, and the results are input to a unified framework for scene labeling. The detector's outputs are often used in two different ways. One is to perform segmentation within the detected areas to more accurately locate the detected objects as done by Larlus and Jurie [2008]. However, both the true and false positive detections may be included in the final results. To overcome this problem, the other way is to use the detection results as soft constraints that can be rejected in a unified framework for image labeling as done in Ladicky et al. [2010] and Maire et al. [2011].

5. SUPERVISED CLASSIFICATION METHODS

The significance of supervised classification methods to object class detection is embodied in two main aspects: (i) Parameters contained in different types of appearance models are usually estimated during classifier training. (ii) Categorization and detection results of the testing images are commonly predicted by trained classifiers, and the value of the classifier function, namely the *classification score*, acts as the matching

score $S(c, R)$ in practice. Some additional benefits include the combination of multiple feature channels, reducing human supervision, etc.

5.1. Parametric vs. Nonparametric Methods

A classifier is called *nonparametric* if its classifier function contains no parameter; otherwise it is called *parametric*. More accurately, nonparametric classifiers are distribution free, that is, they make no assumption about the distribution of sample descriptors or the conditional distribution of categorical labels given sample descriptors. Thus, they do not need parameter estimation or inference. Nonparametric classifiers commonly used for object categorization and detection include the Nearest-Neighbor (NN) and the K-Nearest-Neighbor (KNN) methods. For example, Boiman et al. [2008] propose an NBNN (Naïve Bayes NN) classifier for image-level object categorization, which determines the categorical label of an input image by searching for the nearest neighbor from a set of labeled images in terms of the image-to-class distance (refer to Section 3.1.3). Liu et al. [2009a] adopt the KNN search for candidates in object categorization and in scene parsing based on dense scene alignment. Note that, though no parameter estimation or inference is involved, nonparametric methods still need human supervision, for example, to provide a set of labeled instances to compute the reference feature set for each category [Boiman et al. 2008].

In contrast, parametric methods usually make some parametric assumptions about the sample distribution or the conditional distribution, and thus need parameter estimation and inference, which are often complex and time consuming. Some parametric classifiers, such as AdaBoost and SVM, are very powerful as they achieve strong generalization capabilities through training, which is intrinsically a kind of data mining of the training set. It is noteworthy that in the majority of current literature, as well as in the portion cited in this survey, parametric classifiers are the most commonly used classification methods.

5.2. Probabilistic vs. Nonprobabilistic Methods

If the matching score $S(c, \mathrm{x}) \in [0, 1]$ and $\sum_c S(c, \mathrm{x}) = 1$, then it can be further defined as a posterior probability $p(c|\mathrm{x})$, and the classification can be performed by the so-called *Maximum A Posteriori* (MAP) criterion: $c^* = \mathrm{argmax}_c\, p(c|\mathrm{x})$. Classifiers defined in this way are commonly called *probabilistic models*, including the Bayesian framework [Alexe et al. 2010; Fei-Fei and Perona 2005; Lin 2009], the MRF [Bray et al. 2006; Kumar et al. 2005, 2010], the Hidden Markov Mode (HMM) [Ji et al. 2010], the CRF [Hoiem et al. 2007a; Levin and Weiss 2006; Rabinovich et al. 2007; Xiang et al. 2010], etc. The remaining classifiers are commonly called *nonprobabilistic* ones, including AdaBoost (or boosting) [Grabner et al. 2007; Opelt et al. 2006; Perrotton et al. 2010; Saffari et al. 2010; Shotton et al. 2005, 2008a, 2006, 2009; Torralba et al. 2004; Wu and Nevatia 2007a, 2007b, 2007c], the SVM [Csurka et al. 2004, 2006; Dalal and Triggs 2005; Felzenszwalb et al. 2008; Jurie and Triggs 2005; Lampert et al. 2008; Maji et al. 2008; Vedaldi et al. 2009], etc. Note that some probabilistic models, for example, those in Fei-Fei and Perona [2005] and Rubinstein and Hastie [1997], adopt a criterion different from MAP: $c^* = argmax_c\, p(x|c)$, where $p(\mathrm{x}|c)$ is the appearance description distribution of the c-th category. Since an appearance descriptor belonging to a specified category can be generated from $p(\mathrm{x}|c)$, models based on this criterion are generative ones, which are discussed in the next subsection.

Comparatively, probabilistic models can describe very complex dependencies, particularly ambiguous ones such as dependencies of objects on their context, in a straightforward way, while nonprobabilistic ones cannot. Besides predicting the best labels, probabilistic models can output additional information of posterior probabilities, the confidence scores of its predictions, etc. [Wang and Mori 2010]. However, their training

is usually complex and often needs a large training set. In contrast, nonprobabilistic ones, in particular kernel-based classifiers such as the SVM, can be trained using a very small set.

Note that these two families are not as distinct as they look apparently. Indeed, they can be unified, as nonprobabilistic models can always be transformed into probabilistic ones. For example, to combine the AdaBoost classifier for part detection with the probabilistic model describing the parts' dependencies (following the PS model), Andriluka et al. [2009] interpret the normalized margin of an AdaBoost classifier as a probabilistic likelihood.

5.3. Generative vs. Discriminative Methods

In general, there are two methods to obtain the posterior probability $p(c|\mathrm{x})$ for classification. The first one is to use a parametric model $p(c|\mathrm{x}, \theta_d)$, where θ_d denotes the model parameter vector. This kind of model is called a *discriminative model*, also known as a *diagnostic paradigm* historically. Nonprobabilistic models belong to this family, and as mentioned previously they predict the class label c conditioned on the given description x and can be transformed into probabilistic ones. Furthermore, some probabilistic models such as logistic regression and the CRF, are also discriminative ones. The second method is to model the joint distribution instead with parametric models $p(c, \mathrm{x}|\theta_g)$, which are called *generative models* in the literature and also known as *informative models* or *sampling paradigms* previously. Here, θ_g is the parameter vector. An alternative approach to formulate the generative models is to model the *category-conditional probability* $p(x|c, \lambda)$ and the *category prior* $p(c|\pi)$ separately, where λ and π are the model parameters. According to Bayes' theorem, we have $p(c, \mathrm{x}|\theta_g) \stackrel{\wedge}{=} p(c, \mathrm{x}|\pi, \lambda) = p(c|\pi)p(\mathrm{x}|c, \lambda)$, where $\theta_g = (\lambda, \pi)$. Hence, the posterior probability needed can be derived from the joint distribution as

$$p(c|\mathrm{x}, \theta_g) = p(c, \mathrm{x}|\theta_g) \Big/ \sum\nolimits_{c'} p(c', \mathrm{x}|\theta_g), \tag{1}$$

or from the category-conditional probability and the category prior as

$$p(c|\mathrm{x}, \theta_g) = p(c|\mathrm{x}, \pi, \lambda) = p(c|\pi)p(\mathrm{x}|c, \lambda) \Big/ \sum\nolimits_{c'} p(c'|\pi)p(\mathrm{x}|c', \lambda). \tag{2}$$

This means that a discriminative model can be derived from some generative one. These two models are often called a discriminative-generative pair, for example, naïve Bayes and logistic regression. Note that the derivation cannot be reversed!

As both generative and discriminative methods can be applied to classification, a never-ending question is on how to choose between them. Since the 1970's, a series of theoretical and empirical comparisons of these two families [Liang and Jordan 2008; Ng and Jordan 2002; Sutton and McCallum 2006; Ulusoy and Bishop 2006, 2005; Xue and Titterington 2008] have been reported. The interested reader can refer to Xue [2008] for a review of these comparisons. In this survey, due to the page limitations and the limited scope, we briefly summarize the main conclusions, which are instructive for choosing or for designing classifiers for object class detection or recognition: (i) While discriminative models enjoy lower asymptotic error rates, generative methods might be better when the number of training samples is limited [Ng and Jordan 2002]. (ii) Discriminative models can be derived from their generative counterparts as shown in Eqs. (1) and (2); while the reverse derivation is impossible. Moreover, through marginalization of generative models, one can obtain the marginal probability $p(\mathrm{x}|\theta_g)$, which can be trained with unlabeled samples and used to synthesize new input data (by sampling) or to detect abnormal data. Thus, generative models are more informative [Lasserre et al. 2006]. (iii) The generative classifiers learn the class densities, while

the discriminative ones learn the class boundaries without regard to the underlying group densities [Rubinstein and Hastie 1997]. This means that a discriminative model focuses the learning effort on the ultimate decision to be made (e.g., whether or not an object of interest is present in the input window), whereas a generative one further models variability about the category that may be irrelevant to the task. (iv) The performance of probabilistic models depends on the correctness of the modeling, the bias, the efficiency and consistency of learning, and the reliability of the training data [Xue and Titterington 2010]. Using more data tends to reduce variance, but at a cost of being more sensitive to *model misspecification*, which is the difference between the true distribution of the process generating the data and the distribution specified by the model [Liang and Jordan 2008].

From the preceding discussions, it is obvious that generative and discriminative models have some complementary properties, which motivates a number of researchers to exploit the best of both families. They propose the mixed discriminants [Enzweiler and Gavrila 2008; Rubinstein and Hastie 1997; Salzmann and Urtasun 2010], the Generative-Discriminative Trade-off (GDT) [Xue and Titterington 2010], hybrid methods [Grabner et al. 2007; Lasserre et al. 2006; Tu 2007; Zhang and Chang 2006], the integration of discriminative classifier into generative framework [Andriluka et al. 2009], etc.

5.4. Latent and Hidden Models

Latent or hidden variables are always employed by modern object class detection and recognition systems to mitigate the human burden of sample labeling, that is, to reduce human supervision. *Latent variables* are variables not directly observed but are inferred from other observed variables. They are always used to describe some intermediate concepts to complete the conceptual hierarchy to better explain some observations. Models aiming to explain observed variables in terms of latent variables are called *latent models*, among which the Probabilistic Latent Semantic Analysis (PLSA) [Hofmann 2001] and Latent Dirichlet Allocation (LDA) [Blei et al. 2003] model may be the most famous ones. LDA is a three-layer hierarchical Bayesian model, originally proposed for modeling collections of discrete data. Take text corpora modeling for example, where the observations are the words in a given set of documents, LDA posits that each document is a mixture over a set of topics which are unobservable and defined to be latent variables, and each topic is characterized by a distribution over words. By drawing an analogy between textual and visual contents, Fei-Fei and Perona [2005] adapt LDA to scene categorization. Specifically, they regard an *image* as a *document*, a *visual word* (or local *patch*) as a *word*, and introduce the intermediate concept *theme*, defined to be a latent variable, as the concept *topic* in LDA. In this way, they not only can complete the conceptual hierarchy explaining "scene category," but also can avoid manual labeling of the scene themes in training images, and hence can reduce the human burden of sample annotation. Differently, to discover objects and their locations in a set of unlabeled images, Sivic et al. [2005] treat object categories as topics, and hence an image containing objects of multiple categories as a mixture of topics. In addition, they use a different model, that is, the PLSA model. It is notable that their proposed method enables the construction of categorical models from an unsupervised analysis of images. Another latent model worth noting is the LSVM proposed by Felzenszwalb et al. [2008, 2010b]. The authors define the relative part locations in their star-structured part model to be latent variables which are estimated during training (see also Section 3.2.2). Hence, their proposed part model can be trained using images annotated with only bounding boxes of the entire objects but not of their parts, that is, *weakly labeled training samples*, which reduce human supervision significantly. A very similar while more general model is the Latent Structural SVM (LSSVM) [Yu and

Joachims 2009] mentioned previously in Section 3.2.2, which defines unavailable modeling information as latent variables and learns them, as well as the classifier, through Concave-Convex Programming (CCCP). Owing to its generality, LSSVM enjoys more applications in top-performing part-based detectors, particularly tree-structured ones such as Girshick et al. [2011], Pedersoli et al. [2011], and Zhu et al. [2010], to learn models from weakly labeled training samples.

Hidden variables can be regarded as special latent variables, which correspond to some aspects of physical reality and could be measured in principle, but may not be for practical reasons. Classifiers containing hidden variables are called *hidden models*. The latent variables defining part locations in the LSVM mentioned before are actually hidden ones which can be but have not been observed from training images because no parts' labels are available. Similarly, Wang and Mori [2009, 2010] adopt the hidden part model but apply the hidden CRF (HCRF) classifier to a different problem, namely human action recognition.

Latent and hidden variables are always estimated by some iterative algorithms, for example, stochastic gradient descent [Felzenszwalb et al. 2010b]. Hence, the training process is very time consuming. Moreover, the convergence to local maximum depends on the initialization. Thus, in some sense, the reduction of human labor is at the cost of more computations and lower estimation accuracy.

5.5. Some Aspects regarding Classifier Training

For the problem of object categorization and detection, the appearance vectors, for example, the HOG or BoF vectors of subwindows, fed into a classifier are always of very high dimensions, which incur a high computational cost to both training and testing. This problem is more severe for the training process, in particular, when a large set of training samples is used, which remarkably increases the footprint of memory. To address this problem, incremental learning techniques can be exploited [Li and Fei-Fei 2010], and a set of specially modified classifiers, such as the Light SVM [Joachims 1998], Large-Scale Multiple Kernel Learning [Sonnenburg et al. 2006], and large-scale L1-regularized logistic regression [Koh et al. 2007], can also be used.

Another direction is to control the quantity of negatives in the training sets. This is particularly important or even indispensable to subwindow classifiers, as too many negatives can be harvested from each training image that are impractical to be considered simultaneously. To address this problem, it is common to construct a training set consisting of only the positive and "hard negative" instances, which can be implemented with bootstrapping [Andriluka et al. 2009; Shotton et al. 2005]: a model is first trained with an initial subset of negative examples, and then the negatives that are incorrectly classified by this initial model are collected to form a set of hard negatives for retraining. This process can be iterated multiple times to improve the results. More discussions on data mining of hard examples can be found in Felzenszwalb et al. [2010b].

Altogether, classification methods are very important for object class detection. The interested reader may refer to the review of Kotsiantis [2007] for more information in this related domain.

6. EVALUATION AND DATASETS

In the previous sections, we have introduced a large number of models, strategies, and methods. They are a small but representative portion of the existing body of literature. The existence of such a large number of competing techniques causes a natural problem to both developers and users of object class recognition and detection systems, that is, how to choose among them? Typically, empirical comparison is an effective and may be the most useful approach to address this problem. To make fair

and objective comparisons, standardization of testing data and of evaluation metrics is extremely crucial. The standardization can also benefit researchers because it would allow them to choose the best method among different ideas and to test new approaches against existing ones. Additionally, some standardized datasets, such as PASCAL VOC [Everingham et al. 2010], are extended constantly by updating with more challenging samples and by adding new tasks. As well, the annual open contests that are held undoubtedly help to stimulate research in this area.

6.1. Performance Evaluation Metrics

The metrics introduced in this section are mainly quantitative accuracy evaluation measures of categorization, detection, and segmentation methods. The evaluation of efficiency, that is, their computational and spatial complexities, is rather straightforward and is not discussed here.

Existing accuracy evaluation metrics need ground-truth data (often provided by human annotators) to judge whether or not an algorithmic output is correct, and thereby to compute the overall accuracy, which is detailed with respect to different vision tasks in the following.

6.1.1. Image Categorization Results Evaluation. The standard outputs of a categorization method on a testing image I are usually a predicted category label c and its confidence level f, which can be the classification score, for comparing with a threshold to determine whether c is accepted. The accuracy is commonly evaluated in a category-specific manner, that is, computing the averaged accuracy for each object class separately. In detail, for the evaluation of c_0-th category, denoting the ground-truth category label of I as c_g, the algorithmic output c is a true positive if $c = c_g = c_0$; a false positive if $c = c \neq c_g$; a true negative if $c = c_g \neq c_0$; and a false negative if $c \neq c_g = c_0$. For the whole testing set of this category, we count the total numbers of true positives, false positives, true negatives, and false negatives returned by the algorithm under evaluation and denote them respectively as, TPs, FPs, TNs, and FNs. Then the *precision* and *recall* measures can be computed by and $p = TPs/(TPs + FPs)$ and $r = TPs/(TPs + FNs)$. Obviously, $p, r \in [0, 1]$ and, when the threshold of the confidence level f increases, r increases while p decreases. Thus, there is an inherent trade-off between them [Zhu 2004]. By varying the confidence-level threshold, we obtain different pairs of (p, r) and the corresponding 2D *Precision-Recall Curve* (PRC), which illustrates this trade-off quantitatively. Furthermore, if precision is regarded as a function of recall, that is, $p(r)$, the *Average Precision* (AP) can be used to summarize the shape of PRC. Formally, partitioning the recall level $[0, 1]$ into n intervals $[0, 1/n, \ldots, (n-1)/n, 1]$, we can compute the AP by averaging the maxima of p over all n intervals, namely $AP = (\sum_{i=1}^{n} \max_{r \in [i-1/n, i/n]} p(r))/n$. The interested reader may refer to the tutorial [Zhu 2004] on recall, precision, and AP for more information.

The aforesaid method considers only the two-class evaluation, but categorization systems are also often evaluated in terms of multiclass recognition rates and confusion matrices. Denote the total number of categories in a testing set as C. The *recognition rate* for category c ($c = 1, \ldots, C$) is measured by the fraction of test images of this category that are correctly categorized by the algorithm under evaluation. Therefore, it is equal to the *recall* for category c, and hence varies with the threshold of the confidence-level f. We can further average the recognition rates over all categories and obtain a single performance measure, called the *mean recognition rate per class*. Usually, the highest mean recognition rate corresponds to the optimal value of the confidence-level threshold. Using the mean recognition rate is simple. However, it discards many evaluation results of the categorization method which may be useful for future improvement, for example, which categories are most frequently confused by

the current categorizer. To summarize such results, the *confusion matrix* is often used to provide more detailed accuracy evaluation.

The confusion matrix M_{ij} is a $C \times C$ matrix in which each element (i, j) stores the fraction of the test images from the i-th class that are categorized as the j-th class. Thus, each main diagonal element (i, i) stores the recognition rate for the i-th category. From the confusion matrix of an algorithm, for each category i, we can easily find out which category j i is most often confused with by inspecting the value of (i, j) for all j. When the confusion matrix has many significant off-diagonal values, then the appearance model adopted by the algorithm is not discriminative enough for these categories, and thus should be improved or replaced with better models.

6.1.2. Object Detection Results Evaluation. The standard outputs of a detection algorithm on a testing image I are usually the bounding box B for each object and its predicted category label c together with its confidence level f. The accuracy is also evaluated in a category-specific manner. For a testing image I, a predicted bounding box B is regarded as correct, that is, a true positive, if: (i) it corresponds to the same category with the ground-truth label c_g, and (ii) the overlap ratio a between the predicted bounding box B and the ground-truth one, B_g, is not smaller than a predefined threshold a_0, that is, $a = area(B|B_g)/area(B \cup B_g) \geq a_0$. A typical value of a_0 is 0.5. Note that multiple objects in the same image are considered as separate detection results. The detection results are also evaluated using precision, recall, and AP on a testing image set.

6.1.3. Object Segmentation Results Evaluation. Similar to detection, the standard segmentation outputs of an object in a testing image I consist of a segmented region S, its predicted category label c, and the confidence level f. The segmentation accuracy of each object of a specified category can be computed by $sa = area(S|S_g)\big/area(S \cup S_g)$, where S_g is the ground-truth region of the object. The final category-specific segmentation accuracy is the average over all testing objects of the same category.

The object segmentation results can also be evaluated in a multiclass manner. The *segmentation accuracy* of a single class in a testing set can be measured by the proportion of pixels in that class that are correctly categorized (i.e., labeled) by the segmenter under evaluation. Then, the *global* and *average accuracy* [Shotton et al. 2008b] can be further used to summarize the segmentation accuracies over the entire testing set. The former is the total proportion of pixels in that set that are correctly categorized, while the latter is the average of segmentation accuracies over all classes. Similar to multiclass image categorization, the confusion matrix can also be used to provide a more detailed evaluation result.

6.2. Datasets and the State-of-the-Art Results

Nowadays, there are many publicly available datasets on the Internet. Different datasets often serve different vision tasks (i.e., categorization, detection, or segmentation), collect images from different sources (e.g., Web images, photos captured by handheld cameras, or video sequences captured using on-board cameras), and provide different types of annotation (e.g., category names, bounding boxes, or pixel-level categorical labels). Among them, the PASCAL Visual Object Classes (VOC) dataset [Everingham et al. 2010] may be the most widely used one, which currently contains 20 different object classes and supports the evaluation of image categorization, object detection and segmentation, person layout classification, and action recognition algorithms. The images in this dataset were randomly selected and downloaded from the Flickr photo portal and the ground truths were produced by meticulous manual work. A very important activity that promotes this area of research is that the organizers hold an annual open contest to quantitatively compare newly proposed competing approaches. Other popular datasets mainly include Caltech-101 [Fei-Fei et al. 2004]

Table II. The Best Results of PASCAL VOC 2010 Competition

	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow
Categorization	93.3	79.0	71.6	77.8	54.3	85.9	80.4	79.4	64.5	66.2
Detection	58.4	55.3	19.2	21.0	35.1	55.5	49.1	47.7	20.0	31.5
Segmentation	58.3	27.4	39.0	37.8	47.4	63.2	62.4	42.6	9.1	36.8

	dining table	dog	horse	motorbike	person	potted plant	sheep	sofa	train	tv/monitor
Categorization	62.9	71.1	82.0	84.4	91.6	53.3	66.3	59.6	89.4	77.2
Detection	27.7	37.2	51.9	56.3	47.5	13.0	37.8	33.0	50.3	41.9
Segmentation	25.2	34.1	37.5	60.6	44.9	36.8	50.3	21.9	45.6	48.5

and Caltech-256 [Griffin et al. 2007] for image categorization, the Caltech dataset for pedestrian detection [Dollar et al. 2011], and MSRC-21 for semantic segmentation [Shotton et al. 2006]. Section 14.6 of Szeliski's textbook [Szeliski 2010] has an excellent summary of these datasets. The interested reader may refer to it to learn more about existing datasets.

Due to the page limitations, we only present in the following a very brief overview of the state-of-the-art results by summarizing the best (categorization, detection, and segmentation) results of the PASCAL VOC 2010 competition in Table II. The detailed results of all participants are published at http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2010/results/index.html, in which the categorization and detection results are evaluated by the AP (%) measure defined in Section 6.1.1, while the segmentation accuracy is measured by averaging the accuracy measure $sa(\%)$, defined in Section 6.1.3, over all testing objects of the same class. Note that the best results are in fact achieved by different algorithms, that is, *no single algorithm wins the contest with respect to all classes*.

Though the previous overview is brief and incomplete, we believe that it will be useful for researchers since it can help them to quickly look up for the best benchmark performance for each category when developing their own algorithms. Moreover, observing and analyzing these results (including all those published at the preceding URL) can reveal some interesting insights, which we believe to be instructive for object class detection methodology development. (i) Studying the technical details of the winners, for example, Song et al. [2011] for categorization, Zhang et al. [2011b] for detection, and Gonfaus et al. [2010] for segmentation, we discover that a combination of features, the incorporation of context, the use of kernel-based classifiers and the fusion of multiple models will help to achieve better overall accuracy, as the winning algorithms have adopted some of these strategies. This point is also implied by the fact that no single method can win the contests in all classes. (ii) The accuracy level achieved by the state-of-the-art detection and segmentation methods is far from satisfying the requirements of general-purpose practical applications. Hence, there is still much room for future improvement.

7. OFFSHOOTS: NEWLY EMERGING APPROACHES AND APPLICATIONS

In the new century, in particular during the last five years, we have witnessed an unprecedented boom and popularization of the Internet, fueled by the popular social Web sites such as Flickr and Facebook, each of which posts daily a huge number of manually tagged images uploaded by users from around the world. This phenomenon brings about remarkable changes to our thoughts, to our lives, as well as to the scientific world. As to the research in object class recognition and detection, this offers new solutions to existing hard problems, and motivates new ideas and new applications. In fact, these new issues are related to an emerging subfield called *Internet Vision*, which has received much attention recently.

Before discussing these new issues, let us look at the characteristics of images posted on social Web sites, called *social images*. We summarize their main features, relative to traditional images, as: (i) *very huge and constantly increasing in number*. In 2010, the pictures hosted on Flickr amounted to more than 5 billion, and those on Facebook were even more. Moreover, they keep increasing every second. (ii) They are *rich in contextual information*. These pictures are captured by different people, in different places, and at different events. They are accompanied with textual titles, tags, descriptions given by users, comments posted by viewers, and sometimes even correlated with videos, audios, or hyperlinks. The titles, tags, and descriptions usually describe the contents of the corresponding pictures, including the objects in them, the events they depict, etc. In addition, textual tags may contain information about the camera parameters, capturing time and locations. Comments from viewers are always aesthetic remarks on the photographs. (iii) They suffer *tag noise*. Due to the users' subjectivity and casualness, the tags and comments given are unavoidably noisy and imprecise.

7.1. New Solutions

Object class recognition and detection systems always need a large number of labeled instances to train the classifier. The labeling process usually incurs heavy burdens to the users. Though latent or hidden models can be used to partly overcome this, they will require more computational resources. Fortunately, the rapid proliferation of social images provides new ways to deal with this issue. In addition, they also provide new ways to implement categorization, detection, and segmentation.

7.1.1. Obtaining Large Sets of Labeled Images for Training from the Internet. As accompanied with textual tags, we can use keywords such as category names to collect object images from social datasets or Web sites. However, since the tags are noisy, some postprocessing steps are needed to refine the search results to obtain the final category images. To collect training images for object categorization, Wnuk and Soatto [2008] employ a nonparametric *strangeness* measure in the space of holistic image representations, and perform iterative visual feature elimination to remove outliers from the user queried images. A similar work is done by Fergus et al. [2010]. Xu et al. [2009] explore using Flickr's related tags to automatically annotate Web images. They derive a semantic correlation matrix from Flickr's related tags and present a new CRF model for Web image annotation, which integrates the derived keyword correlations, the textual and visual features of Web images into a unified graph model to improve the annotation accuracy. Fan et al. [2010] focus mainly on the tag denoising problem when harvesting weakly tagged social images for detection or recognition classifier training. They propose a cross-modal tag cleansing and junk image filtering algorithm to remove irrelevant images and to find the most relevant tags for each image. This algorithm integrates both the visual similarity contexts between images and the semantic similarity contexts between their tags. Quite differently, Li and Fei-Fei [2010] propose the Online Picture collecTion via Incremental MOdel Learning (OPTIMOL) system, which emulates the contiguous knowledge updating process of human observers and performs two tasks simultaneously: to automatically collect object datasets from the Web and to incrementally learn categorical models.

Another method is to obtain training images in a social Web manner, in particular, to involve online users. The LabelMe dataset [Russell et al. 2008] employs this strategy: a Web-based tool allowing easy image annotation is developed and deployed on the Internet, by which a large set of user-labeled images are collected. A different system, the I2T (image-to-text) conversion system, is proposed by Yao et al. [2010], by which: (i) images from the Internet are parsed in an interactive way to build an AND-OR Graph (AoG; see also Section 3.2.2) for visual knowledge representation, and hence to make

the parsing process more automatic using the learned AoG model; and (ii) images/videos in specific domains are parsed automatically to generate text reports that are useful for real-world applications.

7.1.2. Augmenting Visual Cues with Texts. The accompanied texts of social images provide additional cues for categorization, detection, and segmentation. Wang et al. [2009a] propose a text-based image feature to augment visual cues for hard object classification problems. They build an auxiliary dataset of social images, which are annotated with tags. Their proposed text feature, a normalized textual item name histogram, of an unannotated image is obtained from the tags of its k-nearest neighbors in this auxiliary collection. The aforementioned works of Hwang and Grauman [2010] and Lin et al. [2010] also make use of textual tags to augment visual features. Specifically, Hwang et al. extract "unspoken" cues of objects' prominence, mutual scale constraints, and relative spatial relations from the accompanied tag list of social images; Lin et al. exploit multidomain contextual information, for example, event in temporal domain, relative locations in spatial domain, which are embedded in the textual tags of each picture and the whole personal photo collections.

7.1.3. Categorization, Detection and Segmentation by Direct Matching. We have seen in the preceding discussions that social images facilitate the construction of large labeled image sets. As a result, a large number of different instances belonging to the same category, but with remarkable intra-class variations are collected, which largely reduce the occurrence frequencies of unseen objects and hence open up the possibility of implementing categorization, detection, and segmentation through direct matching with the training images rather than using them to learn the model parameters. Russell et al. [2007] propose this idea and present a system for recognizing object categories by scene alignment. In their subsequent work, Torralba et al. [2008] collected about 80 million color images from the Web, rescaled them to 32×32, and loosely labeled each of them with a noun in English from the WordNet lexicon (http://wordnet.princeton.edu/). The resulting database gives a comprehensive coverage of all object categories and scenes and can be used to perform object categorization by direct matching. Later, Liu et al. [2009a] proposed to perform categorization and scene parsing simultaneously using dense scene alignment based on a modified, coarse-to-fine SIFT flow algorithm. Similarly, Malisiewicz and Efros [2008] pose the recognition problem as data association, in which a novel object is explained in terms of a small set of visually similar exemplar objects. To detect and segment objects in a novel image, they first partition it into a set of regions using multiple bottom-up segmentation processes, and then associate these segmented regions with the exemplar regions by some learned distance functions.

7.2. New Applications

The existence of a huge number of social images also renews some related applications such as photo manipulation and image retrieval.

7.2.1. Social Image-Based Photo Manipulation. Traditional photo manipulation approaches either use smooth continuation or borrow pixels from other parts of the same image. With the advent of huge repositories of social images, it often makes more sense to find a different image to serve as the source of the missing pixels. Moreover, the accurate segmentation of these images into objects and the organization of the resulting segments into scene element databases allow object-level borrowing for photo manipulation, which will surely help to obtain more visually appealing results.

An early example is the AutoCollage system developed by Rother et al. [2006], which detects sky and faces in the input images to handle them more appropriately, that is, to preserve whole faces and to place sky regions at the top in the resulting collage.

One of the later representatives is the Photo-Clip-Art system proposed by Lalonde et al. [2007], which recognizes and segments objects of interest, such as pedestrians, in Internet photo collections and then allows users to paste them into their own photos. Similar works include Atkins [2008], Liu et al. [2009b], and Yang et al. [2008a], differing in the shapes of the resulting collages. Another representative is the powerful image synthesis system, the Sketch2Photo platform, developed by Chen et al. [2009]. It can compose a realistic picture from a simple freehand sketch annotated with text labels. The background and object images used are also obtained through searching and segmenting social images.

7.2.2. Attribute-Based Image Retrieval. The huge quantity and inaccurate user-given tags of social images often lead the images retrieved (from them) using keywords to be noisy, large quantity, low quality, and disorganized. In other words, retrieval of really useful information from a huge number of social images is very difficult. To address this problem, researchers explore a new direction of image retrieval, that is, attribute-based image retrieval, which needs automatic fine-grained object categorization to provide attribute labels for object images. The FaceTracer, a search engine for large sets of face images collected from the Internet, presented by Kumar et al. [2008], is just one such engine supporting attribute-based search, which can query images using natural language, such as "smiling men with blond hair and mustaches." Another attribute-based image search engine is the SkyFinder system of Tao et al. [2009], which uses a set of automatically extracted, semantic sky attributes such as layout, richness, and horizon.

8. CONCLUSIONS AND OPEN ISSUES

We have presented a comprehensive survey highlighting the current technical achievements, as well as the evaluation issues, emerging directions, and new research areas. However, as shown and analyzed in Section 6.2, the currently achieved accuracy level is still too low for general practical applications. Much work is left to be done in the future.

The ultimate goal of the community is to develop detection systems that can accurately and efficiently recognize and localize instances of all classes in all possible scenes, which can compete with, or even outperform, the human visual system. However, it is obvious that there remains a long way to go for such an ultimate goal, and we can only approach it gradually. To conjecture the roadmap toward this goal, we coarsely decompose it into two, more attainable subgoals, that is: (i) to constantly enhance the accuracy and efficiency of class-specific detection, and (ii) to cover more and more categories. Three keywords can be further used to summarize these subgoals, namely *accuracy*, *efficiency*, and *multiclass*. Since the hardware computational capabilities are increasing daily and history has repeatedly witnessed that any novel algorithm can be somewhat computationally optimized after its invention, we optimistically regard the enhancement of *efficiency*, compared with that of *accuracy*, less challenging and can be focused less in the foreseeable future.

To implement object class detection, local features, appearance models, localization strategies, and classification methods are very critical. Which of these factors are the most important for accurate categorization and detection? From the discussion in Section 2.2, the existence of large intra-class variations and possibly small inter-class differences challenge detection methods from achieving high averaged accuracies. Thus, we believe that the development of powerful appearance models, which can effectively deal with various kinds of intra-class variations and can capture the discriminative aspects between different categories, is the most important for enhancing detection accuracy. This is in agreement with the conclusion of the recent work of Parikh and Zitnick [2010], in which they explore the same problem through human studies and machine experiments, and find that, compared with learning algorithms and the amount of

training data, the choice of features is the most important for the accuracies of image-level object categorization. Actually, the word "feature" they used corresponds to the "mid-level representation of window-based models" discussed in this survey, as they regard color histogram, GIST, and BoF computed over an image (window) as features.

As discussed in Section 3, compared with unstructured models, structured ones are often more powerful for deformable classes, but they are commonly ineffective for amorphous classes. Additionally, in the family of structured models, window-based models are usually more appropriate for small objects or objects in canonical poses, while part-based ones are well suited for articulated objects in highly variable poses. Thus, to enhance accuracy as well as to cover more categories, future work should focus more on how to combine different models together to exploit their relative merits and to address the following issues: (i) combination of multiple features channels. As different classes usually have different relevant visual aspects and different feature channels are always complementary to each other, a powerful appearance model should make combinatorial use of multiple feature channels. Since 2007, this problem has received more and more attention. Early approaches [Varma and Ray 2007; Vedaldi et al. 2009] mainly use Multiple Kernel Learning (MKL), while boosting has been shown to be even more powerful recently [Gehler and Nowozin 2009]. However, these existing approaches are mostly based on the simple window-based representation. How to integrate diverse features efficiently and effectively for the more complex part-based models is still an open problem. (ii) Adaptively variable model configuration, as analyzed in Section 2.2, we see that different classes always have different types of intra-class variations. Thus the part configuration, as well as the feature combination, of the combined model should be adaptively varied along with the target classes. For example, it may include only one part for amorphous classes such as sky and clouds, and may adopt star-structured part topology for rigid classes such as bottle, and tree structure for complex, articulated classes such as human body. (iii) Capitalizing on contextual information, we see that context can be used to deal with background clutters and occlusions, particularly large degrees of occlusions which may render the appearance models ineffective. How to efficiently incorporate multisource, multilevel, and even multidomain contextual information is still left to be explored [Galleguillos and Belongie 2010].

In addition to powerful appearance models, we believe that the introduction of novel machine learning and feature analysis techniques will also create new advances in object class detection, similar to what AdaBoost, SVM, MKL, and Metric Learning, SIFT, and HOG have done in the past.

Last but not least, the constantly increasing volume of information in social media, in particular, social images, creates new opportunities as well as new challenges. As these images are commonly accompanied with rich correlated information on content, geo-location, time, hyperlinks, etc., they not only extend the connotation of context, but also bring the research related to object class detection to a new interdisciplinary scenario, in which social networking, textual information processing, geography, time-series analysis, machine learning, and even aesthetics and psychology are involved.

Finally, the research field of object class recognition and detection is still far from maturity. We may be at the beginning of a new era. More breakthroughs are expected in the future.

9. ELECTRONIC APPENDIX

The electronic appendix to this article is available in the ACM Digital Library.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their constructive comments. We also thank Gustavo Carneiro for permitting us to base Figure 8 in this article on one of his own figures. Thanks are also

given to Vittorio Ferrari, Pietro Perona, and Rob Fergus for permitting us to use the images of INRIA Horse and CALTECH Motorcycles datasets in Figure 3 and Figure 10, respectively. We also thank several Flickr users, including Erkhemchukhal Dorj, Viktoriia Vitkovskaya, James Wallace, Mike Seamons, and Kevin Dempsey, who have kindly allowed us to use (in Figure 4) their pictures shared on Flickr. X. Zhang gratefully acknowledges Heming Liu for her encouragement and Wei Li for his help when collecting the images needed for preparing other figures.

REFERENCES

Aggarwal, J. K. and Ryoo, M. S. 2011. Human activity analysis: A review. *ACM Comput. Surv. 43*, 1–43.

Alexe, B., Deselaers, T., and Ferrari, V. 2010. What is an object? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*.

An, S. J., Peursum, P., Liu, W. Q., and Venkatesh, S. 2009. Efficient algorithms for subwindow search in object detection and localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*.

Andriluka, M., Roth, S., and Schiele, B. 2009. Pictorial structures revisited: People detection and articulated pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*.

Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. 2009. From contours to regions: An empirical evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*.

Atkins, C. B. 2008. Blocked recursive image composition. In *Proceedings of the ACM International Conference on Multimedia (ACM/MM'08)*.

Aytar, Y. and Zisserman, A. 2011. Tabula rasa: Model transfer for object category detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'11)*.

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. 2008. Speeded-up robust features (surf). *Comput Vis. Image Understand. 110*, 346–359.

Bay, H., Tuytelaars, T., and Van Gool, L. 2006. SURF: Speeded up robust features. In *Proceedings of the European Conference on Computer Vision (ECCV'06)*.

Belongie, S., Malik, J., and Puzicha, J. 2001. Matching shapes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'01)*.

Belongie, S., Malik, J., and Puzicha, J. 2002. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell. 24*, 509–522.

Bentley, J. 1984. Programming pearls: Algorithm design techniques. *Comm. ACM 27*, 865–873.

Biederman, I., Mezzanotte, R., and Rabinowitz, J. 1982. Scene perception: Detecting and judging objects undergoing relational violations. *Cogn. Psychol. 14*, 143–177.

Blei, D. M., Ng, A. Y., and Jordan, M. I. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res. 3*, 993–1022.

Boiman, O., Shechtman, E., and Irani, M. 2008. In defense of nearest-neighbor based image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*.

Borenstein, E., Sharon, E., and Ullman, S. 2004. Combining top-down and bottom-up segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04)*.

Borenstein, E. and Ullman, S. 2008. Combined top-down/bottom-up segmentation. *IEEE Trans. Pattern Anal. Mach. Intell. 30*, 2109–2125.

Bosch, A., Zisserman, A., and Munoz, X. 2007a. Representing shape with a spatial pyramid kernel. In *Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR'07)*.

Bosch, A., Zisserman, A., and Muoz, X. 2007b. Image classification using random forests and ferns. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'07)*.

Bouchard, G. and Triggs, B. 2005. Hierarchical part-based visual object categorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05)*.

Boureau, Y. L., Bach, F., Lecun, Y., and Ponce, J. 2010. Learning mid-level features for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*.

Bray, M., Kohli, P., and Torr, P. 2006. PoseCut: Simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts. In *Proceedings of the European Conference on Computer Vision (ECCV'06)*.

Cai, H. P., Yan, F., and Mikolajczyk, K. 2010. Learning weights for codebook in image classification and retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*.

Cao, Y., Wang, C. H., Li, Z. W., Zhang, L. Q., and Zhang, L. 2010. Spatial-bag-of-features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*.

Carneiro, G. and Lowe, D. 2006. Sparse flexible models of local features. In *Proceedings of the European Conference on Computer Vision (ECCV'06)*.

CARREIRA, J., LI, F., AND SMINCHISESCU, C. 2011. Object recognition by sequential figure-ground ranking. *Int. J. Comput. Vis. 98,* 3, 243–262.

CARREIRA, J. AND SMINCHISESCU, C. 2010. Constrained parametric min-cuts for automatic object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10).*

CHEN, T., CHENG, M.-M., TAN, P., SHAMIR, A., AND HU, S.-M. 2009. Sketch2Photo: Internet image montage. In *Proceedings of the ACM SIGGRAPH Asia Papers.*

CHEN, Y., ZHU, L. L., LI, C. L., YUILLE, A., AND ZHANG, H. 2007. Rapid inference on a novel and/or graph for object detection, segmentation and parsing. In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS'07).*

CHIA, A. Y. S., RAHARDJA, S., RAJAN, D., AND LEUNG, M. K. H. 2009. Structural descriptors for category level object detection. *IEEE Trans. Multimedia 11*, 1407–1421.

CHRISTOUDIAS, C. M., URTASUN, R., AND DARRELL, T. 2008. Unsupervised feature selection via distributed coding for multi-view object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08).*

CRANDALL, D., FELZENSZWALB, P., AND HUTTENLOCHER, D. 2005. Spatial priors for part-based recognition using statistical models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05).*

CSURKA, G., DANCE, C., FAN, L., WILLAMOWSKI, J., AND BRAY, C. 2004. Visual categorization with bags of keypoints. In *Proceedings of the ECCV Workshop on Statistical Learning in Computer Vision (ECCVW'04).*

CSURKA, G., DANCE, C., PERRONNIN, F., AND WILLAMOWSKI, J. 2006. Generic visual categorization using weak geometry. In *Toward Category-Level Object Recognition*, J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, Eds., Springer, 207–224.

DALAL, N. 2006. Finding people in images and videos. Tech. rep., Institut National Polytechnique de Grenoble.

DALAL, N. AND TRIGGS, B. 2005. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05).*

DALAL, N., TRIGGS, B., AND SCHMID, C. 2006. Human detection using oriented histograms of flow and appearance. In *Proceedings of the European Conference on Computer Vision (ECCV'06).*

DATTA, R., JOSHI, D., LI, J., AND WANG, J. Z. 2008. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv. 40*, 1–60.

DESELAERS, T. AND FERRARI, V. 2010. Global and efficient self-similarity for object classification and detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10).*

DICKINSON, S. 2009. The evolution of object categorization and the challenge of image abstraction. In *Object Categorization: Computer and Human Vision Perspectives*, A. L. S. Dickinson, B. Schiele, and M. Tarr, Eds., Cambridge University Press, 1–37.

DIVVALA, S. K., HOIEM, D., HAYS, J. H., EFROS, A. A., AND HEBERT, M. 2009. An empirical study of context in object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09).*

DOLLAR, P., BELONGIE, S., AND PERONA, P. 2010. The fastest pedestrian detector in the west. In *Proceedings of the British Machine Vision Conference (BMVC'10).* BMVA Press.

DOLLAR, P., TU, Z., PERONA, P., AND BELONGIE, S. 2009. Integral channel features. In *Proceedings of the British Machine Vision Conference (BMVC'09).*

DOLLAR, P., WOJEK, C., SCHIELE, B., AND PERONA, P. 2011. Pedestrian detection: An evaluation of the state of the art. *IEEE Trans. Pattern Anal. Mach. Intell. 34,* 4, 743–761.

ENDRES, I. AND HOIEM, D. 2010. Category independent object proposals. In *Proceedings of the European Conference on Computer Vision (ECCV'10).*

ENZWEILER, M. AND GAVRILA, D. M. 2008. A mixed generative-discriminative framework for pedestrian classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08).*

EVERINGHAM, M., VAN GOOL, L., WILLIAMS, C., WINN, J., AND ZISSERMAN, A. 2010. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis. 88*, 2, 303–338.

FAN, J. P., SHEN, Y., ZHOU, N., AND GAO, Y. L. 2010. Harvesting large-scale weakly-tagged image databases from the web. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10).*

FEI-FEI, L., FERGUS, R., AND PERONA, P. 2004. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04).*

FEI-FEI, L. AND PERONA, P. 2005. A Bayesian hierarchical model for learning natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'04).*

Fei-Fei, L., Vanrullen, R., Koch, C., and Perona, P. 2002. Rapid natural scene categorization in the near absence of attention. *Proc. Nat. Acad. Sci. 2,* 9596–9601.

Fei-Fei, L., Fergus, R., and Torralba, A. 2005. Recognizing and learning object categories. In *International Conference on Computer Vision Short Course (ICCV'05).* MIT.

Fei-Fei, L., Fergus, R., and Torralba, A. 2007. Recognizing and learning object categories. In *Computer Vision and Pattern Recognition Short Course (CVPR'07).*

Fei-Fei, L., Fergus, R., and Torralba, A. 2009. Recognizing and learning object categories. In *International Conference on Computer Vision Short Course (ICCV'09).*

Felleman, D. J. and Van Essen, D. C. 1991. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex 1*, 1–47.

Felzenszwalb, P., Mcallester, D., and Ramanan, D. 2008. A discriminatively trained, multiscale, deformable part model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08).*

Felzenszwalb, P. F., Girshick, R. B., and Mcallester, D. 2010a. Cascade object detection with deformable part models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10).*

Felzenszwalb, P. F., Girshick, R. B., Mcallester, D., and Ramanan, D. 2010b. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell. 32*, 1627–1645.

Felzenszwalb, P. F. and Huttenlocher, D. P. 2005. Pictorial structures for object recognition. *Int. J. Comput. Vis. 61*, 55–79.

Felzenszwalb, P. F. and Veksler, O. 2010. Tiered scene labeling with dynamic programming. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10).*

Ferencz, A., Learned-Miller, E., and Malik, J. 2008. Learning to locate informative features for visual identification. *Int. J. Comput. Vis. 77*, 3–24.

Fergus, R., Li, F.-F., Perona, P., and Zisserman, A. 2010. Learning object categories from internet image searches. *Proc. IEEE. 98*, 1453–1466.

Fergus, R., Perona, P., and Zisserman, A. 2003. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03).*

Fergus, R., Perona, P., and Zisserman, A. 2005. A sparse object category model for efficient learning and exhaustive recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05)*.

Fergus, R., Perona, P., and Zisserman, A. 2007. Weakly supervised scale-invariant learning of models for visual recognition. *Int. J. Comput. Vis. 71*, 273–303.

Ferrari, V., Fevrier, L., Jurie, F., and Schmid, C. 2008. Groups of adjacent contour segments for object detection. *IEEE Trans. Pattern Anal. Mach. Intell. 30*, 36–51.

Fischler, M. A. and Elschlager, R. A. 1973. The representation and matching of pictorial structures. *IEEE Trans. Comput. C-22*, 67–92.

Fleuret, F. and Geman, D. 2001. Coarse-to-fine face detection. *Int. J. Comput. Vis. 41*, 85–107.

Fulkerson, B., Vedaldi, A., and Soatto, S. 2009. Class segmentation and object localization with superpixel neighborhoods. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'09).*

Gall, J. and Lempitsky, V. 2009. Class-specific hough forests for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09).*

Gallagher, A., Neustaedter, C., Cao, L., Luo, J., and Chen, T. 2008. Image annotation using personal calendars as context. In *Proceedings of the ACM International Conference on Multimedia (ACM/MM'08).*

Gallagher, A. C. and Chen, T. 2008. Estimating age, gender, and identity using first name priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08).*

Galleguillos, C. and Belongie, S. 2010. Context based object categorization: A critical survey. *Comput Vis. Image Understand. 114*, 712–722.

Galleguillos, C., Mcfee, B., Belongie, S., and Lanckriet, G. 2010. Multi-class object localization by combining local contextual interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10).*

Gehler, P. and Nowozin, S. 2009. On feature combination for multiclass object classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'09).*

Girshick, R. B., Felzenszwalb, P. F., and Mcallester, D. 2011. Object detection with grammar models. In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS'11).*

Gonfaus, J. M., Boix, X., Van de Weijer, J., Bagdanov, A. D., Serrat, J., and Gonzalez, J. 2010. Harmony potentials for joint classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10).*

GOULD, S., FULTON, R., AND KOLLER, D. 2009a. Decomposing a scene into geometric and semantically consistent regions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*.

GOULD, S., GAO, T. S., AND KOLLER, D. 2009b. Region-based segmentation and object detection. In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS'09)*.

GRABNER, H., ROTH, P. M., AND BISCHOF, H. 2007. Eigenboosting: Combining discriminative and generative information. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07)*.

GRAUMAN, K. AND LEIBE, B. 2011. Visual object recognition. *Synthesis Lectures Artif. Intell. Mach. Learn. 5*, 1–181.

GRIFFIN, G., HOLUB, A., AND PERONA, P. 2007. Caltech-256 object category dataset. Tech. rep., California Institute of Technology, 1-20. http://authors.library.caltech.edu/7694/1/CNS-TR-2007-001.pdf.

GU, C. H., LIM, J. J., ARBELAEZ, P., AND MALIK, J. 2009. Recognition using regions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*.

GUILLAUMIN, M., VERBEEK, J., AND SCHMID, C. 2010. Multimodal semi-supervised learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*.

HAYS, J. AND EFROS, A. A. 2008. IM2GPS: Estimating geographic information from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*.

HE, X. M., ZEMEL, R., AND RAY, D. 2006. Learning and incorporating top-down cues in image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV'06)*.

HE, X. M., ZEMEL, R. S., AND CARREIRA-PERPINAN, M. A. 2004. Multiscale conditional random fields for image labeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'04)*.

HEITZ, G., ELIDAN, G., PACKER, B., AND KOLLER, D. 2009. Shape-based object localization for descriptive classification. *Int. J. Comput. Vis. 84*, 40–62.

HOCHSTEIN, S. AND AHISSAR, M. 2002. View from the top: Hierarchies and reverse hierarchies in the visual system. *Neuron 36*, 791–804.

HOFMANN, T. 2001. Unsupervised learning by probabilistic latent semantic analysis. *Mach. Learn. 42*, 177–196.

HOIEM, D., EFROS, A., AND HEBERT, M. 2008. Putting objects in perspective. *Int. J. Comput. Vis. 80*, 3–15.

HOIEM, D., ROTHER, C., AND WINN, J. 2007a. 3D layoutcrf for multi-view object class recognition and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07)*.

HOIEM, D., STEIN, A., EFROS, A., AND HEBERT, M. 2007b. Recovering occlusion boundaries from a single image. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'07)*.

HUANG, Y. Z., HUANG, K. Q., WANG, L. S., TAO, D. C., TAN, T. N., AND LI, X. L. 2008. Enhanced biologically inspired model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*.

HWANG, S. J. AND GRAUMAN, K. 2010. Reading between the lines: Object localization using implicit cues from image tags. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*.

JAIN, A., GUPTA, A., AND DAVIS, L. 2010. Learning what and how of contextual models for scene labeling. In *Proceedings of the European Conference on Computer Vision (ECCV'10)*.

JHUANG, H., SERRE, T., WOLF, L., AND POGGIO, T. 2007. A biologically inspired system for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'07)*.

JI, R. R., YAO, H. X., SUN, X. S., ZHONG, B. N., AND GAO, W. 2010. Towards semantic embedding in visual vocabulary. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*.

JIANG, Y.-G., YANG, J., NGO, C.-W., AND HAUPTMANN, A. G. 2010. Representations of keypoint-based semantic concept detection: A comprehensive study. *IEEE Trans. Multimedia 12*, 42–53.

JOACHIMS, T. 1997. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In *Proceedings of the International Conference on Machine Learning (ICML'97)*.

JOACHIMS, T. 1998. Making large-scale support vector machine learning practical. In *Advances in Kernel Methods: Support Vector Machines*, B. Scholkopf, J. C. Burges, and A. J. Smola, Eds. MIT Press, Cambridge, MA, 169–184.

JONES, J. P. AND PALMER, L. A. 1987. An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *J. Neurophys. 58*, 1233–1258.

JURIE, F. AND TRIGGS, B. 2005. Creating efficient codebooks for visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'05)*.

KARLINSKY, L., DINERSTEIN, M., HARARI, D., AND ULLMAN, S. 2010. The chains model for detecting parts by their context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*.

KE, Y. AND SUKTHANKAR, R. 2004. PCA-sift: A more distinctive representation for local image descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'04)*.

KNOPP, J., PRASAD, M., AND GOOL, L. V. 2011. Scene cut: Class-specific object detection and segmentation in 3d scenes. In *Proceedings of the International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT'11)*.

KOH, K., KIM, S.-J., AND BOYD, S. 2007. An interior-point method for large-scale l1-regularized logistic regression. *J. Mach. Learn. Res. 8*, 1519–1555.

KOHLI, P., LADICKY, L., AND TORR, P. 2008. Robust higher order potentials for enforcing label consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*.

KOTSIANTIS, S. B. 2007. Supervised machine learning: A review of classification techniques. *Informatica 31*, 249–268.

KRÜGER, V., KRAGIC, D., UDE, A., AND GEIB, C. 2007. The meaning of action: A review on action recognition and mapping. *Advan. Robot. 21*, 1473–1501.

KUETTEL, D. AND FERRARI, V. 2012. Figure-ground segmentation by transferring window masks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12)*.

KUMAR, M. P., TON, P. H. S., AND ZISSERMAN, A. 2005. OBJCUT. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05)*.

KUMAR, M. P., TORR, P. H. S., AND ZISSERMAN, A. 2010. OBJCUT: Efficient segmentation using top-down and bottom-up cues. *IEEE Trans. Pattern Anal. Mach. Intell. 32*, 530–545.

KUMAR, N., BELHUMEUR, P., AND NAYAR, S. 2008. FaceTracer: A search engine for large collections of images with faces. In *Proceedings of the European Conference on Computer Vision (ECCV'08)*.

LADICKY, L., STURGESS, P., ALAHARI, K., RUSSELL, C., AND TORR, P. 2010. What, where and how many? Combining object detectors and crfs. In *Proceedings of the European Conference on Computer Vision (ECCV'10)*.

LALONDE, J.-F., HOIEM, D., EFROS, A. A., ROTHER, C., WINN, J., AND CRIMINISI, A. 2007. Photo clip art. In *Proceedings of the International Conference and Exhibition on Computer Graphics and Interactive Techniques (ACM/SIGGRAPH'07)*.

LALONDE, J.-F., NARASIMHAN, S. G., AND EFROS, A. A. 2010. What do the sun and the sky tell us about the camera? *Int. J. Comput. Vis. 88*, 24–51.

LAMPERT, C. H., BLASCHKO, M. B., AND HOFMANN, T. 2008. Beyond sliding windows: Object localization by efficient sub-window search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*.

LAPTEV, I. 2006. Improvements of object detection using boosted histograms. In *Proceedings of the British Machine Vision Conference (BMVC'06)*.

LAPTEV, I. 2009. Improving object detection with boosted histograms. *Image Vis. Comput. 27*, 535–544.

LARLUS, D. AND JURIE, F. 2008. Combining appearance models and markov random fields for category level object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*.

LARLUS, D., VERBEEK, J., AND JURIE, F. 2010. Category level object segmentation by combining bag-of-words models with dirichlet processes and random fields. *Int. J. Comput. Vision 88*, 238–253.

LASSERRE, J. A., BISHOP, C. M., AND MINKA, T. P. 2006. Principled hybrids of generative and discriminative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06)*.

LAZEBNIK, S., SCHMID, C., AND PONCE, J. 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06)*.

LEE, H., BATTLE, A., RAINA, R., AND NG, A. Y. 2006. Efficient sparse coding algorithms. *Adv. Neural Inf. Process. Syst. 19,* 2007.

LEE, Y. J. AND GRAUMAN, K. 2010. Object-graphs for context-aware category discovery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*.

LEIBE, B., LEONARDIS, A., AND SCHIELE, B. 2004. Combined object categorization and segmentation with an implicit shape model. In *Proceedings of the ECCV Workshop on Statistical Learning in Computer Vision (ECCVW'04)*.

LEIBE, B., LEONARDIS, A., AND SCHIELE, B. 2006. An implicit shape model for combined object categorization and segmentation. In *Toward Category-Level Object Recognition*, J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, Eds., Springer, 508–524.

LEIBE, B., LEONARDIS, A., AND SCHIELE, B. 2008. Robust object detection with interleaved categorization and segmentation. *Int. J. Comput. Vis. 77*, 259–289.

LEIBE, B., SEEMANN, E., AND SCHIELE, B. 2005. Pedestrian detection in crowded scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05).*

LEMPITSKY, V., KOHLI, P., ROTHER, C., AND SHARP, T. 2009. Image segmentation with a bounding box prior. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'09).*

LEVIN, A. AND WEISS, Y. 2006. Learning to combine bottom-up and top-down segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV'06).*

LI, L.-J. AND FEI-FEI, L. 2007. What, where and who? Classifying events by scene and object recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'07).*

LI, L.-J. AND FEI-FEI, L. 2010. OPTIMOL: Automatic online picture collection via incremental model learning. *Int. J. Comput. Vis. 88*, 147–168.

LIANG, P. AND JORDAN, M. I. 2008. An asymptotic analysis of generative, discriminative, and pseudolikelihood estimators. In *Proceedings of the International Conference on Machine Learning (ICML'08).*

LIEBELT, J. AND SCHMID, C. 2010. Multi-view object class detection with a 3d geometric model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10).*

LIEBELT, J., SCHMID, C., AND SCHERTLER, K. 2008. Viewpoint-independent object class detection using 3d feature maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08).*

LIN, D., KAPOOR, A., HUA, G., AND BAKER, S. 2010. Joint people, event, and location recognition in personal photo collections using cross-domain context. In *Proceedings of the European Conference on Computer Vision (ECCV'10).*

LIN, Z. 2009. Modeling shape, appearance and motion for human movement analysis. Tech. rep., Department of Electrical and Computer Engineering, University of Maryland, College Park, Md. http://hdl.handle.net/1903/9279.

LIN, Z. AND DAVIS, L. S. 2010. Shape-based human detection and segmentation via hierarchical part-template matching. *IEEE Trans. Pattern Anal. Mach. Intell. 32*, 604–618.

LIN, Z., DAVIS, L. S., DOERMANN, D., AND DEMENTHON, D. 2007. Hierarchical part-template matching for human detection and segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'07).*

LIU, C., YUEN, J., AND TORRALBA, A. 2009a. Nonparametric scene parsing: Label transfer via dense scene alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09).*

LIU, C., YUEN, J., TORRALBA, A., SIVIC, J., AND FREEMAN, W. 2008. SIFT flow: Dense correspondence across different scenes. In *Proceedings of the European Conference on Computer Vision (ECCV'08).*

LIU, T., WANG, J. D., SUN, J., ZHENG, N. N., TANG, X. O., AND SHUM, H. Y. 2009b. Picture collage. *IEEE Trans. Multimedia 11*, 1225–1239.

LOWE, D. G. 2004. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision 60*, 91–110.

LU, Z. W. AND IP, H. H. S. 2009. Image categorization with spatial mismatch kernels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09).*

LUO, J., BOUTELL, M., AND BROWN, C. 2006. Pictures are not taken in a vacuum. *IEEE Signal Process. Mag. 23*, 101–114.

MAIRE, M., YU, S. X., AND PERONA, P. 2011. Object detection and segmentation from joint embedding of parts and pixels. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'11).*

MAJI, S., BERG, A. C., AND MALIK, J. 2008. Classification using intersection kernel support vector machines is efficient. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08).*

MALISIEWICZ, T. AND EFROS, A. A. 2008. Recognition by association via learning per-exemplar distances. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08).*

MARSZAŁEK, M. AND SCHMID, C. 2007. Accurate object localization with shape masks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07).*

MIKOLAJCZYK, K. AND SCHMID, C. 2005. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell. 27*, 1615–1630.

MOOSMANN, F., NOWAK, E., AND JURIE, F. 2008. Randomized clustering forests for image classification. *IEEE Trans. Pattern Anal. Mach. Intell. 30,* 1632–1646.

MU, Y., YAN, S., LIU, Y., HUANG, T., AND ZHOU, B. 2008. Discriminative local binary patterns for human detection in personal album. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08).*

MUTCH, J. AND LOWE, D. 2008. Object class recognition and localization using sparse features with limited receptive fields. *Int. J. Comput. Vis. 80,* 45–57.

MUTCH, J. AND LOWE, D. G. 2006. Multiclass object recognition with sparse, localized features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06).*

NAKAYAMA, H., HARADA, T., AND KUNIYOSHI, Y. 2010. Global gaussian approach for scene categorization using information geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*.

NARASIMHAN, S. AND NAYAR, S. 2002. Vision and the atmosphere. *Int. J. Comput. Vis. 48*, 233–254.

NG, A. AND JORDAN, M. 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS'02)*.

NI, B. B., YAN, S. C., AND KASSIM, A. 2009. Contextualizing histogram. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*.

NISTER, D. AND STEWENIUS, H. 2006. Scalable recognition with a vocabulary tree. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06)*.

NOWAK, E., JURIE, F., AND TRIGGS, B. 2006. Sampling strategies for bag-of-features image classification. In *Proceedings of the European Conference on Computer Vision (ECCV'06).*

OJALA, T., PIETIKAINEN, M., AND MAENPAA, T. 2002. Multi-resolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell. 24*, 971–987.

OLIVA, A. AND TORRALBA, A. 2001. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vis. 42*, 145–175.

OLIVA, A. AND TORRALBA, A. 2006. Building the gist of a scene: The role of global image features in recognition. *Progress Brain Res. 155*, 23–36.

OLIVA, A. AND TORRALBA, A. 2007. The role of context in object recognition. *Trends Cogn. Sci. 11*, 520–527.

OPELT, A., PINZ, A., AND ZISSERMAN, A. 2006. A boundary-fragment-model for object detection. In *Proceedings of the European Conference on Computer Vision (ECCV'06).*

PALMESE, M. AND TRUCCO, A. 2008. From 3-d sonar images to augmented reality models for objects buried on the seafloor. *IEEE Trans. Instrument. Measure. 57*, 820–828.

PARIKH, D. AND ZITNICK, C. L. 2010. The role of features, algorithms and data in visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10).*

PARK, D., RAMANAN, D., AND FOWLKES, C. 2010. Multiresolution models for object detection. In *Proceedings of the European Conference on Computer Vision (ECCV'10).*

PEDERSOLI, M., VEDALDI, A., AND GONZALEZ, J. 2011. A coarse-to-fine approach for fast deformable object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11).*

PERRONNIN, F. 2008. Universal and adapted vocabularies for generic visual categorization. *IEEE Trans. Pattern Anal. Mach. Intell. 30*, 1243–1256.

PERROTTON, X., STURZEL, M., AND ROUX, M. 2010. Implicit hierarchical boosting for multi-view object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10).*

PINZ, A. 2005. Object categorization. *Foundat. Trends Comput. Graph. Vis. 1,* 4, 255–353.

PONCE, J., BERG, T. L., EVERINGHAM, M., FORSYTH, D. A., HEBERT, M., LAZEBNIK, S., MARSZAŁEK, M., SCHMID, C., RUSSELL, B. C., TORRALBA, A., WILLIAMS, C. K. I., ZHANG, J., AND ZISSERMAN, A. 2006a. Dataset issues in object recognition. In *Toward Category-Level Object Recognition*, J. Ponce, M. Hebert, C. Schmid, and A. Zisserman Eds., Springer, 29–48.

PONCE, J., HEBERT, M., SCHMID, C., AND ZISSERMAN, A. 2006b. *Toward Category-Level Object Recognition*. Lecture Notes in Computer Science, vol. 4170, Springer.

PORIKLI, F. 2005. Integral histogram: A fast way to extract histograms in cartesian spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05).*

RABINOVICH, A., VEDALDI, A., GALLEGUILLOS, C., WIEWIORA, E., AND BELONGIE, S. 2007. Objects in context. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'07).*

RAVISHANKAR, S., JAIN, A., AND MITTAL, A. 2008. Multi-stage contour based detection of deformable objects. In *Proceedings of the European Conference on Computer Vision (ECCV'08).*

RAZAVI, N., GALL, J., AND VAN GOOL, L. 2010. Backprojection revisited: Scalable multi-view object detection and similarity metrics for detections. In *Proceedings of the European Conference on Computer Vision (ECCV'10).*

REN, X. AND MALIK, J. 2003. Learning a classification model for segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'03).*

RIESENHUBER, M. AND POGGIO, T. 1999. Hierarchical models of object recognition in cortex. *Nature Neurosci. 2*, 1019–1025.

ROTHER, C., BORDEAUX, L., HAMADI, Y., AND BLAKE, A. 2006. AutoCollage. *ACM Trans. Graph. 25,* 3, 847–852.

ROTHER, C., KOLMOGOROV, V., AND BLAKE, A. 2004. "GrabCut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph. 23,* 3, 309–314.

RUBINSTEIN, D. AND HASTIE, T. 1997. Discriminative vs informative learning. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDDM'97)*.

RUBNER, Y., TOMASI, C., AND GUIBAS, L. J. 2000. The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vis. 40*, 99–121.

RUI, X., LI, M., LI, Z., MA, W.-Y., AND YU, N. 2007. Bipartite graph reinforcement model for web image annotation. In *Proceedings of the ACM International Conference on Multimedia (ACM/MM'07).*

RUSSELL, B., TORRALBA, A., LIU, C., FERGUS, R., AND FREEMAN, W. 2007. Object recognition by scene alignment. In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS'07).*

RUSSELL, B., TORRALBA, A., MURPHY, K., AND FREEMAN, W. 2008. LabelMe: A database and web-based tool for image annotation. *Int. J. Comput. Vis. 77*, 157–173.

SABZMEYDANI, P. AND MORI, G. 2007. Detecting pedestrians by learning shapelet features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07).*

SAFFARI, A., GODEC, M., POCK, T., LEISTNER, C., AND BISCHOF, H. 2010. Online multi-class lpboost. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10).*

SALAKHUTDINOV, R., TORRALBA, A., AND TENENBAUM, J. 2011. Learning to share visual appearance for multiclass object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11).*

SALZMANN, M. AND URTASUN, R. 2010. Combining discriminative and generative methods for 3d deformable surface and articulated pose reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10).*

SAVARESE, S. AND LI, F.-F. 2007. 3D generic object categorization, localization and pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'07).*

SAVARESE, S., WINN, J., AND CRIMINISI, A. 2006. Discriminative object class models of appearance and shape by correlatons. In *Preceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06)*.

SCHINDLER, K., VAN GOOL, L., AND DE GELDER, B. 2008. Recognizing emotions expressed by body pose: A biologically inspired neural model. *Neural Netw. 21*, 1238–1246.

SCHROFF, F. 2009. Semantic image segmentation and web-supervised visual learning. Tech. rep., Robotics Research Group, Department of Engineering Science. University of Oxford, Oxford, UK. http://www.robots.ox.ac.uk/~vgg/publications/papers/schroff09.pdf.

SEEMANN, E., LEIBE, B., AND SCHIELE, B. 2006. Multi-aspect detection of articulated objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06).*

SERRE, T., OLIVA, A., AND POGGIO, T. 2007a. A feed-forward architecture accounts for rapid categorization. *Proc. National Acad. Sci. 104*, 6424–6429.

SERRE, T., WOLF, L., BILESCHI, S., RIESENHUBER, M., AND POGGIO, T. 2007b. Robust object recognition with cortex-like mechanisms. *IEEE Trans. Pattern Anal. Mach. Intell. 29*, 411–426.

SERRE, T., WOLF, L., AND POGGIO, T. 2005. Object recognition with features inspired by visual cortex. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05).*

SHECHTMAN, E. AND IRANI, M. 2007. Matching local self-similarities across images and videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07).*

SHIN, Y., KIM, Y., AND KIM, E. Y. 2010. Automatic textile image annotation by predicting emotional concepts from visual features. *Image Vis. Comput. 28*, 526–537.

SHOTTON, J., BLAKE, A., AND CIPOLLA, R. 2005. Contour-based learning for object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'05).*

SHOTTON, J., BLAKE, A., AND CIPOLLA, R. 2008a. Multiscale categorical object recognition using contour fragments. *IEEE Trans. Pattern Anal. Mach. Intell. 30*, 1270–1281.

SHOTTON, J., JOHNSON, M., AND CIPOLLA, R. 2008b. Semantic texton forests for image categorization and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08).*

SHOTTON, J., WINN, J., ROTHER, C., AND CRIMINISI, A. 2006. TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV'06).*

SHOTTON, J., WINN, J., ROTHER, C., AND CRIMINISI, A. 2009. TextonBoost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *Int. J. Comput. Vis. 81*, 2–23.

Simon, I. and Seitz, S. 2008. Scene segmentation using the wisdom of crowds. In *Proceedings of the European Conference on Computer Vision (ECCV'08)*.

Sivic, J., Russell, B. C., Efros, A. A., Zisserman, A., and Freeman, W. T. 2005. Discovering objects and their location in images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'05)*.

Sivic, J. and Zisserman, A. 2003. Video google: Text retrieval approach to object matching in videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'03)*.

Snavely, N., Simon, I., Goesele, M., Szeliski, R., and Seitz, S. M. 2010. Scene reconstruction and visualization from community photo collections. *Proc. IEEE. 98*, 1370–1390.

Song, D. J. and Tao, D. C. 2010. Biologically inspired feature manifold for scene classification. *IEEE Trans. Image Process. 19*, 174–184.

Song, Z., Chen, Q., Huang, Z., Hua, Y., and Yan, S. 2011. Contextualizing object detection and classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*.

Sonnenburg, S., Rutsch, G., Schafer, C., and Scholkopf, B. 2006. Large scale multiple kernel learning. *J. Mach. Learn. Res. 7*, 1531–1565.

Strat, T. 1993. Employing contextual information in computer vision. In *Proceedings of the ARPA Image Understanding Workshop*. 217–229.

Sutton, C. and McCallum, A. 2006. An introduction to conditional random fields for relational learning. In *Introduction to Statistical Relational Learning*, L. Getoor and B. Taskar, Eds., MIT Press. http://people.cs.umass.edu/~mccallum/papers/crf-tutorial.pdf.

Szeliski, R. 2010. *Computer Vision: Algorithms and Applications*. Springer.

Tao, L., Yuan, L., and Sun, J. 2009. SkyFinder: Attribute-based sky image search. In *ACM SIGGRAPH Papers*.

Thomas, A., Ferrari, V., Leibe, B., Tuytelaars, T., Schiele, B., and Van Gool, L. 2006. Towards multi-view object class detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06)*.

Torralba, A. 2003. Contextual priming for object detection. *Int. J. Comput. Vis. 53*, 169–191.

Torralba, A., Fergus, R., and Freeman, W. T. 2008. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell. 30*, 1958–1970.

Torralba, A., Murphy, K., and Freeman, W. 2006. Shared features for multiclass object detection. In *Toward Category-Level Object Recognition*, J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, Eds., Springer, 345–361.

Torralba, A., Murphy, K. P., and Freeman, W. T. 2004. Sharing features: Efficient boosting procedures for multiclass object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'04)*.

Torralba, A., Murphy, K. P., Freeman, W. T., and Rubin, M. A. 2003. Context-based vision system for place and object recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'03)*.

Tu, Z. W. 2007. Learning generative models via discriminative approaches. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07)*.

Ulusoy, I. and Bishop, C. 2006. Comparison of generative and discriminative techniques for object detection and classification. In *Toward Category-Level Object Recognition*, J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, Eds., Springer, 173–195.

Ulusoy, I. and Bishop, C. M. 2005. Generative versus discriminative methods for object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05)*.

Van De Sande, K., Gevers, T., and Snoek, C. 2008. Evaluation of color descriptors for object and scene recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*.

Van De Sande, K., Gevers, T., and Snoek, C. 2010. Evaluating color descriptors for object and scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell. 32*, 1582–1596.

Van De Sande, K., Uijlings, J., Gevers, T., and Smeulders, A. 2011. Segmentation as selective search for object recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'11)*.

Van Gemert, J. C., Veenman, C. J., Smeulders, A. W. M., and Geusebroek, J. M. 2010. Visual word ambiguity. *IEEE Trans. Pattern Anal. Mach. Intell. 32*, 1271–1283.

Vapnik, V. N. 1998. *Statistical Learning Theory*. A Wiley-Interscience Publication, New York.

Varma, M. and Ray, D. 2007. Learning the discriminative power-invariance trade-off. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'07)*.

Vedaldi, A., Gulshan, V., Varma, M., and Zisserman, A. 2009. Multiple kernels for object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'09)*.

VERBEEK, J. AND TRIGGS, B. 2007a. Region classification with markov field aspect models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07)*.

VERBEEK, J. AND TRIGGS, B. 2007b. Scene segmentation with conditional random fields learned from partially labeled images. In *Proceedings of the Conference on Advances in Neural Information Processing Systems. (NIPS'07)*.

VIJAYANARASIMHAN, S. AND GRAUMAN, K. 2011. Efficient region search for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*.

VIOLA, P. AND JONES, M. 2001. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01)*.

WALK, S., MAJER, N., SCHINDLER, K., AND SCHIELE, B. 2010. New features and insights for pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*.

WANG, G., GALLAGHER, A., LUO, J., AND FORSYTH, D. 2010a. Seeing people in social context: Recognizing people and social relationships. In *Proceedings of the European Conference on Computer Vision (ECCV'10)*.

WANG, G., HOIEM, D., AND FORSYTH, D. 2009a. Building text features for object image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*.

WANG, J., YANG, J., YU, K., LV, F., HUANG, T., AND GONG, Y. 2010b. Locality-constrained linear coding for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*.

WANG, X. AND GRIMSON, E. 2007. Spatial latent dirichlet allocation. In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS'07)*.

WANG, X., HAN, T. X., AND YAN, S. 2009b. An hog-lbp human detector with partial occlusion handling. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'09)*.

WANG, Y. AND MORI, G. 2009. Max-margin hidden conditional random fields for human action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*.

WANG, Y. AND MORI, G. 2010. Hidden part models for human action recognition: Probabilistic vs. max-margin. *IEEE Trans. Pattern Anal. Mach. Intell. 33,* 7, 1310–1323.

WANG, Z., HU, Y., AND CHIA, L.-T. 2010c. Image-to-class distance metric learning for image classification. In *Proceedings of the European Conference on Computer Vision (ECCV'10)*.

WATANABE, T., ITO, S., AND YOKOI, K. 2009. Co-occurrence histograms of oriented gradients for pedestrian detection. In *Proceedings of the Pacific-Rim Symposium on Image and Video Technology (PSIVT'09)*.

WEI, Y. C. AND TAO, L.T. 2010. Efficient histogram-based sliding window. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*.

WINN, J., CRIMINISI, A., AND MINKA, T. 2005. Object categorization by learned universal visual dictionary. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'05)*.

WNUK, K. AND SOATTO, S. 2008. Filtering internet image search results towards keyword based category recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*.

WOJEK, C. AND SCHIELE, B. 2008. A performance evaluation of single and multi-feature people detection. In *Proceedings of the German Association for Pattern Recognition (DAGM'08)*.

WOJEK, C., WALK, S., AND SCHIELE, B. 2009. Multi-cue onboard pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*.

WRIGHT, J., YI, M., MAIRAL, J., SAPIRO, G., HUANG, T. S., AND SHUICHENG, Y. 2010. Sparse representation for computer vision and pattern recognition. *Proc. IEEE 98*, 1031–1044.

WU, B. AND NEVATIA, R. 2005. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'05)*.

WU, B. AND NEVATIA, R. 2007a. Cluster boosted tree classifier for multi-view, multi-pose object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'07)*.

WU, B. AND NEVATIA, R. 2007b. Improving part based object detection by unsupervised, online boosting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07)*.

WU, B. AND NEVATIA, R. 2007c. Simultaneous object detection and segmentation by boosting local shape feature based classifier. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Reconition (CVPR'07)*.

WU, Z., KE, Q. F., ISARD, M., AND SUN, J. 2009. Bundling features for large scale partial-duplicate web image search. In *Proceedings of the IEEE Conferenc on Computer Vision and Pattern Recognition (CVPR'09)*.

XIANG, Y., ZHOU, X. D., LIU, Z. T., CHUA, T. S., AND NGO, C.-W. 2010. Semantic context modeling with maximal margin conditional random fields for automatic image annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*.

Xu, H., Zhou, X., Wang, M., Xiang, Y., and Shi, B. 2009. Exploring flickr's related tags for semantic annotation of web images. In *Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR'09)*.

Xu, Z., Chen, H., Zhu, S.-C., and Luo, J. 2008. A hierarchical compositional model for face representation and sketching. *IEEE Trans. Pattern Anal. Mach. Intell. 30*, 955–969.

Xue, J.-H. 2008. Aspects of generative and discriminative classifiers. Tech. rep., Information and Mathematical Sciences, Department of Statistics, University of Glasgow.

Xue, J.-H. and Titterington, D. 2008. Comment on "on discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes". *Neural Process. Lett. 28*, 169–187.

Xue, J.-H. and Titterington, D. M. 2010. On the generative-discriminative tradeoff approach: Interpretation, asymptotic efficiency and classification performance. *Comput. Statist. Data Anal. 54*, 438–451.

Yan, P. K., Khan, S. M., and Shah, M. 2007. 3D model based object class detection in an arbitrary view. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'07)*.

Yang, B., Mei, T., Sun, L.-F., Yang, S.-Q., and Hua, X.-S. 2008a. Free-shaped video collage. In *Proceedings of the 14th International Conference on Advances in Multimedia Modeling*.

Yang, J. C., Yu, K., Gong, Y. H., and Huang, T. 2009. Linear spatial pyramid matching using sparse coding for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*.

Yang, L., Jin, R., Sukthankar, R., and Jurie, F. 2008b. Unifying discriminative visual codebook generation with classifier training for object category recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*.

Yang, Y. and Ramanan, D. 2011. Articulated pose estimation with flexible mixtures-of-parts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*.

Yao, B. Z., Yang, X., Lin, L., Lee, M. W., and Zhu, S. C. 2010. I2T: Image parsing to text description. *Proc. IEEE. 98*, 1485–1508.

Yeh, T., Lee, J. J., and Darrell, T. 2009. Fast concurrent object localization and recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*.

Yu, C. N. J. and Joachims, T. 2009. Learning structural svms with latent variables. In *Proceedings of the International Conference on Machine Learning (ICML'09)*.

Zhang, C., Liu, J., Tian, Q., Xu, C., Lu, H., and Ma, S. 2011a. Image classification by non-negative sparse coding, low-rank and sparse decomposition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*.

Zhang, D. Q. and Chang, S. F. 2006. A generative-discriminative hybrid method for multi-view object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06)*.

Zhang, J., Marszałek, M., Lazebnik, S., and Schmid, C. 2007. Local features and kernels for classification of texture and object categories: A comprehensive study. *Int. J. Comput. Vis. 73*, 213–238.

Zhang, J. G., Huang, K. Q., Yu, Y. N., and Tan, T. N. 2011b. Boosted local structured hog-lbp for object localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*.

Zhang, Z. Q., Cao, Y., Salvi, D., Oliver, K., Waggoner, J., and Wang, S. 2010. Free-shape subwindow search for object localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*.

Zheng, W. S., Gong, S. G., and Xiang, T. 2009. Quantifying contextual information for object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'09)*.

Zhu, L., Chen, Y., Lin, C., and Yuille, A. 2011. Max margin learning of hierarchical configural deformable templates (hcdts) for efficient object parsing and pose estimation. *Int. J. Comput. Vis. 93*, 1–21.

Zhu, L., Chen, Y. H., Yuille, A., and Freeman, W. 2010. Latent hierarchical structural learning for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*.

Zhu, M. 2004. Recall, precision and average precision. Working paper, University of Waterloo.

Zhu, Q., Yeh, M. C., Cheng, K. T., and Avidan, S. 2006. Fast human detection using a cascade of histograms of oriented gradients. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06)*.

Zhu, S.-C. and Mumford, D. 2006. A stochastic grammar of images. *Foundations Trends Comput. Graph. Vis. 2*, 259–362.

Received May 2011; revised September 2012; accepted January 2013