Survey of The Problem of Object Detection In Real Images

Dilip K. Prasad *dilipprasad@gmail.com*
School of Computer Engineering
Nanyang Technological University
Singapore, 639798

Abstract

Object detection and recognition are important problems in computer vision. Since these problems are meta-heuristic, despite a lot of research, practically usable, intelligent, real-time, and dynamic object detection/recognition methods are still unavailable. The accuracy level of any algorithm or even Google glass project is below 16% for over 22,000 object categories. With this accuracy, it's practically unusable. This paper reviews the various aspects of object detection and the challenges involved. The aspects addressed are feature types, learning model, object templates, matching schemes, and boosting methods. Most current research works are highlighted and discussed. Decision making tips are included with extensive discussion of the merits and demerits of each scheme. The survey presented in this paper can be useful as a quick overview and a beginner's guide for the object detection field. Based on the study presented here, researchers can choose a framework suitable for their own specific object detection problems and further optimize the chosen framework for better accuracy in object detection.

Keywords: Boosting, Object Detection, Machine learning, Survey.

1. INTRODUCTION

Object detection is a technologically challenging and practically useful problem in the field of computer vision. Object detection deals with identifying the presence of various individual objects in an image. Great success has been achieved in controlled environment for object detection/recognition problem but the problem remains unsolved in uncontrolled places, in particular, when objects are placed in arbitrary poses in cluttered and occluded environment. As an example, it might be easy to train a domestic help robot to recognize the presence of coffee machine with nothing else in the image. On the other hand imagine the difficulty of such robot in detecting the machine on a kitchen slab that is cluttered by other utensils, gadgets, tools, etc. The searching or recognition process in such scenario is very difficult. So far, no effective solution has been found for this problem.

A lot of research is being done in the area of object recognition and detection during the last two decades. The research on object detection is multi-disciplinary and often involves the fields of image processing, machine learning, linear algebra, topology, statistics/probability, optimization, etc. The research innovations in this field have become so diverse that getting a primary first hand summary of most state-of-the-art approaches is quite difficult and time consuming,

This paper is an effort to briefly summarize the various aspects of object detection and the main steps involved for most object detection algorithm or system. Section 2 provides brief introduction about the generic object detection framework and the importance of this study. Section 3 discusses various types of features used as key points for learning and subsequent object detection. Section 4 elaborates on generative and discriminative learning and comparison among them. Section 5 briefly discuss about the various types of representation used for storing the features after the machine learning stage. Various types of matching schemes used by various algorithms for object detection have been discussed in Section 6. Section 7 elaborates about boosting steps of object detection framework. The paper is concluded in Section 8.

2. PURPOSE AND SCOPE OF THE STUDY

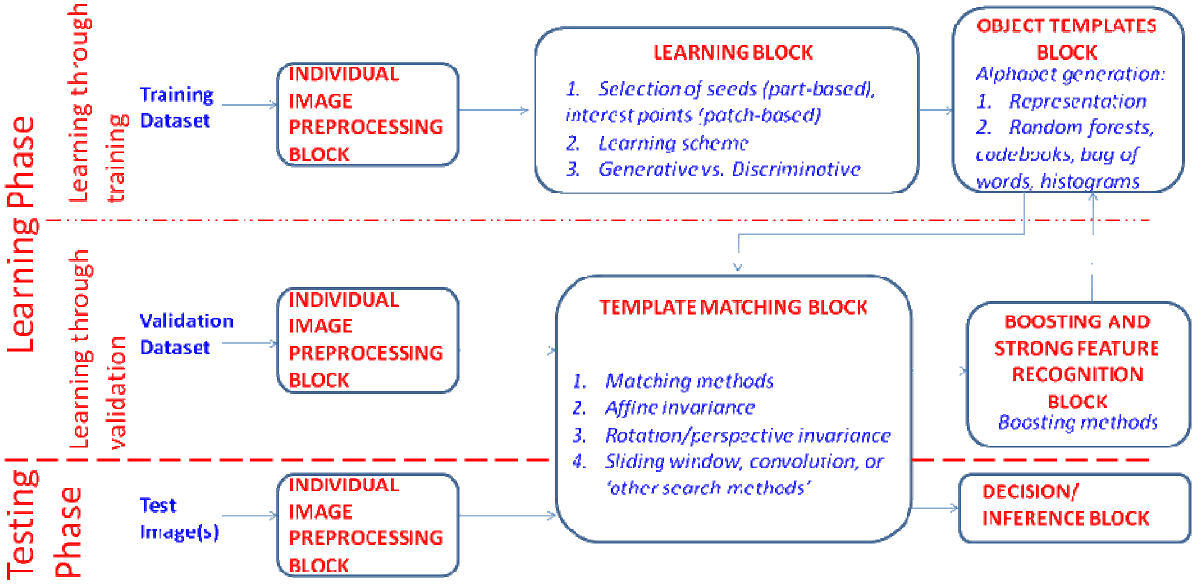

Learning Phase
Learning through training
Training Dataset
INDIVIDUAL IMAGE PREPROCESSING BLOCK
LEARNING BLOCK
1. Selection of seeds (part-based), interest points (patch-based)
2. Learning scheme
3. Generative vs. Discriminative
OBJECT TEMPLATES BLOCK
Alphabet generation:
1. Representation
2. Random forests, codebooks, bag of words, histograms
Learning through validation
Validation Dataset
INDIVIDUAL IMAGE PREPROCESSING BLOCK
TEMPLATE MATCHING BLOCK
1. Matching methods
2. Affine invariance
3. Rotation/perspective invariance
4. Sliding window, convolution, or 'other search methods'
BOOSTING AND STRONG FEATURE RECOGNITION BLOCK
Boosting methods
Testing Phase
Test Image(s)
INDIVIDUAL IMAGE PREPROCESSING BLOCK
DECISION/ INFERENCE BLOCK

FIGURE 1: Basic block diagram of a typical object detection/recognition system.

In order to facilitate the discussion about the methods and ideas of various research works, we first present a general block diagram applicable to any object detection/recognition method in FIGURE 1 Specific methods proposed by various researchers may vary slightly from this generic block diagram.

Any such algorithm can be divided into two different phases, viz. learning phase and testing phase. In the learning phases, the machine uses a set of images which contains objects belonging to specific pre-determined class(es) in order to learn to identify the objects belonging to those classes. Once the algorithm has been trained for identifying the objects belonging to the specified classes, in the testing phase, the algorithm uses its knowledge to identify the specified class objects from the test image(s).

The algorithm for learning phase can be further subdivided into two parts, viz. learning through training and learning through validation. A set of images containing objects of the specified classes, called the training dataset, is used to learn the basic object templates for the specified classes. Depending upon the type of features (edge based features or patch based features), the training images are pre-processed and passed into the learning block. The learning block then learns the features that characterize each class. The learnt object features are then stored as object templates. This phase is referred to as 'learning through training'. The object templates learnt in this stage are termed as weak classifiers. The learnt object templates are tested against the validation dataset in order to evaluate the existing object templates. By using boosting techniques, the learnt object templates are refined in order to achieve greater accuracy while testing. This phase is referred to as 'learning through validation' and the classifiers obtained after this stage are called strong classifiers.

The researchers have worked upon many specific aspects of the above mentioned system. Some examples include the choice of feature type (edge based or patch based features), the method of generating the features, the method of learning the consistent features of an object class, the specificity of the learning scheme (does it concentrate on inter-class variability or intra-class variability), the representation of the templates, the schemes to find a match between a test/validation image and an object template (even though the size and orientation of an object in the test image may be different from the learnt template), and so on. The following discussion considers one aspect at a time and provides details upon the work done in that aspect.

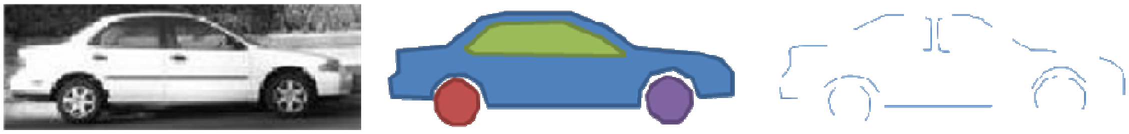
3. FEATURE TYPES

Most object detection and recognition methods can be classified into two categories based on the feature type they use in their methods. The two categories are edge-based feature type and patch based feature type. It is notable that some researchers have used a combination of both the edge-based and patch-based features for object detection [1-5]. In our opinion, using a combination of these two features shall become more and more prevalent in future because such scheme would yield a system that derives the advantages of both the feature types. A good scheme along with the advances in computational systems should make it feasible to use both feature types in efficient and semi-real time manner.

3.1 Edge-based features

The methods that use edge-based feature type extract the edge map of the image and identify the features of the object in terms of edges. Some examples include [1, 2, 6-22]. Using edges as features is advantageous over other features due to various reasons. As discussed in [6], they are largely invariant to illumination conditions and variations in objects' colors and textures. They also represent the object boundaries well and represent the data efficiently in the large spatial extent of the images.

In this category, there are two main variations: use of the complete contour (shape) of the object as the feature [7-12, 14, 17] and use of collection of contour fragments as the feature of the object [1, 2, 6, 13-20, 23, 24]. FIGURE 2 shows an example of complete contour and collection of contours for an image.



(a) Example image

(b) Contour (shape) as feature

(c) Contour fragments as feature

FIGURE 2: Edge-based feature types for an example image

The main motivation of using the complete contours as features is the robustness of such features to the presence of clutter [6, 11, 17, 25]. One of the major concerns regarding such feature type is the method of obtaining the complete contours (especially for training images). In real images, typically incomplete contours are inevitable due to occlusion and noise. Various researchers have tried to solve this problem to some extent [7, 11, 12, 14, 17, 20]. Hamsici [7] identified a set of landmark points from the edges and connected them to obtain a complete shape contour. Schindler [11] used segmenting approaches [26, 27] to obtain closed contours from the very beginning (he called the areas enclosed by such closed contours as super pixels). Ferrari [17, 20] used a sophisticated edge detection method that provides better edges than contemporary methods for object detection. These edges were then connected across the small gaps between them to form a network of closed contours. Ren [14] used a triangulation to complete the contours of the objects in natural images, which are significantly difficult due to the presence of background clutter. Hidden state shape model was used by Wang [28] in order to detect the contours of articulate and flexible/polymorphic objects. It is noticeable that all of these methods require additional computation intensive processing and are typically sensitive to the choice of various empirical contour parameters. The other problem involving such feature is that in the test and validation images, the available contours are also incomplete and therefore the degree of match with the complete contour is typically low [11]. Though some measures, like kernel based [7, 29] and histogram based methods [8, 9], can be taken to alleviate this problem, the detection of the severely occluded objects is still very difficult and unguaranteed [30]. Further, such features are less capable of incorporating the pose or viewpoint changes, large intra-class variability, articulate objects (like horses) and flexible/polymorphic objects (like cars) [11, 17, 20].

This can be explained as follows. Since this feature type deals with complete contours, even though the actual impact of these situations is only on some portions of the contour, the complete contour has to be trained.

On the other hand, the contour fragment features are substantially robust to occlusion if the learnt features are good in characterizing the object [1, 6, 8, 16, 17, 20, 31]. They are less demanding in computation as well as memory as the contour completion methods need not be applied and relatively less data needs to be stored for the features. The matching is also expected to be less sensitive to occlusion [6, 32]. Further, special cases like viewpoint changes, large intra-class variability, articulate objects and flexible/polymorphic objects can be handled efficiently by training the fragments (instead of the complete contour) [2, 6, 8, 17, 20, 32].

However, the performance of the methods based on contour fragment features significantly depends upon the learning techniques. While using these features, it is important to derive good feature templates that represent the object categories well (in terms of both inter-class and intra-class variations) [1, 33]. Learning methods like boosting [31, 33-54] become very important for such feature types.

The selection of the contour fragments for characterizing the objects is an important factor and can affect the performance of the object detection/recognition method. While all the contour fragments in an image cannot be chosen for this purpose, it has to be ensured that the most representative edge fragments are indeed present and sufficient local variation is considered for each representative fragment. In order to look for such fragments, Opelt [1] used large number of random seeds that are used to find the candidate fragments and finally derives only two most representative fragments as features. Shotton [6] on the other hand generated up to 100 randomly sized rectangular units in the bounding box of the object to look for the candidate fragments. It is worth noting that the method proposed in [1] becomes computationally very expensive if more than two edge fragments are used as features for an object category. While the method proposed by Shotton [6] is computationally efficient and expected to be more reliable as it used numerous small fragments (as compared to two most representative fragments), it is still limited by the randomness of choosing the rectangular units. Other computationally efficient way of approximating the contour fragments is by using dominant points or key points of the contours [55-59], guideline to choose suitable dominant point detection method has been given in [57, 60].

On the other hand, Chia [15] used some geometrical shape support (ellipses and quadrangles) in addition to the fragment features for obtaining more reliable features. Use of geometrical structure, relationship between arcs and lines, and study of structural properties like symmetry, similarity and continuity for object retrieval were proposed in [61]. Though the use of geometrical shape (or structure) [62-65] for estimating the structure of the object is a good idea, there are two major problems with the methods in [15, 61]. First problem is that some object categories may not have strong geometrical (elliptic [66, 67] and quadrangle) structure (example horses) and the use of weak geometrical structure may not lead to robust descriptors of such objects. Though [15] demonstrates the applicability for animals, the geometrical structure derived for animals is very generic and applicable to many classes. Thus, the inter-class variance is poor. The classes considered in [15], viz., cars, bikes and four-legged animals (four-legged animals is considered a single class) are very different from each other. Similarly, [61] concentrates on logos and the images considered in [61] have white background, with no natural background clutter and noise. Its performance may degrade significantly in the presence of noise and natural clutter. The second problem is that sometimes occlusion or flexibility of the object may result in complete absence of the components of geometrical structure. For example, if the structural features learnt in [61] are occluded, the probability of detecting the object is very low. Similarly, if the line features learnt in [15], used for forming the quadrangle are absent, the detection capability may reduce significantly.

Though we strongly endorse the idea of using geometric shapes for object detection [68], we suggest that such information should not be used as the only features for object detection. In addition, they can be used to derive good fragment features and reduce the randomness of selection of the fragments.

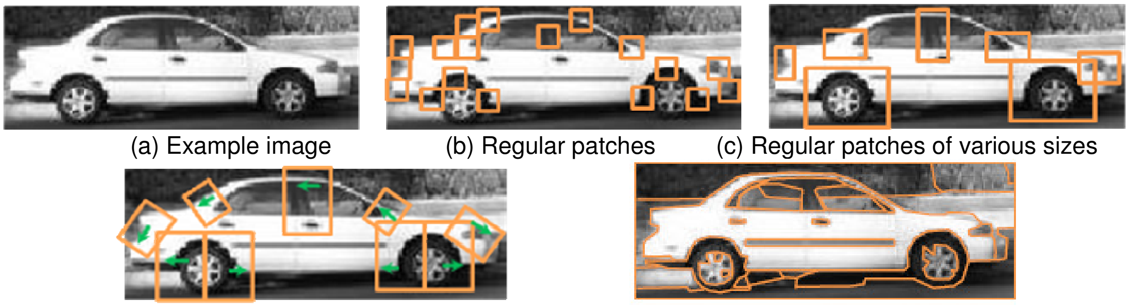
3.2 Patch-based features

The other prevalent feature type is the patch based feature type, which uses appearance as cues. This feature has been in use since more than two decades [69], and edge-based features are relatively new in comparison to it. Moravec [69] looked for local maxima of minimum intensity gradients, which he called corners and selected a patch around these corners. His work was improved by Harris [70], which made the new detector less sensitive to noise, edges, and anisotropic nature of the corners proposed in [69].

In this feature type, there are two main variations:

1) Patches of rectangular shapes that contain the characteristic boundaries describing the features of the objects [1, 71-76]. Usually, these features are referred to as the local features.

2) Irregular patches in which, each patch is homogeneous in terms of intensity or texture and the change in these features are characterized by the boundary of the patches. These features are commonly called the region-based features.



(a) Example image (b) Regular patches (c) Regular patches of various sizes

(d) Oriented regular patches (e) Irregular region patches

FIGURE 3: Patch-based feature types for an example image. Feature types shown in (b)-(d) are called local features, while the feature type shown in (e) is called region-based features.

FIGURE 3 shows these features for an example image. Subfigures (b)-(d) show local features while subfigure (e) shows region based features (intensity is used here for extracting the region features). As shown in FIGURE 3(b)-(d), the local features may be of various kinds. The simplest form of such features use various rectangular or square local regions of the same size in order to derive the object templates [77]. Such features cannot deal with multi-scaling (appearance of the object in various sizes) effectively. A fixed patch size may not be suitable because of the following reason. If the patch size is small, it may not cover a large but important local feature. Information of such feature may be lost in the smaller patch. On the other hand, if the patch size is large, it may cover more than one independent feature, which may or may not be present simultaneously in other images. Further, there is no way to determine the size that is optimal for all the images and various classes. Another shortcoming is that many small rectangular patches need to be learnt as features and stored in order to represent the object well. This is both computationally expensive and memory intensive.

A better scheme is to use features that may be small or big in order to appropriately cover the size of the local feature such that the features are more robust across various images, learning is better and faster, and less storage is required [78].

A pioneering work was done by Lowe [74], which enabled the use of appropriately oriented variable sized features for describing the object. He proposed a scale invariant feature transformation (SIFT) method. Lowe describes his method of feature extraction in three stages.

He first identified potential corners (key points) using difference of Gaussian function, such that these feature points were invariant to scale and rotation. Next, he identified and selected the corners that are most stable and determined their scale (size of rectangular feature). Finally, he computed the local image gradients at the feature points and used them to assign orientations to the patches. The use of oriented features also enhanced the features' robustness to small rotations. With the use of orientation and scale, the features were transformed (rotated along the suitable orientation and scaled to a fixed size) in order to achieve scale and rotational invariance. In order to incorporate the robustness to illumination and pose/perspective changes, the features were additionally described using the Gaussian weighing function along various orientations.

One of the major concerns in all the above schemes is the identification of good corner points (or key-points) that are indeed representative of the data. This issue has been studied by many researchers [4, 56, 57, 60, 74, 79-81]. Lowe [74] studied the stability of the feature points. However, his proposal would apply to his schema of features only. Carneiro [80] and Comer [82] proposed stability measures that could be applied to wide range and varieties of algorithms.

Another major concern is to describe these local features. Though the features can be directly described and stored by saving the pixel data of the local features, such method is naive and inefficient. Researchers have used many efficient methods for describing these local features. These include PCA vectors of the local feature (like PCA-SIFT) [21, 83], Fischer components [84, 85], wavelets and Gabor filters [13], Eigen spaces [86], kernels [7, 21, 29, 87, 88], dominant points [56-59], etc. It is important to note that though these methods use different tools for describing the features, the main mathematical concept behind all of them is the same except for the dominant points. The concept is to choose sufficient (and yet not many) linearly independent vectors to represent the data in a compressed and efficient manner [13]. Another advantage of using such methods is that each linearly independent vector describes a certain property of the local feature (depending on the mathematical tool used). For example, a Gabor wavelet effectively describes an oriented stroke in the image region [13]. Yet another advantage of such features is that while matching the features in the test images, properties of linear algebra (like linear dependence, orthogonality, null spaces, rank, etc.) can be used to design efficient matching techniques [13].

The region-based features are inspired by segmentation approaches and are mostly used in algorithms whose goal is to combine localization, segmentation, and/or categorization. While intensity is the most commonly used cue for generating region based features [51, 79, 89], texture [2, 89-92], color [91-93], and minimum energy/entropy [94, 95] have also been used for generating these features. It is notable that conceptually these are similar to the complete contours discussed in edge-based features. Such features are very sensitive to lighting conditions and are generally difficult from the perspective of scale and rotation invariance. However, when edge and region based features are combined efficiently, in order to represent the outer boundary and inner common features of the objects respectively, they can serve as powerful tools [2]. Some good reviews of feature types can also be found in [71, 96, 97].

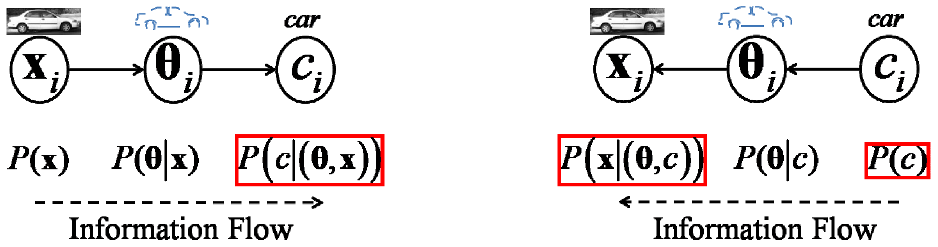
In our opinion, SIFT features provide a very strong scheme for generating robust object templates [74, 98]. It is worth mentioning that though SIFT and its variants were proposed for patch-based features, they can be adapted to edge-fragments based features too. Such adaptation can use the orientation of edges to make the matching more efficient and less sensitive to rotational changes. Further, such scheme can be used to incorporate articulate and flexible/polymorphic objects in a robust manner.

It has been argued correctly by many researchers that a robust object detection and characterization scheme shall typically require more than one feature types to obtain good performance over large number of classes [1, 2, 5, 17, 18, 20, 50, 99-104]. Thus, we shall use region features along with contour fragments. As compared to [1], which has used only one kind of object template for making the final decision, we shall use a combined object template that stores edge, shape, and region features and assigns a strength value to each feature so that combined probabilistic decision can be made while testing. Such scheme shall ensure that potential objects are identified more often, though the trust (likelihood) may vary and the decision can be made by choosing appropriate threshold. This shall be especially useful in severely occluded or noisy images.

4. GENERATIVE MODEL VS. DISCRIMINATIVE MODEL OF LEARNING

The relationship (mapping) between the images and the object classes is typically non-linear and non-analytic (no definite mathematical model applicable for all the images and all the object classes is available). Thus, typically this relationship is modeled using probabilistic models [105]. The images are considered as the observable variables, the object classes are considered as the state variables, and the features are considered as intermediate (sometimes hidden) variables. Such modeling has various advantages. First, it provides a generic framework which is useful for both the problems of object detection and recognition (and many other problems in machine vision and outside it). Second, such framework can be useful in evaluating the nature and extent of information available while training, which subsequently helps us to design suitable training strategies.

The probabilistic models for our problems can be generally classified into two categories, viz. discriminative models and generative models [106-110]. It shall be helpful to develop a basic mathematical framework for understanding and comparing the two models. Let the observable variables (images) be denoted by $\mathbf{x}_i$, $i = 1 \text{ to } N$, where N is the number of training images. Let the corresponding state variables (class labels) be denoted as c_i and the intermediate variables (features/ feature descriptors) be denoted as $\boldsymbol{\theta}_i$. Accordingly, a simplistic graphical representation [107] of the discriminative and generative models is presented in FIGURE 4.



(a) Discriminative model

(b) Generative model

FIGURE 4: Graphical illustration of the discriminative and generative models. The probabilities in boxes are the model defining probabilities for the respective models.

As seen in the FIGURE 4, the discriminative model uses a map from the images to the class labels, and thus the flow of information is from the observables (images) to the state variables (class labels) [107]. Considering the joint probability $P(c,\boldsymbol{\theta},\mathbf{x})$, discriminative models expand $P(c,\boldsymbol{\theta},\mathbf{x})$ as $P(c,\boldsymbol{\theta},\mathbf{x}) = P\left(c|(\boldsymbol{\theta},\mathbf{x})\right)P(\boldsymbol{\theta}|\mathbf{x})P(\mathbf{x})$. Thus, $P\left(c|(\boldsymbol{\theta},\mathbf{x})\right)$ is the model defining probability [106] and the training goal is:

$$P\left(c|(\boldsymbol{\theta},\mathbf{x})\right) = \begin{cases} \alpha & \text{if } \mathbf{x} \text{ contains object of class } c \\ \beta & \text{otherwise} \end{cases} \tag{1}$$

Ideally, $\alpha = 1$ and $\beta = 0$. Indeed, practically this is almost impossible to achieve, and values between [0,1] are chosen for α and β.

In contrast, the generative model uses a map from the class labels to the images, and thus the flow of information is from the state variables (class labels) to the observables (images) [107]. Generative models use the expansion of the joint probability $P(c,\boldsymbol{\theta},\mathbf{x}) = P\left(\mathbf{x}\middle|(\boldsymbol{\theta},c)\right)P(\boldsymbol{\theta}|c)P(c)$.

Thus, $P\left(\mathbf{x}\middle|(\boldsymbol{\theta},c)\right)$ and $P(c)$ are the model defining probabilities [106] and the training goal is:

$$P\left(\mathbf{x}\middle|(\boldsymbol{\theta},c)\right)P(c) = \begin{cases} \alpha & \text{if } \mathbf{x} \text{ contains object of class } c \\ \beta & \text{otherwise} \end{cases} \tag{2}$$

Ideally, $\alpha = 1$ and $\beta = 0$. Indeed, practically this is almost impossible to achieve, and some realistic values are chosen for α and β. It is important to note that in unsupervised methods, the prior probability of classes, $P(c)$ is also unknown.

Further mathematical details can be found in [106, 107]. The other popular model is the descriptive model, in which every node is observable and is interconnected to every other node. It is obvious that the applicability of this model to the considered problem is limited. Therefore, we do not discuss this model any further. It shall suffice to make a note that such models are sometimes used in the form of conditional random fields/forests [12, 51, 90].

With the above mentioned mathematical structure as a reference, we can now compare the discriminative and generative models from various aspects, in the following sub-sections.

4.1 Comparison of their functions

As the name indicates, the main function of the discriminative models is that for a given image, it should be able to discriminate the possibility of occurrence of one class from the rest. This is evident by considering the fact that the probability $P\left(c\middle|(\boldsymbol{\theta},\mathbf{x})\right)$ is the probability of discriminating the class labels c for a given instance of image $\mathbf{x}$. On the other hand, the main function of generative models is to be able to predict the possibility of generating the object features $\boldsymbol{\theta}$ in an image $\mathbf{x}$ if the occurrence of the class c is known. In other words, the probabilities $P\left(\mathbf{x}\middle|(\boldsymbol{\theta},c)\right)P(c)$ together represent the probability of generating random instances of $\mathbf{x}$ conditioned to class c. In this context, it is evident that while discriminative models are expected to perform better for object detection purposes, generative models are expected to perform better for object recognition purposes [18]. This can alternatively be understood as the generative models are used to learn class models (and be useful even in large intra-class variation) [50, 75, 111, 112] while discriminative models are useful for providing maximum inter-class variability [112].

4.2 Comparison of the conditional probabilities of the intermediate variables

In the discriminative models, the intermediate conditional probability is $P(\boldsymbol{\theta}|\mathbf{x})$, while in the generative models, the intermediate conditional probability is $P(\boldsymbol{\theta}|c)$. Since we are interested in the joint probability $P(c,\boldsymbol{\theta},\mathbf{x})$, the probabilities $P(\boldsymbol{\theta}|\mathbf{x})$ and $P(\boldsymbol{\theta}|c)$ play an important role, though they do not appear in the training goals. In the discriminative models, $P(\boldsymbol{\theta}|\mathbf{x})$ represents the strength of the features $\boldsymbol{\theta}$ in representing the image well [17, 20], while in the generative models, $P(\boldsymbol{\theta}|c)$ represent the strength of features in representing the class well. Though ideally we would like to maximize both, depending upon the type of feature and the problem, the maximum value of these probabilities is typically less than one. Further, it is difficult to quantitatively measure these probabilities in practice. In our opinion, while the shape features (closed contours) and region features (irregular patches) are more representative of the class (the object's 3-dimensional or 2-dimensional model), the edge fragments and local features are more representative of the images [1, 50]. Thus, while shape and region features are widely used for segmentation and recognition, local features and edge fragments have been used more often for object detection [17, 18, 20, 50, 101]. Considering this argument, though most methods that use multiple feature types choose these feature types randomly, we recommend to choose a combination of two feature types where one feature is robust for characterizing the image, while the other is good in characterizing the class. In this regard, combining edge fragments and region features is the combination that is easiest to handle practically. Due to this many new methods have used a combination of these features [2, 5, 102-104].

4.3 Training data size and supervision

Mathematically, the training data size required for generative model is very large (at least more than the maximum dimension of the observation vector $\mathbf{x}$). On the other hand, discriminative models perform well even if the training dataset is very small (more than a few images for each class type). This is expected because the discriminative models invariably use supervised training dataset (the class label is specifically mentioned for each image). On the other hand, generative models are unsupervised (semi-supervised, at best) [113]. Not only the posterior probability $P\left(\mathbf{x}\middle|(\boldsymbol{\theta},c)\right)$ is unknown, the prior probability of the classes $P(c)$ is also unknown for the generative models [106]. Another point in this regard is that since generative models do not require supervision and the training dataset can be appended incrementally [18, 106, 111] as vision system encounters more and more scenarios, generative models are an important tool for expanding the knowledge base, learning new classes, and keeping the overall system scalable in its capabilities.

4.4 Comparison of accuracy and convergence

The discriminative models usually converge fast and correctly (explained by supervised dataset). If the size of training dataset is asymptotically large, the convergence is guaranteed for the generative models as well. However, such convergence may be correct convergence or misconvergence. If the generative models converge correctly, then the accuracy of generative models is comparable to the accuracy of the discriminative models. But, if there has been a misconvergence, then the accuracy of the generative models is typically poorer than the discriminative models [114]. Since the dataset is typically finite, and in most cases small, it is important to compare the accuracy of these models when the dataset is finite. Mathematical analysis has shown that in such cases, the accuracy of the generative models is always lower than the discriminative methods [114]. It is notable that due to their basic nature, generative models provide good recall but poor precision, while discriminative models provide poorer recall but good precision. The restrictive nature of generative models has prompted more and more researchers to consider discriminative models [1, 17, 20, 93, 115-121]. On the other hand, considering the scalability, generalization properties, and non-supervised nature of generative models, other researchers are trying to improve the performance of generative models by using partial supervision or coupling the generative models and discriminative models in various forms [4, 18, 31, 75, 93, 111, 113, 122].

4.5 Learning methods

Generative models use methods like Bayesian classifiers/networks [18, 31, 75, 111], likelihood maximization [111, 122], and expectation maximization [4, 93, 113, 122]. Discriminative models typically use methods like logistic regression, support vector machines [17, 20, 93, 115-119], and k-nearest neighbors [93, 120, 121]. The k-nearest neighbors scheme can also be used for multi-class problems[109, 123-141] directly, as demonstrated in [120]. Boosting schemes are also examples of methods for learning discriminative models [1], though they are typically applied on already learnt weak features (they shall be discussed later in greater detail). In the schemes where generative and discriminative models are combined [93, 142], there are two main variations: generative models with discriminative learning [4, 102, 106], and discriminative models with generative learning [107]. In the former, typically maximum likelihood or Bayesian approaches are combined with boosting schemes or incremental learning schemes [4, 50, 102, 106, 111], while in the latter, usual discriminative schemes are augmented by 'generate and test' schemes in the feedback loop [107, 143]. Learning scheme can be offline or online based on the demand of the application [144]. Online learning is now feasible due to advancement of cloud technology [145].

5. OBJECT TEMPLATES AND THEIR REPRESENTATION

The learning method has to learn a mapping between the features and the classes. Typically, the features are extracted first, which is followed by either the formation of class models (in generative models) or the most discriminative features for each class (in discriminative models) or random fields of features in which a cluster represents an object class (descriptive models, histogram based schemes, Hough transform based methods, etc). Based on them, the object templates suitable for each class are learnt and stored for the future use (testing). This section will discuss various forms of object templates used by researchers in computer vision.

While deciding on an object template, we need to consider factors like:

Is the template most representative form of the class (in terms of the aimed specificity, flexibility of the object, intra-class variation, etc)? For example, does it give the required intra-class and inter-class variability features? Does it need to consider some common features among various classes or instances of hierarchical class structure? Does it need to consider various poses and/or perspectives? Does it need to prioritize certain features (or kind of features)?

Is the model representation an efficient way of storing and using the template? Here, memory and computations are not the only important factors. We need to also consider if the representation enables good decision mechanisms.

The above factors will be the central theme in discussing the specific merits and demerits of the various existing object templates. We begin with the object templates that use the spatial location of the features. Such templates specifically represent the relative position of the features (edge fragments, patches, regions) in the image space. For this, researchers typically represent each feature using a single representative point (called the centroid) and specify a small region in which the location of the centroid may vary in various objects belonging to the same class [1, 6]. Then all the centroids are collected together using a graph topology. For example some researchers have used a cyclic/chain topology [11]. This simplistic topology is good to represent only the external continuous boundary of the object. Due to this, it is also used for complete contour representation, where the contour is defined using particular pivot points which are joined to form the contour [11]. Such a topology may fail if the object is occluded at one of the centroid locations, as the link between the chain is not found in such case and the remaining centroids are also not detected as a consequence. Further, if some of the characteristic features are inside the object boundary, deciding the most appropriate connecting link between the centroids of the external and internal boundaries may be an issue and may impact the performance of the overall algorithm. Other topology in use is the constellation topology [111, 146, 147], in which a connected graph is used to link all the centroids. A similar representation is being called multi-parts-tree model in [94], though the essentials are same. However, such topology requires extra computation in order to find an optimal (neither very deep nor very wide) representation. Again, if the centroids that are linked to more than one centroid are occluded, the performance degrades (though not as strongly as the chain topology). The most efficient method in this category is the star topology, in which a central (root) node is connected to all the centroids [1, 6, 8, 76]. The root node does not correspond to any feature or centroid and is just a virtual node (representing the virtual centroid of the complete object). Thus, this topology is able to deal with occlusion better than the other two topologies and does not need any extra computation for making the topology.

Other methods in which the features are described using transformation methods (like the kernel based methods, PCA, wavelets, etc., discussed in section 3), the independent features can be used to form the object templates. The object templates could be binary vectors that specify if a particular feature is present in an object or not. Such object templates are called bag-of-words, bag of visual words, or bag of features [1, 95, 115, 116, 119, 148-150]. All the possible features are analogous to visual words, and specific combinations of words (in no particular order) together represent the object classes. Such bag of words can also be used for features like colors, textures, intensity, shapes [95], physical features (like eyes, lips, nose for faces, and wheels, headlights, mirrors for cars) etc. [93, 149, 151]. As evident, such bag of words is a simple yet powerful technique for object recognition and detection but may perform poorly for object localization and segmentation. Fusing the generic object template and visual saliency for salient object detection has been explored by Chang et. al. [152].As opposed to them, spatial object templates are more powerful for image localization and segmentation.

In either of the above cases, the object templates can also be in the form of codebooks [1, 6, 17, 20, 75, 76, 150, 153]. A codebook contains a specific code of features for each object class. The code contains the various features that are present in the corresponding class, where the sequence of features may follow a specific order or not. An unordered codebook is in essence similar to the concept of bag of words, where the bag of words may have greater advantage in storing and recalling the features and the object templates. However, codebooks become more powerful if the features in the code are ordered. A code in the order of appearance of spatial templates can help in segmentation [6], while a code in the order of reliability or strength of a feature for a class shall make the object detection and recognition more robust.

Other hierarchical (tree like) object templates may be used to combine the strengths of both the codebooks and bag of words, and to efficiently combine various feature types [4, 18, 75, 84, 89, 92, 102, 113, 122, 147, 150, 154].

Another important method of representing the object templates is based on random forests/fields [90, 143, 155]. In such methods, no explicit object template is defined. Instead, in the feature space (where each feature represents one dimension), clusters of images belonging to same object class are identified [74, 75, 155]. These clusters in the feature space are used as the probabilistic object templates [84]. For every test image, its location in feature space and distance from these clusters determine the decision.

We prefer a hierarchical codebook, similar to the multi-parts-tree model [94, 113], which combines at least two feature types. We intend to place the strongest (most consistent and generic) features at the highest level and weaker features in subsequent nodes. Any single path in the hierarchy shall serve as a weak but sufficient object template and typically the hope is that more than one path are traversed if object of the class is present in an image. If all the paths are traversed, the image has a strong presence of the object class. The final inference will be based on the number and depth of the paths traversed. It is worth mentioning that while [94] used a minimization of the energy and Mahalanobis distance of the parts for generating the tree, we shall use the likelihood of each feature independently, and likelihood of each feature conditioned to the presence of higher level features in the tree. We might have considered another hierarchical structure where the strongest (but few) descriptors appear at the leaf nodes and the path towards the root incrementally confirms the presence of the object. But that would either require multiple bottom-up traversals (in order to reach the root) or a top-down traversal with very low initial confidence. On the other hand, the chosen top-down structure will ensure that we begin with a certain degree of confidence (due to the generic features with high likelihood at the highest level, details in section 6) in the presence of the object class and then tweak our confidence as we go further down the tree. If we cannot go further down the tree, we need not look for multiple other traversal paths beginning again from the top.

6. MATCHING SCHEMES AND DECISION MAKING

Once the object templates have been formed, the method should be capable of making decisions (like detecting or recognizing objects in images) for input images (validation and/or test images). We first discuss about the methods of finding a match between the object template and the input image and then discuss about the methods of making the final decision.

Discussion regarding matching schemes is important because of various reasons. While the training dataset can be chosen to meet certain requirements, it cannot be expected that the test images also adhere to those requirements. For example, we may choose that all the training images are of a particular size, illumination condition, contain only single object of interest viewed from a fixed perspective, in uncluttered (white background), etc., such restrictions cannot be imposed on the real test images, which may be of varying size, may contain many objects of interest and may be severely cluttered and occluded and may be taken from various viewpoints. The problem of clutter and occlusion is largely a matter of feature selection and learning methods. Still, they may lead to wrong inferences if improper matching techniques are used. However, making the matching scheme scale invariant, rotation and pose invariant (at least to some degree), illumination independent, and capable of inferring multiple instances of multiple classes is important and has gained attention of many researchers [6, 68, 80, 82, 147, 156-185].

If the features in the object templates are pixel based (for example patches or edges), the Euclidean distance based measures like Hausdorff distance [174, 184, 186, 187] and Chamfer distance [1, 6, 17, 25, 94, 161, 170] provide quick and efficient matching tools. However, the original forms of both these distances were scale, rotation, and illumination dependent. Chamfer distance has become more popular in this field because of a lot of incremental improvement in Chamfer distance as a matching technique. These improvements include making it scale invariant, illumination independent, rotation invariant, and more robust to pose variations and occlusions [1, 6, 17, 25, 94, 161, 170]. Further, Chamfer distance has also been adapted for hierarchical codebooks [161]. In region based features, concepts like structure entropy [95, 188], mutual information [95, 154], and shape correlation have been used for matching and inference [157, 158]. Worth attention is the work by Wang [95] that proposed a combination of local and global matching scheme for region features. Such scheme can perform matching and similarity evaluation in an efficient manner (also capable of dealing with deformation or pose changes) by incorporating the spatial mutual information with the local entropy in the matching scheme. Amiri et. al. have proposed an improved SIFT based matching of potential interest points identified by searching for local peaks in Difference-of-Gaussian (DoG) images[189].

Another method of matching/inferring is to use the probabilistic model in order to evaluate the likelihood ratio [2, 4, 75] or expectation in generative models [105, 113]. Otherwise, correlation between the object template and the input image can be computed or probabilistic Hough transform can be used [77, 92, 93, 118]. Each of these measures is linked directly or indirectly with the defining ratio of the generative model , $P\left(\mathbf{x}\middle|(\boldsymbol{\theta},c)\right)$, which can be computed for an input image and a given class through the learnt hidden variables $\boldsymbol{\theta}$ [13]. For example, in the case of wavelet form of features, $P\left(\mathbf{x}\middle|(\boldsymbol{\theta},c)\right)$ will depend upon the wavelet kernel response to the input image for a particular class [13]. Similarly, the posterior probability can be used for inference in the discriminative models. Or else, in the case of classifiers like SVM, k-nearest neighbors based method, binary classifiers, etc, the features are extracted for the input image and the posterior probability (based on the number of features voted into each class) can be used for inference [17, 20, 84]. If two or more classes have the high posterior probability, multiple objects may be inferred [75, 94]. However, if it is known that only one object is present in an image, refined methods based on feature reliability can be used.

If the object class is represented using the feature spaces, the distance of the image from the clusters in feature space is used for inference. Other methods include histograms corresponding to the features (the number of features that were detected) to decide the object category [74, 84, 149, 155].

7. BOOSTING METHODS - LEARNING WHILE VALIDATION

The weak object templates learnt during training can be made more class specific by using boosting mechanisms in the validation phase [190-211]. Boosting mechanisms typically consider an ensemble of weak features (in the object templates) and gives a boost to the stronger features corresponding to the object class. Technically, boosting method can be explained as follows. Suppose validation images $\mathbf{x}_i$, $i = 1 \text{ to } N$ contain the corresponding class labels $c_i = \pm 1$, where the value 1 indicates that the object of the considered class is present and -1 represents its absence. Let the weak classifier learnt while training be a combination of several individual classifiers $h_j(\cdot)$, $j = 1 \text{ to } J$. Here, $h_j(\cdot)$ operates on the input image and gives an inference/decision regarding the presence/absence of class object. Evidently, $h_j(\cdot)$ is determined by the feature θ_j in the codebook and the inference mechanisms. Further, let us say that we want to extract maximum T strong classifiers. Then most boosting methods can be generally explained using the algorithm below:

Step 1: Initialize the image weights $w_{i,t} = 1/N; \forall i$

Step 2: For $t = 1 \text{ to } T$

Step 2.1: Find the strongest classifier, $h_t(\cdot)$, using the current image weights. For this, first compute the error function for each classifier: $\varepsilon_j = \sum_i \left(w_{i,t} I_{j,i} \right)$, where $I_{j,i} = 1$ if $c_i = h_j(\mathbf{x}_i)$, and 0 otherwise. Here, the index j is used to denote the jth classifier and the index i is used to denote the ith image. Find the classifier that resulted in minimum error (this is the strongest classifier for the weights $w_{i,t}$): $h_t(\cdot) = \arg\left(\min\left(\varepsilon_j \right) \right)$.

Step 2.2: Update the classifier weight for the chosen classifier: $\alpha_t = f\left(\varepsilon_t \right)$, where the function $f(\cdot)$ depends upon the chosen boosting technique and ε_t is the error corresponding to the current strongest classifier $h_t(\cdot)$.

Step 2.3: If a termination condition is satisfied, then go to step 3. The termination condition depends upon the application or the boosting method used.

Step 2.4: Update the weights $w_{i,t+1} = w_{i,t} g(\alpha_t I_i)$. Here, $g(\cdot)$ is the function that changes the weight distribution given to the validation images and is generally called the loss function. The general characteristic of $g(\cdot)$ is that it reduces the weight of the images that resulted in correct classification, so that in the next iteration, the method is less biased towards the current strong feature. Typically, $w_{i,t+1}$ is normalized after computation such that the sum of all the weights is 1.

Step 3: The output of the boosting algorithm is typically specified as the strong classifier $h_{\text{strong}}(\cdot) = \sum_t \alpha_t h_t(\cdot)$.

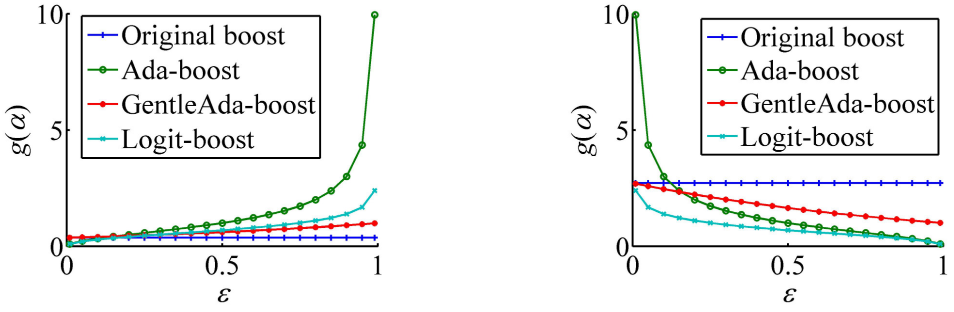
FIGURE 5: Generic algorithm for boosting

It is notable that some features may be repeatedly selected in step 2 of FIGURE 5, which indicates that though the method is getting lesser and lesser biased towards that feature, that feature is strong enough to be selected again and again.

There are many variations of boosting methods, which are typically differentiated based upon their loss function $g(\cdot)$ and the classifier update function $f(\cdot)$. We discuss some prominent methods used often in computer vision. The original boost used a constant value for the classifier update function $f(\cdot) = 1$ and an exponential loss function $g(\alpha_t I_i) = \exp\left(-\alpha_t c_i h_t(\mathbf{x}_i) \right)$ [212, 213]. It was shown that such technique performed marginally better than the random techniques used for selecting the features from a codebook. However, the performance of boosting method was greatly enhanced by the introduction of adaptive boosting (Adaboost) [1, 212-215]. Here, the main difference is the classifier update function $f(\varepsilon_t) = 0.5 \ln\left(\left(1 - \varepsilon_t \right) / \varepsilon_t \right)$. Since the value of $f(\cdot) = 0$ implies no further optimization, the termination condition is set as $\varepsilon_t \geq 0.5$. This boosting method was adapted extensively in the object detection and recognition field. Though it is efficient in avoiding the problem of over-fitting, it is typically very sensitive to noise and clutter.

A variation on the Ada-boost, Logit-boost [212, 213, 216] used similar scheme but a logistic regression function based loss function, $g(\alpha_t I_i) = \ln\left(1+\exp\left(-\alpha_t c_i h_t(\mathbf{x}_i)\right)\right)$. As compared to the Ada-boost, it is more robust to the noisy and cluttered scenarios. This is because as compared to the Ada-boost, this loss function is flatter and provides a softer shift towards the noise images. Another variation on the Ada-boost is the GentleAda-boost [6, 8, 212, 213], which is similar to Ada-boost but uses a linear classifier update function $f(\varepsilon_t) = (1-\varepsilon_t)$. The linear form of the classifier update function ensures that the overall update scheme is not severely prejudiced.

In order to understand and compare the four boosting schemes, we present the plots between the error ε and the loss function (which also incorporates the classifier update function through α) for the four boosting schemes in FIGURE 6. FIGURE 6(a) shows the value of loss function when the chosen classifier gives the correct inference for an image. If the classifier is weak (high error) and yet generates a correct inference for an image, that image is boosted so that the classifier gets boosted. Similarly, FIGURE 6(b) shows the plot when the chosen classifier generates incorrect inference for an image. If the classifier is strong (low error ε) and still generates an incorrect inference for an image, the classifier can be suppressed or weakened by boosting such image.



(a) Loss function when the inference is correct (b) Loss function when the inference is incorrect

FIGURE 6: Comparison of boosting techniques. (a) Loss function when the inference is correct (b) loss function when the inference is incorrect.

It is evident that the desired property is not emulated well by the original boosting, which explains its slight (insignificant) improvement over the random selection of classifiers. On the other hand, Ada-boost is too strict in weakening or boosting the classifiers. Logit-boost and GentleAda-boost demonstrate a rather tempered performance, among whom, evidently Gentle-boost is the least non-linear and indeed the most gentle in weakening or boosting the classifiers. However, in our opinion, Logit-boost is the best among these methods precisely because of its combination of being gentle as well as non-linear. Due to the non-linearity, it is expected to converge faster than the GentleAda-boost and due to its gentle boosting characteristic, it is expected to be more robust than Ada-boost for noisy and cluttered images, where wrong inferences cannot be altogether eliminated.

The convergence of boosting techniques (except the original one) discussed above can be enhanced by using a gradient based approach for updating the weights of the images. Such approach is sometimes referred to as the Gradient-boost [123, 213, 217, 218]. However, this concept can be used within the framework of most boosting approaches. Similar contribution comes from the LP-boost (linear programming boost) methods [36, 212], where concepts of linear programming are used for computing the weights of the images. In both the schemes, the iteration (step 2 of FIGURE 5) is cast as an optimization problem in terms of the loss function, such that the convergence direction and rate can be controlled. Such schemes also reduce the number of control parameters and make boosting less sensitive to them.

A recent work by Mallapragada [219], Semi-boost, is an interesting recent addition to the body of boosting algorithms. While the existing boosting methods assume that every image in the validation dataset is labeled, [219] considers a validation dataset in which only a few images need to be labeled. In this sense, it provides a framework for incorporating semi-supervised boosting. In each iteration (step 2 of FIGURE 5), two major steps are done in addition to and before the mentioned steps. First, each unlabeled image is pseudo-labeled by computing the similarity of the unlabeled images with the labeled images, and a confidence value is assigned to each pseudo-label. Second, the pseudo-labeled images with high confidence values are pooled with the labeled images as the validation set to be used in the remaining steps of the iteration. As the strong features are identified iteratively, the pseudo-labeling becomes more accurate and the confidence of the set of unlabeled data increases. It has been shown in [219] that Semi-boost can be easily incorporated in the existing framework of many algorithms. This method provides three important advantages over the existing methods. First, it can accommodate scalable validation sets (where images may be added at any stage with or without labeling). Second, since semi-boost learns to increase the confidence of labeling the unlabeled images, and not just fitting the features to the labeled data, it is more efficient in avoiding over-fitting and providing better test performances. Third, though not discussed in [219], in our opinion, the similarity and pseudo-labeling schemes should help in identifying the presence of new (unknown) classes, and thus provide class-scalability as well. Although another recent work by Joshi [117] tries to attack the same problem as [219] by using a small seed training set that is completely labeled in order to learn from other unsupervised training dataset, his approach is mainly based on support vector machine (SVM) based learning. It may have its specific advantages, like suitability for multi-class data. However, semi-boost is an important improvement within the boosting algorithms, which have wider applicability than SVM based learning methods.

Another important method in the boosting techniques is the Joint-boost [1, 90], first proposed in [40, 220]. It can handle multi-class inferences directly (as opposed to other boosting techniques discussed above which use binary inference for one class at a time). The basis of joint boosting is that some features may be shared among more than one class [40, 220]. For this, the error metric is defined as $\varepsilon_j = \sum_{\kappa} \sum_{i} \left({}^{\kappa}w_{i,t} \; {}^{\kappa}I_i \right)$, where $\kappa = 1 \text{ to } K$ represents various classes, and the inference ${}^{\kappa}I_i$ is the binary inference for class κ. Thus, instead of learning the class-specific strong features, we can learn strong shared features. Such features are more generic over the classes and very few features are sufficient for representing the classes generically. Typically, the number of sufficient features is the logarithmic value of the number of classes [40, 220]. However, better inter-class distances can be achieved by increasing the number of features. Even then the number of features required for optimal generality and specificity is much lesser than boosting for one class at a time. Such scheme is indeed very beneficial if a bag of words is used for representing the object templates. Joint boost has also been combined with principal component analysis based system in [121] to further improve the speed of training.

8. CONCLUSION

This review paper addresses all the major aspects of an object detection framework. These include feature selection, learning model, object representation, matching features and object templates, and the boosting schemes. For each aspect, the technologies in use and the state-of-the-art research works are discussed. The merits and demerits of the works are discussed and key indicators helpful in choosing a suitable technique are also presented. Thus, the paper presents a concise summary of the state-of-the-art techniques in object detection for upcoming researchers. This study provides a preliminary, concise, but complete background of the object detection problem. Thus, based on this study, for a given problem environment and data availability, a proper framework can be chosen easily and quickly. Background codes/theories can be used from the works cited here relevant to the chosen framework and the focus of research can be dedicated to improving or optimizing the chosen framework for better accuracy in the given problem environment.

9. REFERENCES

[1] A. Opelt, A. Pinz, and A. Zisserman, "Learning an alphabet of shape and appearance for multi-class object detection," *International Journal of Computer Vision,* vol. 80, pp. 16-44, 2008.

[2] Z. Si, H. Gong, Y. N. Wu, and S. C. Zhu, "Learning mixed templates for object recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 272-279.

[3] R. Fergus, P. Perona, and A. Zisserman, "A sparse object category model for efficient learning and exhaustive recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 380-387.

[4] Y. Chen, L. Zhu, A. Yuille, and H. J. Zhang, "Unsupervised learning of probabilistic object models (POMs) for object classification, segmentation, and recognition using knowledge propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 31, pp. 1747-1774, 2009.

[5] J. Shotton, "Contour and texture for visual recognition of object categories," Doctoral of Philosphy, Queen's College, University of Cambridge, Cambridge, 2007.

[6] J. Shotton, A. Blake, and R. Cipolla, "Multiscale categorical object recognition using contour fragments," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 30, pp. 1270-1281, 2008.

[7] O. C. Hamsici and A. M. Martinez, "Rotation invariant kernels and their application to shape analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 31, pp. 1985-1999, 2009.

[8] L. Szumilas and H. Wildenauer, "Spatial configuration of local shape features for discriminative object detection," in *Lecture Notes in Computer Science* vol. 5875, ed, 2009, pp. 22-33.

[9] L. Szumilas, H. Wildenauer, and A. Hanbury, "Invariant shape matching for detection of semi-local image structures," in *Lecture Notes in Computer Science* vol. 5627, ed, 2009, pp. 551-562.

[10] M. P. Kumar, P. H. S. Torr, and A. Zisserman, "OBJCUT: Efficient Segmentation Using Top-Down and Bottom-Up Cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 32, pp. 530-545, 2009.

[11] K. Schindler and D. Suter, "Object detection by global contour shape," *Pattern Recognition,* vol. 41, pp. 3736-3748, 2008.

[12] N. Alajlan, M. S. Kamel, and G. H. Freeman, "Geometry-based image retrieval in binary image databases," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 30, pp. 1003-1013, 2008.

[13] Y. N. Wu, Z. Si, H. Gong, and S. C. Zhu, "Learning Active Basis Model for Object Detection and Recognition," *International Journal of Computer Vision,* pp. 1-38, 2009.

[14] X. Ren, C. C. Fowlkes, and J. Malik, "Learning probabilistic models for contour completion in natural images," *International Journal of Computer Vision,* vol. 77, pp. 47-63, 2008.

[15] A. Y. S. Chia, S. Rahardja, D. Rajan, and M. K. H. Leung, "Structural descriptors for category level object detection," *IEEE Transactions on Multimedia,* vol. 11, pp. 1407-1421, 2009.

[16] J. Winn and J. Shotton, "The layout consistent random field for recognizing and segmenting partially occluded objects," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 37-44.

[17] V. Ferrari, T. Tuytelaars, and L. Van Gool, "Object detection by contour segment networks," in *Lecture Notes in Computer Science* vol. 3953, ed, 2006, pp. 14-28.

[18] K. Mikolajczyk, B. Leibe, and B. Schiele, "Multiple object class detection with a generative model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 26-33.

[19] R. C. Nelson and A. Selinger, "Cubist approach to object recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 1998, pp. 614-621.

[20] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid, "Groups of adjacent contour segments for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 30, pp. 36-51, 2008.

[21] S. Ali and M. Shah, "A supervised learning framework for generic object detection in images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2005, pp. 1347-1354.

[22] I. A. Rizvi and B. K. Mohan, "Improving the Accuracy of Object Based Supervised Image Classification using Cloud Basis Function Neural Network for High Resolution Satellite Images," *International Journal of Image Processing (IJIP),* vol. 4, pp. 342-353, 2010.

[23] D. K. Prasad and M. K. H. Leung, "A hybrid approach for ellipse detection in real images," in *2nd International Conference on Digital Image Processing*, Singapore, 2010, pp. 75460I-6.

[24] D. K. Prasad and M. K. H. Leung, "Methods for ellipse detection from edge maps of real images," in *Machine Vision - Applications and Systems*, F. Solari, M. Chessa, and S. Sabatini, Eds., ed: InTech, 2012, pp. 135-162.

[25] P. F. Felzenszwalb, "Learning models for object recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2001, pp. 1056-1062.

[26] E. Borenstein and S. Ullman, "Learning to segment," in *Lecture Notes in Computer Science* vol. 3023, ed, 2004, pp. 315-328.

[27] E. Borenstein and J. Malik, "Shape guided object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 969-976.

[28] J. Wang, V. Athitsos, S. Sclaroff, and M. Betke, "Detecting objects of variable shape structure with Hidden State Shape Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 30, pp. 477-492, 2008.

[29] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2006, pp. 13-13.

[30] D. K. Prasad, C. Quek, and M. K. H. Leung, "Fast segmentation of sub-cellular organelles," *International Journal of Image Processing (IJIP),* vol. 6, pp. 317-325, 2012.

[31] Y. Amit, D. Geman, and X. Fan, "A coarse-to-fine strategy for multiclass shape detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 26, pp. 1606-1621, 2004.

[32] J. Shotton, A. Blake, and R. Cipolla, "Contour-based learning for object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2005, pp. 503-510.

[33] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer, "Generic object recognition with boosting," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 28, pp. 416-431, 2006.

[34] Y. Freund, "Boosting a Weak Learning Algorithm by Majority," *Information and Computation,* vol. 121, pp. 256-285, 1995.

[35] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 23, pp. 349-361, 2001.

[36] A. Demiriz, K. P. Bennett, and J. Shawe-Taylor, "Linear programming boosting via column generation," *Machine Learning,* vol. 46, pp. 225-254, 2002.

[37] S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 26, pp. 1475-1490, 2004.

[38] R. Fergus, P. Perona, and A. Zisserman, "A visual category filter for google images," in *Lecture Notes in Computer Science* vol. 3021, ed, 2004, pp. 242-256.

[39] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer, "Weak hypotheses and boosting for generic object detection and recognition," in *Lecture Notes in Computer Science* vol. 3022, ed, 2004, pp. 71-84.

[40] A. Torralba, K. P. Murphy, and W. T. Freeman, "Sharing features: Efficient boosting procedures for multiclass object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004, pp. 762-769.

[41] A. Bar-Hillel, T. Hertz, and D. Weinshall, "Object class recognition by boosting a part-based model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 702-709.

[42] E. Bart and S. Ullman, "Cross-generalization: Learning novel classes from a single example by feature replacement," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 672-679.

[43] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, "Learning object categories from Google's image search," in *Proceedings of the IEEE International Conference on Computer Vision*, 2005, pp. 1816-1823.

[44] F. Jurie and B. Triggs, "Creating efficient codebooks for visual recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2005, pp. 604-610.

[45] Z. Tu, "Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2005, pp. 1589-1596.

[46] W. Zhang, B. Yu, G. J. Zelinsky, and D. Samaras, "Object class recognition using multiple layer boosting with heterogeneous features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 323-330.

[47] P. Dollar, Z. Tu, and S. Belongie, "Supervised learning of edges and object boundaries," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 1964-1971.

[48] A. Opelt, A. Pinz, and A. Zisserman, "Incremental learning of object detectors using a visual shape alphabet," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 3-10.

[49] D. D. Le and S. Satoh, "Ent-Boost: Boosting using entropy measures for robust object detection," *Pattern Recognition Letters,* vol. 28, pp. 1083-1090, 2007.

[50] A. Bar-Hillel and D. Weinshall, "Efficient learning of relational object class models," *International Journal of Computer Vision,* vol. 77, pp. 175-198, 2008.

[51] P. Carbonetto, G. Dorko', C. Schmid, H. Kuck, and N. De Freitas, "Learning to recognize objects with little supervision," *International Journal of Computer Vision,* vol. 77, pp. 219-237, 2008.

[52] L. Fu□rst, S. Fidler, and A. Leonardis, "Selecting features for object detection using an AdaBoost-compatible evaluation function," *Pattern Recognition Letters,* vol. 29, pp. 1603-1612, 2008.

[53] X. Li, B. Yang, F. Zhu, and A. Men, "Real-time object detection based on the improved boosted features," in *Proceedings of SPIE - The International Society for Optical Engineering*, 2009.

[54] J. J. Yokono and T. Poggio, "Object recognition using boosted oriented filter based local descriptors," *IEEJ Transactions on Electronics, Information and Systems,* vol. 129, 2009.

[55] D. K. Prasad and M. K. H. Leung, "Reliability/Precision Uncertainty in Shape Fitting Problems," in *IEEE International Conference on Image Processing*, Hong Kong, 2010, pp. 4277-4280.

[56] D. K. Prasad and M. K. H. Leung, "Polygonal representation of digital curves," in *Digital Image Processing*, S. G. Stanciu, Ed., ed: InTech, 2012, pp. 71-90.

[57] D. K. Prasad, M. K. H. Leung, C. Quek, and S.-Y. Cho, "A novel framework for making dominant point detection methods non-parametric," *Image and Vision Computing,* vol. 30, pp. 843-859, 2012.

[58] D. K. Prasad, C. Quek, and M. K. H. Leung, "A non-heuristic dominant point detection based on suppression of break points," in *Image Analysis and Recognition*. vol. 7324, A. Campilho and M. Kamel, Eds., ed Aveiro, Portugal: Springer Berlin Heidelberg, 2012, pp. 269-276.

[59] D. K. Prasad, C. Quek, M. K. H. Leung, and S. Y. Cho, "A parameter independent line fitting method," in *Asian Conference on Pattern Recognition (ACPR)*, Beijing, China, 2011, pp. 441-445.

[60] D. K. Prasad, "Assessing error bound for dominant point detection," *International Journal of Image Processing (IJIP),* vol. 6, pp. 326-333, 2012.

[61] Y. Chi and M. K. H. Leung, "Part-based object retrieval in cluttered environment," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 29, pp. 890-895, May 2007.

[62] D. K. Prasad and M. K. H. Leung, "An ellipse detection method for real images," in *25th International Conference of Image and Vision Computing New Zealand (IVCNZ 2010)*, Queenstown, New Zealand, 2010, pp. 1-8.

[63] D. K. Prasad, M. K. H. Leung, and S. Y. Cho, "Edge curvature and convexity based ellipse detection method," *Pattern Recognition,* vol. 45, pp. 3204-3221, 2012.

[64] D. K. Prasad, M. K. H. Leung, and C. Quek, "ElliFit: An unconstrained, non-iterative, least squares based geometric Ellipse Fitting method," *Pattern Recognition,* 2013.

[65] D. K. Prasad, C. Quek, and M. K. H. Leung, "A precise ellipse fitting method for noisy data," in *Image Analysis and Recognition*. vol. 7324, A. Campilho and M. Kamel, Eds., ed Aveiro, Portugal: Springer Berlin Heidelberg, 2012, pp. 253-260.

[66] D. K. Prasad and M. K. H. Leung, "Error analysis of geometric ellipse detection methods due to quantization," in *Fourth Pacific-Rim Symposium on Image and Video Technology (PSIVT 2010)*, Singapore, 2010, pp. 58 - 63.

[67] D. K. Prasad, R. K. Gupta, and M. K. H. Leung, "An Error Bounded Tangent Estimator for Digitized Elliptic Curves," in *Discrete Geometry for Computer Imagery*. vol. 6607, ed: Springer Berlin / Heidelberg, 2011, pp. 272-283.

[68] C. F. Olson, "A general method for geometric feature matching and model extraction," *International Journal of Computer Vision,* vol. 45, pp. 39-54, 2001.

[69] H. P. Moravec, "Rover visual obstacle avoidance," in *Proceedings of the International Joint Conference on Artificial Intelligence*, Vancouver, CANADA, 1981, pp. 785-790.

[70] C. Harris and M. Stephens, "A combined corner and edge detector," presented at the Alvey Vision Conference, 1988.

[71] J. Li and N. M. Allinson, "A comprehensive review of current local features for computer vision," *Neurocomputing,* vol. 71, pp. 1771-1787, 2008.

[72] K. Mikolajczyk and H. Uemura, "Action recognition with motion-appearance vocabulary forest," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1-8.

[73] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003, pp. 1615-1630.

[74] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision,* vol. 60, pp. 91-110, 2004.

[75] B. Ommer and J. Buhmann, "Learning the Compositional Nature of Visual Object Categories for Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2010.

[76] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *International Journal of Computer Vision,* vol. 77, pp. 259-289, 2008.

[77] M. Varma and A. Zisserman, "A statistical approach to material classification using image patch exemplars," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 31, pp. 2032-2047, 2009.

[78] P. M. Roth, S. Sternig, H. Grabner, and H. Bischof, "Classifier grids for robust adaptive object detection," in *Proceedings of the IEEE Computer Vision and Pattern Recognition*, Miami, FL, 2009, pp. 2727-2734.

[79] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing,* vol. 22, pp. 761-767, 2004.

[80] G. Carneiro and A. D. Jepson, "The quantitative characterization of the distinctiveness and robustness of local image descriptors," *Image and Vision Computing,* vol. 27, pp. 1143-1156, 2009.

[81] W. T. Lee and H. T. Chen, "Histogram-based interest point detectors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1590-1596.

[82] H. T. Comer and B. A. Draper, "Interest Point Stability Prediction," in *Proceedings of the International Conference on Computer Vision Systems*, Liege, 2009.

[83] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004, pp. 506-513.

[84] H. Zhang, W. Gao, X. Chen, and D. Zhao, "Object detection using spatial histogram features," *Image and Vision Computing,* vol. 24, pp. 327-341, 2006.

[85] R. Sandler and M. Lindenbaum, "Optimizing gabor filter design for texture edge detection and classification," *International Journal of Computer Vision,* vol. 84, pp. 308-324, 2009.

[86] H. Bischof, H. Wildenauer, and A. Leonardis, "Illumination insensitive recognition using eigenspaces," *Computer Vision and Image Understanding,* vol. 95, pp. 86-104, 2004.

[87] C. H. Lampert and J. Peters, "Active structured learning for high-speed object detection," in *Lecture Notes in Computer Science* vol. 5748, ed, 2009, pp. 221-231.

[88] C. Wallraven, B. Caputo, and A. Graf, "Recognition with local features: The kernel recipe," in *Proceedings of the IEEE International Conference on Computer Vision*, 2003, pp. 257-264.

[89] A. Zalesny, V. Ferrari, G. Caenen, and L. Van Gool, "Composite texture synthesis," *International Journal of Computer Vision,* vol. 62, pp. 161-176, 2005.

[90] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "TextonBoost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *International Journal of Computer Vision,* vol. 81, pp. 2-23, 2009.

[91] M. V. Rohith, G. Somanath, D. Metaxas, and C. Kambhamettu, "D - Clutter: Building object model library from unsupervised segmentation of cluttered scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2783-2789.

[92] C. Gu, J. J. Lim, P. Arbeláez, and J. Malik, "Recognition using regions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1030-1037.

[93] A. Bosch, A. Zisserman, and X. Muñoz, "Scene classification using a hybrid generative/discriminative approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 30, pp. 712-727, 2008.

[94] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International Journal of Computer Vision,* vol. 61, pp. 55-79, 2005.

[95] H. Wang and J. Oliensis, "Rigid shape matching by segmentation averaging," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 32, pp. 619-635, 2010.

[96] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 27, pp. 1615-1630, 2005.

[97] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: A survey," *Foundations and Trends in Computer Graphics and Vision,* vol. 3, pp. 177-280, 2007.

[98] L. Juan and O. Gwun, "A Comparison of SIFT, PCA-SIFT and SURF," *International Journal of Image Processing (IJIP),* vol. 3, pp. 143-152, 2009.

[99] N. Adluru and L. J. Latecki, "Contour grouping based on contour-skeleton duality," *International Journal of Computer Vision,* vol. 83, pp. 12-29, 2009.

[100] D. Cailliere, F. Denis, D. Pele, and A. Baskurt, "3D mirror symmetry detection using Hough transform," in *Proceedings of the IEEE International Conference on Image Processing*, San Diego, CA, 2008, pp. 1772-1775.

[101] B. Leibe and B. Schiele, "Analyzing appearance and contour based methods for object categorization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003, pp. 409-415.

[102] P. Schnitzspan, M. Fritz, S. Roth, and B. Schiele, "Discriminative structure learning of hierarchical representations for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2238-2245.

[103] V. Ferrari, T. Tuytelaars, and L. Van Gool, "Simultaneous object recognition and segmentation by image exploration," in *Lecture Notes in Computer Science* vol. 3021, ed, 2004, pp. 40-54.

[104] V. Ferrari, T. Tuytelaars, and L. Van Gool, "Simultaneous object recognition and segmentation from single or multiple model views," *International Journal of Computer Vision,* vol. 67, pp. 159-188, 2006.

[105] A. R. Pope and D. G. Lowe, "Probabilistic models of appearance for 3-D object recognition," *International Journal of Computer Vision,* vol. 40, pp. 149-167, 2000.

[106] J. A. Lasserre, C. M. Bishop, and T. P. Minka, "Principled hybrids of generative and discriminative models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 87-94.

[107] A. E. C. Pece, "On the computational rationale for generative models," *Computer Vision and Image Understanding,* vol. 106, pp. 130-143, 2007.

[108] M. Andriluka, S. Roth, and B. Schiele, "Discriminative Appearance Models for Pictorial Structures," *International Journal of Computer Vision,* vol. 99, pp. 259-280, Sep 2012.

[109] C. Desai, D. Ramanan, and C. C. Fowlkes, "Discriminative Models for Multi-Class Object Layout," *International Journal of Computer Vision,* vol. 95, pp. 1-12, Oct 2011.

[110] Y. Aytar and A. Zisserman, "Tabula Rasa: Model Transfer for Object Category Detection," in *IEEE International Conference on Computer Vision*, 2011, pp. 2252-2259.

[111] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," *Computer Vision and Image Understanding,* vol. 106, pp. 59-70, 2007.

[112] I. Ulusoy and C. M. Bishop, "Generative versus discriminative methods for object recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 258-265.

[113] G. Bouchard and B. Triggs, "Hierarchical part-based visual object categorization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 710-715.

[114] T. M. Mitchell, *Machine Learning*: Mcgraw-Hill International Edition, 2010.

[115] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *Proceedings of the IEEE Computer Vision and Pattern Recognition Workshop*, 2009, pp. 951-958.

[116] T. Yeh, J. J. Lee, and T. Darrell, "Fast concurrent object localization and recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 280-287.

[117] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, "Multi-class active learning for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2372-2379.

[118] S. Maji and J. Malik, "Object detection using a max-margin hough transform," in *Proceedings of the IEEE Computer Vision and Pattern Recognition*, Miami, FL, 2009, pp. 1038-1045.

[119] L. Wu, Y. Hu, M. Li, N. Yu, and X. S. Hua, "Scale-invariant visual language modeling for object categorization," *IEEE Transactions on Multimedia,* vol. 11, pp. 286-294, 2009.

[120] P. Jain and A. Kapoor, "Active learning for large Multi-class problems," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 762-769.

[121] A. Stefan, V. Athitsos, Q. Yuan, and S. Sclaroff, "Reducing JointBoost-Based Multiclass Classification to Proximity Search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 589-596.

[122] D. Parikh, C. L. Zitnick, and T. Chen, "Unsupervised learning of hierarchical spatial structures in images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2743-2750.

[123] S. Chen, J. Q. Wang, Y. Ouyang, B. Wang, C. S. Xu, and H. Q. Lu, "Boosting part-sense multi-feature learners toward effective object detection," *Computer Vision and Image Understanding,* vol. 115, pp. 364-374, Mar 2011.

[124] C. Dubout and F. Fleuret, "Tasting Families of Features for Image Classification," in *IEEE International Conference on Computer Vision*, 2011, pp. 929-936.

[125] P. Felzenszwalb, "Object Detection Grammars," in *IEEE International Conference on Computer Vision Workshops*, 2011.

[126] G. X. Huang, H. F. Chen, Z. L. Zhou, F. Yin, and K. Guo, "Two-class support vector data description," *Pattern Recognition,* vol. 44, pp. 320-329, Feb 2011.

[127] M. Jamieson, Y. Eskin, A. Fazly, S. Stevenson, and S. J. Dickinson, "Discovering hierarchical object models from captioned images," *Computer Vision and Image Understanding,* vol. 116, pp. 842-853, Jul 2012.

[128] L. Jie, T. Tommasi, and B. Caputo, "Multiclass Transfer Learning from Unconstrained Priors," in *IEEE International Conference on Computer Vision*, 2011, pp. 1863-1870.

[129] T. Y. Ma and L. J. Latecki, "From Partial Shape Matching through Local Deformation to Robust Global Shape Similarity for Object Detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1441-1448.

[130] M. Maire, S. X. Yu, and P. Perona, "Object Detection and Segmentation from Joint Embedding of Parts and Pixels," in *IEEE International Conference on Computer Vision*, 2011, pp. 2142-2149.

[131] S. Nilufar, N. Ray, and H. Zhang, "Object Detection With DoG Scale-Space: A Multiple Kernel Learning Approach," *IEEE Transactions on Image Processing,* vol. 21, pp. 3744-3756, Aug 2012.

[132] E. Rahtu, J. Kannala, and M. Blaschko, "Learning a Category Independent Object Detection Cascade," in *IEEE International Conference on Computer Vision*, 2011, pp. 1052-1059.

[133] N. Razavi, J. Gall, and L. Van Gool, "Scalable Multi-class Object Detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1505-1512.

[134] S. Rivera and A. M. Martinez, "Learning deformable shape manifolds," *Pattern Recognition,* vol. 45, pp. 1792-1801, Apr 2012.

[135] R. Salakhutdinov, A. Torralba, and J. Tenenbaum, "Learning to Share Visual Appearance for Multiclass Object Detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1481-1488.

[136] J. Stottinger, A. Hanbury, N. Sebe, and T. Gevers, "Sparse Color Interest Points for Image Retrieval and Object Categorization," *IEEE Transactions on Image Processing,* vol. 21, pp. 2681-2692, May 2012.

[137] G. Tolias and Y. Avrithis, "Speeded-up, relaxed spatial matching," in *IEEE International Conference on Computer Vision*, 2011, pp. 1653-1660.

[138] K. Tzevanidis and A. Argyros, "Unsupervised learning of background modeling parameters in multicamera systems," *Computer Vision and Image Understanding,* vol. 115, pp. 105-116, Jan 2011.

[139] M. Villamizar, J. Andrade-Cetto, A. Sanfeliu, and F. Moreno-Noguer, "Bootstrapping Boosted Random Ferns for discriminative and efficient object classification," *Pattern Recognition,* vol. 45, pp. 3141-3153, Sep 2012.

[140] X. G. Wang, X. Bai, W. Y. Liu, and L. J. Latecki, "Feature Context for Image Classification and Object Detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 961-968.

[141] A. Zaharescu, E. Boyer, and R. Horaud, "Keypoints and Local Descriptors of Scalar Functions on 2D Manifolds," *International Journal of Computer Vision,* vol. 100, pp. 78-98, Oct 2012.

[142] M. Fritz, B. Leibe, B. Caputo, and B. Schiele, "Integrating representative and discriminant models for object category detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2005, pp. 1363-1370.

[143] Y. Li, L. Gu, and T. Kanade, "A robust shape model for multi-view car alignment," in *Proceedings of the IEEE Computer Vision and Pattern Recognition Workshop*, 2009, pp. 2466-2473.

[144] D. K. Prasad, "Adaptive traffic signal control system with cloud computing based online learning," in *8th International Conference on Information, Communications, and Signal Processing (ICICS 2011)*, Singapore, 2011.

[145] D. K. Prasad, "High Availability based Migration Analysis to Cloud Computing for High Growth Businesses," *International Journal of Computer Networks (IJCN),* vol. 4, 2012.

[146] M. F. Demirci, A. Shokoufandeh, and S. J. Dickinson, "Skeletal shape abstraction from examples," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 31, pp. 944-952, 2009.

[147] M. Bergtholdt, J. Kappes, S. Schmidt, and C. Schnörr, "A study of parts-based object class detection using complete graphs," *International Journal of Computer Vision,* vol. 87, pp. 93-117, 2010.

[148] J. R. R. Uijlings, A. W. M. Smeulders, and R. J. H. Scha, "What is the spatial extent of an object?," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 770-777.

[149] F. Perronnin, "Universal and adapted vocabularies for generic visual categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 30, pp. 1243-1256, 2008.

[150] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2169-2178.

[151] D. A. Ross and R. S. Zemel, "Learning parts-based representations of data," *Journal of Machine Learning Research,* vol. 7, pp. 2369-2397, 2006.

[152] K. Y. Chang, T. L. Liu, H. T. Chen, and S. H. Lai, "Fusing Generic Objectness and Visual Saliency for Salient Object Detection," in *IEEE International Conference on Computer Vision*, 2011, pp. 914-921.

[153] S. Lazebnik and M. Raginsky, "Supervised learning of quantizer codebooks by information loss minimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 31, pp. 1294-1309, 2009.

[154] B. Epshtein and S. Ullman, "Feature hierarchies for object classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2005, pp. 220-227.

[155] J. Gall and V. Lempitsky, "Class-specific hough forests for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2009, pp. 1022-1029.

[156] S. Basalamah, A. Bharath, and D. McRobbie, "Contrast marginalised gradient template matching," in *Lecture Notes in Computer Science* vol. 3023, ed, 2004, pp. 417-429.

[157] S. Belongie, J. Malik, and J. Puzicha, "Matching shapes," in *Proceedings of the IEEE International Conference on Computer Vision*, 2001, pp. 454-461.

[158] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, pp. 509-522, 2002.

[159] A. C. Berg, T. L. Berg, and J. Malik, "Shape matching and object recognition using low distortion correspondences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 26-33.

[160] S. Biswas, G. Aggarwal, and R. Chellappa, "Robust estimation of albedo for illumination-invariant matching and shape recovery," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 31, pp. 884-899, 2009.

[161] G. Borgefors, "Hierarchical Chamfer matching: A parametric edge matching algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 10, pp. 849-865, 1988.

[162] A. M. Bronstein, M. M. Bronstein, A. M. Bruckstein, and R. Kimmel, "Partial similarity of objects, or how to compare a centaur to a horse," *International Journal of Computer Vision,* vol. 84, pp. 163-183, 2009.

[163] R. Brunelli and T. Poggio, "Template matching: Matched spatial filters and beyond," *Pattern Recognition,* vol. 30, pp. 751-768, 1997.

[164] G. J. Burghouts and J. M. Geusebroek, "Performance evaluation of local colour invariants," *Computer Vision and Image Understanding,* vol. 113, pp. 48-62, 2009.

[165] J. R. Burrill, S. X. Wang, A. Barrow, M. Friedman, and M. Soffen, "Model-based matching using elliptical features," in *Proceedings of SPIE - The International Society for Optical Engineering*, 1996, pp. 87-97.

[166] M. Ceccarelli and A. Petrosino, "The orientation matching approach to circular object detection," in *Proceedings of the IEEE International Conference on Image Processing*, 2001, pp. 712-715.

[167] S. H. Chang, F. H. Cheng, W. H. Hsu, and G. Z. Wu, "Fast algorithm for point pattern matching: Invariant to translations, rotations and scale changes," *Pattern Recognition,* vol. 30, pp. 311-320, 1997.

[168] F. H. Cheng, "Multi-stroke relaxation matching method for handwritten Chinese character recognition," *Pattern Recognition,* vol. 31, pp. 401-410, 1998.

[169] Y. Chi and M. K. H. Leung, "A local structure matching approach for large image database retrieval," in *Proceedings of the International Conference on Image Analysis and Recognition*, Oporto, PORTUGAL, 2004, pp. 761-768.

[170] T. H. Cho, "Object matching using generalized hough transform and chamfer matching," in *Lecture Notes in Computer Science* vol. 4099, ed, 2006, pp. 1253-1257.

[171] O. Choi and I. S. Kweon, "Robust feature point matching by preserving local geometric consistency," *Computer Vision and Image Understanding,* vol. 113, pp. 726-742, 2009.

[172] P. F. Felzenszwalb, "Representation and detection of deformable shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 27, pp. 208-220, 2005.

[173] P. F. Felzenszwalb and J. D. Schwartz, "Hierarchical matching of deformable shapes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1-8.

[174] Y. S. Gao and M. K. H. Leung, "Line segment Hausdorff distance on face matching," *Pattern Recognition,* vol. 35, pp. 361-371, Feb 2002.

[175] H. Hakalahti, D. Harwood, and L. S. Davis, "Two-dimensional object recognition by matching local properties of contour points," *Pattern Recognition Letters,* vol. 2, pp. 227-234, 1984.

[176] F. Li, M. K. H. Leung, and X. Z. Yu, "A two-level matching scheme for speedy and accurate palmprint identification," in *Proceedings of the International Multimedia Modeling Conference*, Singapore, SINGAPORE, 2007, pp. 323-332.

[177] X. Lin, Z. Zhu, and W. Deng, "Stereo matching algorithm based on shape similarity for indoor environment model building," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1996, pp. 765-770.

[178] Z. Lin and L. S. Davis, "Shape-based human detection and segmentation via hierarchical part-template matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 32, pp. 604-618, 2010.

[179] H.-C. Liu and M. D. Srinath, "Partial shape classification using contour matching in distance transformation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 12, pp. 1072-1079, 1990.

[180] G. Mori, S. Belongie, and J. Malik, "Efficient shape matching using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 27, pp. 1832-1837, 2005.

[181] C. F. Olson and D. P. Huttenlocher, "Automatic target recognition by matching oriented edge pixels," *IEEE Transactions on Image Processing,* vol. 6, pp. 103-113, 1997.

[182] F. C. D. Tsai, "Robust affine invariant matching with application to line features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1993, pp. 393-399.

[183] C. Xu, J. Liu, and X. Tang, "2D shape matching by contour flexibility," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 31, pp. 180-186, 2009.

[184] X. Z. Yu and M. K. H. Leung, "Shape recognition using curve segment Hausdorff distance," in *Proceedings of the International Conference on Pattern Recognition*, Hong Kong, PEOPLES R CHINA, 2006, pp. 441-444.

[185] X. Z. Yu, M. K. H. Leung, and Y. S. Gao, "Hausdorff distance for shape matching," in *Proceedings of the IASTED International Conference on Visualization, Imaging, and Image Processing*, Marbella, SPAIN, 2004, pp. 819-824.

[186] F. Li and M. K. H. Leung, "Two-stage approach for palmprint identification using Hough transform and Hausdorff distance," in *Proceedings of the International Conference on Control, Automation, Robotics and Vision*, Singapore, SINGAPORE, 2006, pp. 1302-1307.

[187] D. G. Sim and R. H. Park, "Two-dimensional object alignment based on the robust oriented Hausdorff similarity measure," *IEEE Transactions on Image Processing,* vol. 10, pp. 475-483, 2001.

[188] S. Lazebnik, C. Schmid, and J. Ponce, "A maximum entropy framework for part-based texture and object recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2005, pp. 832-838.

[189] M. Amiri and H. R. Rabiee, "RASIM: A Novel Rotation and Scale Invariant Matching of Local Image Interest Points," *IEEE Transactions on Image Processing,* vol. 20, pp. 3580-3591, Dec 2011.

[190] D. Gerónimo, A. D. Sappa, D. Ponsa, and A. M. López, "2D-3D-based on-board pedestrian detection system," *Computer Vision and Image Understanding,* p. In press, 2010.

[191] P. Wang and Q. Ji, "Multi-view face and eye detection using discriminant features," *Computer Vision and Image Understanding,* vol. 105, pp. 99-111, 2007.

[192] S. Avidan, "Ensemble tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 29, pp. 261-271, 2007.

[193] M. Enzweiler and D. M. Gavrila, "Monocular pedestrian detection: Survey and experiments," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 31, pp. 2179-2195, 2009.

[194] Z. He, T. Tan, Z. Sun, and X. Qiu, "Toward accurate and fast iris segmentation for iris biometrics," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 31, pp. 1670-1684, 2009.

[195] C. Huang, H. Ai, Y. Li, and S. Lao, "High-performance rotation invariant multiview face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 29, pp. 671-686, 2007.

[196] J. J. LaViola Jr and R. C. Zeleznik, "A practical approach for writer-dependent symbol recognition using a writer-independent symbol recognizer," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 29, pp. 1917-1926, 2007.

[197] S. Z. Li and Z. Q. Zhang, "FloatBoost learning and statistical face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 26, pp. 1112-1123, 2004.

[198] E. Makinen and R. Raisamo, "Evaluation of gender classification methods with automatically detected and aligned faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 30, pp. 541-547, 2008.

[199] J. J. Rodríguez, L. I. Kuncheva, and C. J. Alonso, "Rotation forest: A New classifier ensemble method," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 28, pp. 1619-1630, 2006.

[200] J. Wu, S. C. Brubaker, M. D. Mullin, and J. M. Rehg, "Fast asymmetric learning for cascade face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 30, pp. 369-382, 2008.

[201] S. Baluja and H. A. Rowley, "Boosting sex identification performance," *International Journal of Computer Vision,* vol. 71, pp. 111-119, 2007.

[202] J. H. Elder, S. J. D. Prince, Y. Hou, M. Sizintsev, and E. Olevskiy, "Pre-attentive and attentive detection of humans in wide-field scenes," *International Journal of Computer Vision,* vol. 72, pp. 47-66, 2007.

[203] Y. Liu, X. L. Wang, H. Y. Wang, H. Zha, and H. Qin, "Learning Robust Similarity Measures for 3D Partial Shape Retrieval," *International Journal of Computer Vision,* pp. 1-24, 2009.

[204] J. Porway, Q. Wang, and S. C. Zhu, "A Hierarchical and Contextual Model for Aerial Image Parsing," *International Journal of Computer Vision,* pp. 1-30, 2009.

[205] H. Schneiderman and T. Kanade, "Object detection using the statistics of parts," *International Journal of Computer Vision,* vol. 56, pp. 151-177, 2004.

[206] J. Šochman and J. Matas, "Learning fast emulators of binary decision processes," *International Journal of Computer Vision,* vol. 83, pp. 149-163, 2009.

[207] K. Tieu and P. Viola, "Boosting Image Retrieval," *International Journal of Computer Vision,* vol. 56, pp. 17-36, 2004.

[208] Z. Tu, X. Chen, A. L. Yuille, and S. C. Zhu, "Image parsing: Unifying segmentation, detection, and recognition," *International Journal of Computer Vision,* vol. 63, pp. 113-140, 2005.

[209] P. Viola and M. J. Jones, "Robust Real-Time Face Detection," *International Journal of Computer Vision,* vol. 57, pp. 137-154, 2004.

[210] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *International Journal of Computer Vision,* vol. 63, pp. 153-161, 2005.

[211] B. Wu and R. Nevatia, "Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors," *International Journal of Computer Vision,* vol. 75, pp. 247-266, 2007.

[212] P. J. Bickel, Y. Ritov, and A. Zakai, "Some theory for generalized boosting algorithms," *Journal of Machine Learning Research,* vol. 7, pp. 705-732, 2006.

[213] M. Culp, K. Johnson, and G. Michailidis, "Ada: An R package for stochastic boosting," *Journal of Statistical Software,* vol. 17, pp. 1-27, 2006.

[214] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning,* vol. 37, pp. 297-336, 1999.

[215] Y. Freund, "An adaptive version of the boost by majority algorithm," *Machine Learning,* vol. 43, pp. 293-318, 2001.

[216] M. Collins, R. E. Schapire, and Y. Singer, "Logistic regression, AdaBoost and Bregman distances," *Machine Learning,* vol. 48, pp. 253-285, 2002.

[217] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics,* vol. 29, pp. 1189-1232, 2001.

[218] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics and Data Analysis,* vol. 38, pp. 367-378, 2002.

[219] P. K. Mallapragada, R. Jin, A. K. Jain, and Y. Liu, "SemiBoost: Boosting for semi-supervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 31, pp. 2000-2014, 2009.

[220] A. Torralba, K. P. Murphy, and W. T. Freeman, "Sharing visual features for multiclass and multiview object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 29, pp. 854-869, 2007.