Junwei Han, Dingwen Zhang,
Gong Cheng, Nian Liu, and Dong Xu

Advanced Deep-Learning Techniques for Salient and Category-Specific Object Detection

A survey

©ISTOCKPHOTO.COM/ZAPP2PHOTO

Object detection, including objectness detection (OD), salient object detection (SOD), and category-specific object detection (COD), is one of the most fundamental yet challenging problems in the computer vision community. Over the last several decades, great efforts have been made by researchers to tackle this problem, due to its broad range of applications for other computer vision tasks such as activity or event recognition, content-based image retrieval and scene understanding, etc. While numerous methods have been presented in recent years, a comprehensive review for the proposed high-quality object detection techniques, especially for those based on advanced deep-learning techniques, is still lacking. To this end, this article delves into the recent progress in this research field, including 1) definitions, motivations, and tasks of each subdirection; 2) modern techniques and essential research trends; 3) benchmark data sets and evaluation metrics; and 4) comparisons and analysis of the experimental results. More importantly, we will reveal the underlying relationship among OD, SOD, and COD and discuss in detail some open questions as well as point out several unsolved challenges and promising future works.

Introduction

As a challenging but useful computer vision task, object detection aims at identifying the presence of various individual objects in each given image or video. In this research field, promising results have been achieved when dealing with images with relatively simple image scenes and clear foreground objects. However, the problem is not adequately addressed when dealing with the images and videos containing objects placed in arbitrary poses, with various shapes, and appearing in a cluttered and occluded environment.

The research works in object detection published in the past few decades can be roughly categorized into three directions: OD, SOD, and COD. Specifically, OD [1], [2] aims at detecting all possible objects appearing in each given image, regardless of the specific object category. It has great challenges, as different objects, either within the same object category or from different object categories, can have dramatic appearance variation, due

Digital Object Identifier 10.1109/MSP.2017.2749125
Date of publication: 9 January 2018

1053-5888/18©2018IEEE

to their internal intrinsic characteristics (e.g., living creatures such as cats generally have more deformable appearances than man-made objects such as vehicles) or external capturing conditions such as viewing distances or angles (e.g., deformable objects may appear somewhat rigid at a distance, and even rigid objects may exhibit variations under different viewing angles). Usually, OD algorithms output thousands of object proposals or hypotheses as shown in Figure 1(a), which can benefit a wide range of computer vision tasks like weakly supervised learning [3] and object tracking [4].

SOD [5], [6] is another direction in object detection, which aims at mimicking the visual attention mechanism to highlight objects that draw our attention from each given image [91]. This is inspired by the human visual attention system, which can guide humans to pay special attention to a few informative image regions that are naturally distinct (the bottom-up saliency) or related to certain objects categories determined by cognitive phenomena like knowledge, expectations, reward, and specific tasks (the top-down saliency) [7]. Similar to OD, bottom-up SOD suffers from the challenges that come from large appearance variation in unconstrained object categories, whereas top-down SOD faces the challenge of how to effectively associate the desired visual stimulus (usually at a semantic level) and the corresponding regions in visual scenes. Usually, SOD algorithms output a limited number of object regions based on the obtained saliency maps, as shown in Figure 1(b). They can also benefit a wide range of computer vision tasks like image retrieval [8] and object segmentation [9].

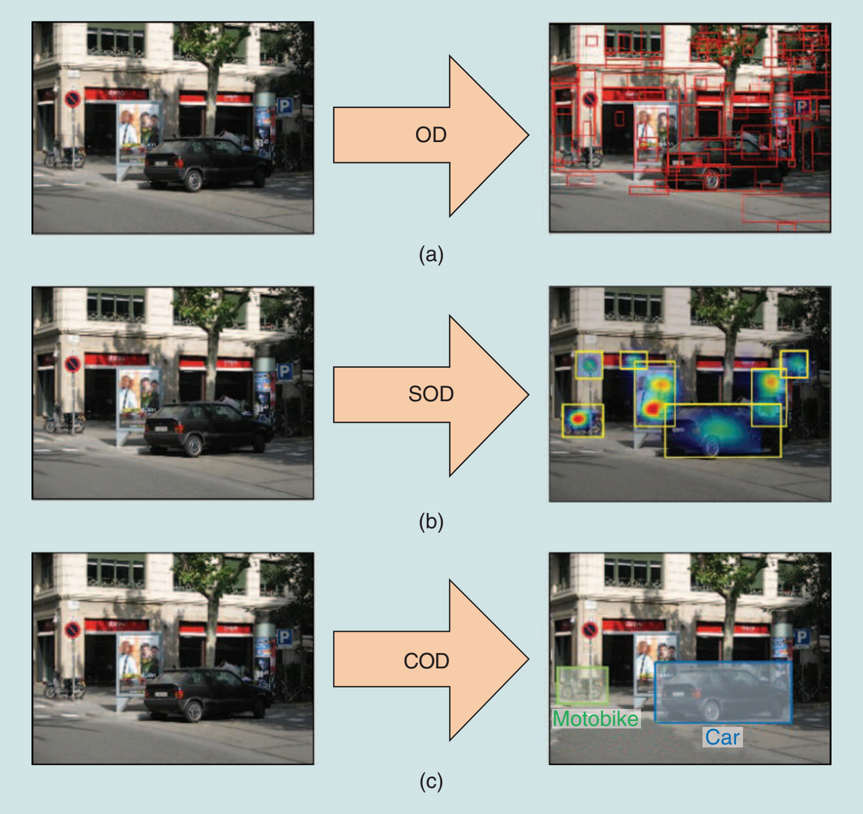

FIGURE 1. The three research directions in object detection: (a) OD, (b) SOD, and (c) COD.

The third direction of object detection is COD [10], [11]. Different from OD, COD aims at detecting multiple predefined object categories from each given image. It needs to not only identify the image regions that may contain the objects of interest but also recognize the specific object category of each detected image region. Compared with SOD, COD has a totally different motivation, i.e., it moves toward solving a pure computational problem without any understanding of the function in the human visual system, e.g., visual attention. Usually, COD is converted to a multiclass classification problem, where discriminative classification functions are trained to separate the extracted image regions in the corresponding feature domain. The main challenge in COD is how to deal with the intraclass appearance variation and interclass appearance similarity. As shown in Figure 1(c), COD approaches usually output multiple image regions assigned with the identified object category. COD can be applied to computer vision tasks like scene parsing [12] and human action recognition [13].

For addressing the challenging problems in object detection, intensive works have been proposed to design better handcrafted features (such as HOG and SIFT) and put forward sophisticated object detection frameworks to incorporate the extracted features with the carefully designed classifiers (e.g., random forest and AdaBoost) in the stage of the whole development of object detection. Convolutional neural networks (CNNs) were first applied to object detection in 2004 [14] and have been widely used since 2013 [15]. The work on region-based CNNs (RCNNs) in [10] led to significant breakthroughs in 2014. It made the earliest efforts to describe an object detection system by using multilayer convolutional networks to extract highly discriminative yet invariant feature representation.

This work has achieved a significant improvement of more than 50% mean average precision (mAP) when compared to the best methods at that time, which were based on the handcrafted image features on the commonly used PASCAL detection benchmark [16]. Since then, several advanced deep-learning-based techniques [17]–[20] have been proposed for high-quality object detection, which covers all three related areas of OD, SOD, and COD. To this end, this article presents a comprehensive survey of recent state-of-the-art approaches.

This article mainly has four motivations:

1) Object detection, including OD, SOD, and COD, is a fundamental yet challenging problem of computer vision. The existing survey papers focus on each individual topic only, without discussing the close relationship.

2) Since many methods have been proposed and breakthrough performances have been achieved in recent years, it would be enlightening to review the recently proposed object detection techniques, especially for those based on deep-learning techniques.
3) It is of great interest to conduct in-depth discussions for a few important questions. For example, why can the recent deep-learning-based frameworks significantly boost the performance of object detection? What is the most intrinsic improvement of such frameworks when compared with previous ones? What are the problems that need to be solved in the future for deep-learning-based approaches?
4) Comprehensive comparison and analysis for the experimental results on publicly available object detection benchmarks would help readers to better understand the performance of each object detection strategy as well as the corresponding network architecture.

Preliminary knowledge

In recent years, there has been rapid development in the research area of deep learning, including its popularization in computer vision. In this section, we briefly introduce one of the advanced deep-learning techniques that has been widely used in the object detection task, i.e., the CNN.

The CNN is one of the most well-known and widely used deep-learning architectures inspired by the natural visual perception mechanism of living creatures, which was first proposed in 1980 by Fukushima [21] and then improved by LeCun [22]. CNNs are designed to process data that come in the form of multiple arrays [23], for example, a color image composed of three two-dimensional arrays that contain pixel intensities in the three color channels. There are four key ideas behind CNNs that take advantage of the properties of natural signals: local connections, shared weights, pooling, and the use of many layers [23].

As shown in Figure 2, the architecture of a typical CNN model is structured as a series of layers as follows:

- *Convolutional layers*: Convolutional layers are the most important for feature extraction. The first several layers usually capture low-level features (like edges, lines, and corners) while the deeper layers are able to learn high-level features (like structures, objects, and shapes) by combining low-level ones. Each unit in a convolutional layer is connected to a local patch in the feature maps of the previous layer through a set of kernels called a *filter bank*. The result of this local weighted sum is then passed through a nonlinearity operation such as a rectified linear unit (ReLU). All units in a feature map share the same filter bank. Different feature maps in a convolutional layer use different filter banks.
- *Pooling layers*: Pooling layers aim to reduce the dimension of the representation and create invariance to small shifts and distortions. A pooling layer is usually placed between two convolutional layers. Each feature map of a pooling layer is connected to its corresponding feature map of the previous convolutional layer. A typical pooling unit computes the maximum of a local patch of units in one feature map.
- *Fully connected layers*: Fully connected layers are typically used as the last few layers of the network for better summarizing the information conveyed by lower-level layers in view of the final decision. As a fully connected layer occupies most of the parameters, overfitting can easily happen. To prevent this, the dropout method is usually employed [24].

Starting with the breakthrough success of AlexNet [24] for ImageNet classification in 2012, significant efforts have been made in developing various CNN models, including VGGNet [25], GoogLeNet [26], and ResNet [27].

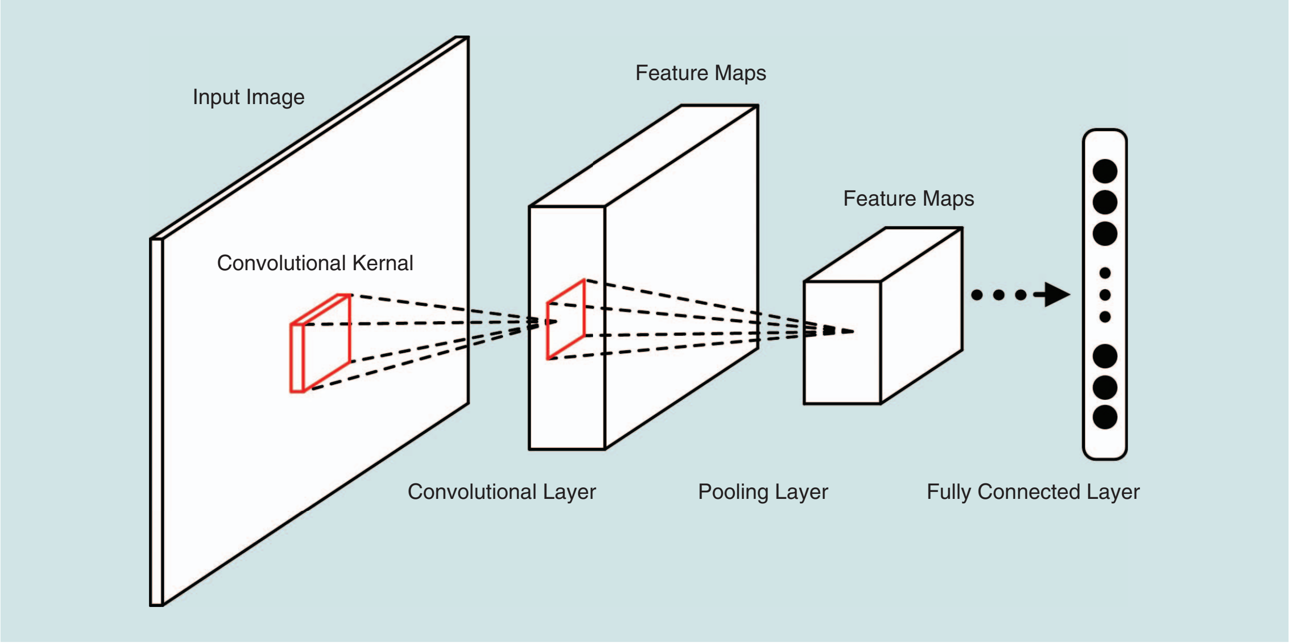

FIGURE 2. The architecture of a typical CNN model.

- *AlexNet*: AlexNet [24] was first proposed by Krizhevsky et al. and won the 2012 ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [28]. It is composed of five convolutional layers and three fully connected layers. It is a milestone study for computer vision and machine learning since it was the first work to employ nonsaturating neurons, graphics processing unit (GPU) implementation of the convolution operation, and dropout to prevent overfitting.
- *VGGNet*: VGGNet [25] was the winner of localization and classification tracks of the ILSVRC 2014 competition. It has two famous architectures: VGGNet-16 and VGGNet-19. The former has been widely used because of its simpler architecture, which has 13 convolutional layers, five pooling layers, and three fully connected layers.
- *GoogLeNet*: GoogLeNet [26] is another representative CNN architecture, which has two main advantages. One is the utilization of filter kernels of different sizes at the same layer, which preserves more spatial information, and the other advantage is the reduction of the number of parameters of the network, which makes it less sensitive to overfitting and allows it to be deeper. In fact, the 22-layer GoogLeNet has more than 50 convolutional layers distributed inside the inception modules, but it has 12 times fewer parameters than AlexNet.
- *ResNet*: ResNet [27] is one of the most successful CNNs and won the Conference on Computer Vision and Pattern Recognition 2016 Best Paper Award. The idea behind ResNet is that each layer should not learn the whole feature space transformation but only a residual correction to the previous layer, which allows training much deeper networks efficiently. Its extremely deep representations have excellent generalization performance and led it to win first place in ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation at the 2015 ILSVRC and COCO competitions.

Modern methods in object detection

Modern methods in OD

The goal of OD is to select a small set of object proposals that cover most of the objects of interest in the given images. To achieve this goal, OD approaches need to 1) generate or select potential bounding boxes that are likely to contain a certain object of interest and 2) infer the objectness scores of the selected bounding boxes. We can generally divide the existing OD approaches into three main categories: region merging, window selecting, and box regressing.

Region-merging approaches

Region-merging approaches attempt to generate object proposals by merging multiple local image regions, e.g., superpixels. One representative region-merging approach is the well-known selective search approach [29], which applied a greedy algorithm to iteratively group image regions together. Specifically, the similarities between all neighboring regions were first calculated. Afterward, the two most similar regions were grouped together, and new similarities were calculated between the resulting region and its neighbors. Such a grouping process was repeated until the whole image became a single region. Along this direction, additional recent approaches have been proposed to better address the problem. For example, Krahenbuhl et al. [30] developed a learning-based seed placement approach to identify a set of seed superpixels for hitting all objects in each given image. Then, they merged the image regions close to each seed superpixel and computed a signed geodesic distance transform to extract a small set of high-quality object proposals. Bazzani et al. [31] made an early effort to adopt the deep-learning technique and proposed a novel region-merging-based OD approach. Specifically, they first generated the initial set of rectangular regions, i.e., the bounding boxes enclosing the extracted segments. Then, they greedily merged the two regions by maximizing the similarity function containing four terms. The four terms are calculated via the CNNs pretrained on ImageNet [28], which include the similarity in classification score drop, the similarity in deep feature, the coverage area of the image, and the distance of the spatial location, respectively.

Window-selecting approaches

Window-selecting approaches attempt to generate object proposals by scoring and selecting pregenerated (sliding) windows. One of the earliest and most well-known window-selecting-based OD methods was proposed by [1]. It first obtained an initial set of proposals from the salient locations [32] in a given image. Then, these proposals were scored by combining multiple information cues including color contrast, edge density, location, size, and "superpixel straddling" cue. Inspired by this work, researchers have conducted extensive studies in the past few years. For example, Ghodrati et al. [17] made one of the earliest efforts to make use of the deep feature maps of a pretrained CNN. The main idea is to generate hypotheses in a sliding-window fashion over different activation layers. The proposed inverse cascade searching went from the final to the initial convolutional layers of the CNN, which can select the most promising object locations and refine their boxes in a coarse-to-fine manner. Similarly, Pinheiro et al. [33], [34] proposed using a CNN with two branches to simultaneously infer the segmentation mask and the objectness score of each input image window. The final bounding-box proposals were obtained by taking the rectangle image regions enclosing the segmentation masks. Different from [17] and [33], Kuo et al. [2] pursued a data-driven, semantic approach for ranking object proposals. They learned a small but effective CNN architecture to rerank proposals extracted by the bottom-up method [35]. In [36], Hu et al. extracted dense sliding windows from the convolutional feature maps and then adopted a unified head module to decode the window features and produce the output confidence score as well as object mask.

Box-regressing approaches

Box-regressing approaches attempt to generate object proposals by directly learning a regression function to obtain the bounding-box locations and objectness scores from the extracted deep feature maps. These approaches emerged with the success of deep-learning techniques in computer vision. Specifically, the first box-regressing-based OD approach was proposed by Erhan et al. [37] in 2014, where OD was defined as a regression problem to the coordinates of bounding-box locations. In addition, it inferred the confidence score of each bounding box, which indicates how likely it is that this box contains an object. The whole system was implemented by a single CNN model with a novel loss function that considered the location and scoring accuracy. Based on [37], Szegedy et al. [38] further built the framework based on the latest Inception-style architecture [39] and utilized multiscale convolutional predictors of bounding-box shape and confidence, leading to an Inception-based postclassification model. Along this direction, Ren et al. [40] proposed a region proposal network (RPN) to learn a regression function for fitting the coordinates of bounding-box locations based on a set of predefined translation-invariant anchors. They also combined the RPN with some additional network layers for precise COD. Similarly, Kong et al. [41] developed a novel hyperfeature to combine the deep but coarse information with the shallow but fine information to extract more abundant features. A zoom-out-and-in network for generating object proposals was established by Li et al. [42], where a zoom-in subnetwork was used to increase the resolution of high-level features via a deconvolution operation and a recursive training pipeline was designed to consecutively regress region proposals at the training stage.

Modern methods in SOD

SOD involves two branches: bottom-up and top-down. The former is stimulus driven and mainly responds to the most interesting and conspicuous regions in visual scenes, while the latter is directed by knowledge and high-level visual tasks, such as intentionally looking for a category-specific object. As mentioned by the previous studies [7], [43], and [44], in the branch of bottom-up SOD, approaches are to detect saliency under free viewing, which is automatically determined by the physical characteristics of the scene, while approaches in the other branch are to detect the task-driven saliency determined by the current goals of the observer. In each of the branches, both supervised and unsupervised frameworks can be established to address the corresponding problems. Next, we will examine these two branches in greater detail.

Bottom-up SOD

Bottom-up SOD aims at accurately discriminating foreground objects from backgrounds in visual scenes. Traditional models mainly rely on the contrast cues. One representative approach proposed by Cheng et al. [6] measured the weighted sum of the color difference between each image region and all other regions in the image in normalized color histograms as the global contrast to detect saliency. Inspired by this work, some researchers also integrated both local and global contrast for saliency detection. Recently, with the great progress in deep learning, deep neural networks (DNNs) have also been adopted to boost the performance of SOD. One of the earliest pioneering works is [45], where Han et al. proposed utilizing the stacked denoising autoencoder to model the background prior for SOD. Besides this work, a number of CNN-based SOD approaches have been proposed in more recent years. For example, Wang et al. [46] proposed integrating both local estimation and global search for saliency detection. Lee et al. [47] combined a low-level distance map of each superpixel and global CNN feature of the whole image. Liu and Han [20] proposed to hierarchically detect salient objects from coarse to fine with global-to-local contexts in an end-to-end fashion. Li and Yu [48] proposed combining a pixel-level fully convolutional network (FCN)-based saliency network with the segment-wise multiscale CNN for saliency detection. Wang et al. [49] proposed a progressive saliency refinement network via a recurrent FCN, in which the previous saliency map and the original image were simultaneously fed to learn to correct its previous errors, leading to better saliency results.

Top-down SOD

Top-down SOD usually aims at highlighting category-specific objects in the scene. Yang and Yang [50] proposed to jointly learn the parameters of conditional random fields and a dictionary for supervised top-down saliency detection. He et al. [51] proposed to detect exemplar-based top-down saliency with the goal of locating objects belonging to the same category with the given exemplar images. Cholakkal et al. [52] proposed a weakly supervised top-down saliency framework using only image labels. They first trained a sparse coding-based spatial-pyramid-matching (ScSPM) classifier using image labels. Then the probabilistic contribution of each patch in an image to the classifier was analyzed to estimate the reverse-ScSPM saliency. Next, contextual patches were utilized to estimate the contextual saliency by using a logistic regression model. The final saliency map can be obtained by combining the two saliency maps. Zhang et al. [53] proposed the excitation backpropagation method for top-down saliency detection based on the top-down winner-take-all process and the backpropagation in DNNs.

Modern methods in COD

COD has been widely studied in the literature for the last few decades. The deformable part model (DPM) [54] and its variants have been the leading methods for many years. These methods employ handcrafted image descriptors as features and scan through the entire image to detect regions with a class-specific maximum response.

More recently, due to the availability of large-scale training data such as ImageNet [28] and the advance of high-performance GPUs, various deep-learning-based methods (especially CNN based) have been proposed to significantly improve the state of the art of COD. In fact, the use of CNNs for detection and recognition can be traced back to the 1980s [22]. However, because of the lack of training data and limited computing resources, there was not much advancement for CNN-based COD before 2012. Since the groundbreaking success of CNNs in the image classification task at ILSVRC in 2012 [28], the CNN-based paradigm (for COD) has recently attracted great deal of research interest. There are generally two main categories of COD methods: object proposal based and regression based.

Object proposal-based approaches

An object proposal-based COD framework first generates a set of proposal bounding-boxes that possibly contain objects by using region proposal methods such as selective search [29] (this process is also known as OD) and then passes the detected object proposals to the CNN classifiers to determine whether they are backgrounds or from a specific object class.

Among various object proposal-based methods (for COD), the work of region-CNN (R-CNN) proposed by Girshick et al. [10] in 2014 is one of the most notable methods. This work opens the door to extracting rich features via deep CNN models, which significantly boosts the performance. The R-CNN framework is a chain of conceptually simple steps: generating object proposals, classifying proposals as background or category-specific objects, and postprocessing detections to improve their fitness to objects. Briefly, R-CNN works as follows. First, it extracts about 2,000 bottom-up region proposals that probably contain objects through the selective search algorithm [29] to reduce the computational cost. Then, these region proposals are warped to a fixed size (e.g., 227 × 227) and a fine-tuned CNN model is used to extract the CNN features from them. Next, category-specific linear support vector machines (SVMs) are used to classify each region proposal as *object* or *nonobject*. Finally, the candidate proposals are refitted to detected objects by using a bounding-box regressor [54] to refine localizations. This simple pipeline has achieved state-of-the-art COD performance on the benchmark data sets with significant performance improvement over all previously published works, which are mainly based on DPM [54]. Here, it is worth mentioning that the CNN models used for extracting deep CNN features from region proposals were usually pretrained on an auxiliary task of image classification based on the ImageNet data set [28] and then fine-tuned on a small set of images with bounding-box annotations for the detection task.

However, in R-CNN, we have to repeatedly resize candidate bounding boxes to a fixed size to extract their CNN features, which is computationally expensive for COD. To speed up R-CNN, some works [18], [55], [56] propose to share the computation in feature extraction. For example, the spatial pyramid pooling network (SPPnet) [55] introduces a spatial pyramid pooling layer to relax the constraint that input must have a fixed size. Unlike R-CNN, SPPnet extracts the feature maps from the whole image exactly once, independent of the region proposals, and then applies spatial pyramid pooling on each region proposal to get a fixed-length representation. This reorganization allows the computation to be easily shared between all region proposals. A drawback of SPPnet is that the fine-tuning algorithm of SPPnet can only update the fully connected layers, which makes it impossible to train the CNN feature extractor and SVM classifier jointly to further improve the performance. To amend this drawback, fast R-CNN [18] was proposed, which is an end-to-end trainable refinement of SPPnet. Under the framework of fast R-CNN, all network layers can be updated during fine-tuning, hence simplifying the learning process and improving the detection accuracy.

Both the frameworks of R-CNN [10] and fast R-CNN [18] need the inputs of region proposals, which generally come from handcrafted region proposal methods such as selective search [29] and EdgeBox [35]. However, the proposal generation is the bottleneck in the entire pipeline. To address this issue, faster R-CNN [40] was proposed, which consists of two modules. The first, called the *regional proposal network* (*RPN*), is an FCN for generating region proposals (each with a proposal bounding box and an objectness score) that will be fed into the second module. The second module is the fast R-CNN network for object detection. Faster R-CNN combines proposal generation and object detection into a unified network, in which the RPN module shares the same convolutional features with the fast R-CNN detection network; hence, it enables nearly cost-free region proposal generation.

Regression-based approaches

Regression-based COD methods are formulated as a regression problem with spatially separated bounding boxes and associated class probabilities [57]–[60]. A regression-based framework (for COD) is much simpler in comparison with object proposal-based methods because it does not require proposal generation and the subsequent pixel/feature resampling stages and encapsulates all stages in a single network [58]. Notice that the main difference between the box-regressing OD method and the regression COD method is that the objective of the former is to predict the box locations and one objectness score for each box location, while the objective of the latter is to predict the box locations and the object category scores (its dimension is dependent on the number of desired object categories) for each box location. Essentially, the models designed for regression COD are usually much more complex than those designed for box-regressing OD because the former need to simultaneously address the tasks of proposal localization and object category recognition. Thus, the multitask loss functions are more commonly used in regression-based COD than in box-regressing OD. You Only Look Once (YOLO) [57] and Single-Shot MultiBox Detector (SSD) [58] are two representative regression-based methods (for COD).

YOLO [57] opens the door to achieve real-time CNN-based object detection by formulating it as a regression problem. A unique feature of YOLO is that it unifies the separate components of object detection into a single convolutional network that simultaneously predicts multiple bounding boxes and class probabilities for those bounding boxes. The neural network globally reasons about the image when making predictions so it implicitly encodes contextual information about classes as well as their appearances. Compared to object proposal-based methods, YOLO is extremely fast, running at 45–150 frames per second without batch processing on a Titan X GPU.

However, it is still difficult to detect small-sized objects and achieve precise localization.

Thereafter, SSD [58] is proposed for improving the YOLO method. Specifically, it discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales at each feature map location, sharing the similar idea with RPN of faster R-CNN. At prediction time, SSD exports the scores for the presence of each object category in each default box and generates adjustments to the boxes to better match the object appearances. In addition, the network combines predictions from multiple feature maps with different resolutions to handle objects with various sizes. With the introduction of multiscale feature maps and the default boxes mechanism, SSD has achieved significant performance improvements to detect small-sized objects and also improved localization accuracy when compared with YOLO.

In addition, some recent works have also been developed to further boost the performance of CNN-based approaches for COD, such as hard negative mining [61], feature enhancement [41], [62], contextual information fusion [63]–[65], and so on. For example, to increase the capability to handle challenging situations with object rotation, within-class variability, and between-class similarity, Cheng et al. [62] proposed a rotation-invariant and Fisher-discriminative CNN model. Building upon the existing high-capacity CNN architectures, it was implemented by additionally introducing a rotation-invariant layer and a Fisher-discriminative layer, respectively.

Relationship among OD, SOD, and COD

Although OD, SOD, and COD are three individual research directions in object detection, a rich relationship can be observed among them.

Relationship between OD and SOD

On one hand, bottom-up SOD is able to provide informative prior knowledge to OD. Intuitively, the extracted bounding-box locations that are more attractive to the human visual system (more salient in the image scenes) would be more likely to contain the objects of interest. Based on this observation, several OD approaches have been designed by relying on some saliency cues. For example, one of the most classic OD approaches [1] selects object proposals by using three saliency cues, i.e., the multiscale saliency, color contrast, and edge density. Similarly, the work in [66] defined objectness as window saliency, which is the cost for composing the window by using the remaining parts of the image. This definition essentially subsumes the global rarity principle and extends it from pixel level (for SOD) to window level (for OD). In addition, Cheng et al. considered OD as a special case of bottom-up SOD [67], which indicates that OD can be effectively formulated by using the detection principles of SOD. Erhan et al. [37] also proposed a saliency-inspired neural network model for OD and achieved a promising performance.

On the other hand, some bottom-up SOD approaches are also established upon the OD results. When provided with the bounding boxes generated by OD, the SOD problem can be simplified as selecting the salient bounding boxes from the nonsalient ones. Based on this intuition, Chang et al. [68] proposed to integrate the objectness prior (including object size and location) and saliency prior to detect salient objects via a unified graphical model. Jiang et al. [69] integrated the objectness prior with focusness and objectness to keep completeness of detected salient regions for SOD. Li et al. [70] proposed to treat object candidates with high saliency values obtained from fixation prediction as salient objects. Since SOD requires uniformly highlighting the complete salient object, which differs from traditional saliency detection that requires only highlighting distinct local regions, the perceptibility of objects should be naturally encoded into effective SOD models. Objectness prior naturally provides an efficacious solution for this requirement.

Relationship between SOD and COD

As top-down SOD is highly task driven and knowledge driven, it requires high-level understanding of visual scenes, especially the category-level information of objects in the scenes. To achieve the goal of locating the intended objects in the scene, top-down SOD approaches usually need to acquire top-down knowledge for guiding the detection process [51]. Such top-down knowledge might come from memory (i.e., locating objects in the scene using knowledge from the corresponding training data, which is the model-based object detection) or object association (i.e., locating the corresponding objects in the scene using known or unknown exemplars, which is the exemplar-based object detection [71]–[73]). For example, in [50], Yang et al. detected top-down saliency by joint CRF and dictionary learning. The CRF model was initialed by training a linear SVM on the image patch representation and the corresponding patch labels, which essentially is a patch-level category-specific object detector. Some experimental results or discussions by previous works indicating that top-down cues provided by some category-specific object detectors (e.g., humans, faces, cars, words appearing in a given image, etc.) are playing an important role in visual attention mechanism can also be found in previous works [74], [75].

Besides SOD, top-down SOD especially can, in turn, provide helpful category-specific object prior for COD, particularly under weak supervision. As we know, weakly supervised object detection approaches [3], [76], [77] aim at learning category-specific object detectors only with image-level tags rather than the instance-level bounding-box annotations. In this scenario, how to obtain the initial object locations for certain object categories is a primary problem that needs to be addressed. By using the SOD approaches, [3] and [76] initialized category-specific object locations effectively and then adopted the iterative learning schemes to gradually refine the object detectors and locations in an iterative fashion. When the learning process converges, stronger object detectors can be learned to perform COD in test data. References [52], [53], and [77] also proposed to apply top-down saliency detection to discover the object locations in the weakly labeled training image, which subsequently can be used to train the category-specific object detectors.

Relationship between COD and OD

Extensive studies have demonstrated that OD can benefit from the COD task directly [10], [11], [18], [55]. Essentially, as we summarized in the section "Object Proposal-Based Approaches," one major stream of the COD approaches is built upon OD techniques, where OD can work as a separate preprocessing step [10] or an intrinsic component designed in the unified object detection framework [40]. COD approaches building upon OD techniques can usually obtain better performance than those building upon the sliding-window-based searching strategy [54], as OD techniques can provide helpful location prior to the object detection task, which can dramatically decrease unnecessary searching on many background image regions and thus effectively reduce false alarms.

The parameters in most of the OD approaches need to be learned from the collected training sets, which are usually from the PASCAL VOC benchmark [78]. Essentially, such training data with the ground-truth bounding boxes of objects from fewer than 20 categories can be considered as the restricted knowledge base for learning objectness detectors (e.g., in [19], [40], and [58]) as there are only limited object classes in the training data set. Although some works have demonstrated their proposed approaches, [2] and [17] are still able to generate object proposals for unseen objects, i.e., the object classes that are not contained in the training data set; such approaches may suffer more or less from performance drop due to the considerable domain shifts. It is worth mentioning that one of the most recent studies [31] has proposed the class-specific OD approach, which has a similar objective as the COD approaches.

Benchmarks and evaluation metrics

Benchmarks of OD

Two benchmarks are widely used in OD: the test set of PASCAL VOC 2007 [79] and the validation set of MS COCO [80]. Specifically, the test set of PASCAL VOC 2007 contains 4,952 images and 14,976 object instances from 20 categories. The large number of objects and high variety of categories, viewpoint, scale, position, occlusion, and illumination make this data set very popular to evaluate the OD methods since the goal of OD is find all possible objects in different image scenes. The MS COCO benchmark contains 80,000 training images and a total of about 500,000 instance annotations. The images in this data set are gathered from complex everyday scenes that contain common objects in their natural context. Thus, it is a more challenging data set for detecting objectness proposals. In most cases, the experiments for evaluating OD performance are carried out on the first 5,000 MS COCO validation images. Sometimes, another nonoverlapped 5,000 images are used as the validation data set.

Benchmarks of SOD

There are several benchmark data sets with different properties in the SOD community. The ECSSD data set [81] has 1,000 images with some overlapped images in the SOD data set [82]. Images in these two data sets usually have cluttered backgrounds and semantically meaningful foreground objects from various locations and scales. The PASCAL-S [70] data set is built on the PASCAL VOC segmentation challenge and has 850 images that usually contain cluttered backgrounds and multiple foreground objects. The HKU-IS [83] data set is a recently released SOD data set with 4,447 images. The images are collected from many challenging situations with multiple disconnected salient objects, salient objects touching image boundaries, and low color contrast. The DUT-OMRON data set [84] consists of 5,168 images, each of which usually has complicated backgrounds and contains contain one or two foreground objects. The THUR15K data set [85] contains 6,232 images coming from five object classes, i.e., butterfly, coffee mug, dog jump, giraffe, and plane. Some of the images in this data set have no foreground objects. The MSRA-10K data set [6] contains 10,000 images with various objects and is the extension of the MSRA-B data set [5]. Most of the images in these two data sets have only one foreground object and clear backgrounds. The SED data set is another widely used data set containing 200 images. Each image in this data set contains one and two foreground object(s).

Benchmarks of COD

The PASCAL VOC 2007 [79] and PASCAL VOC 2012 [16] data sets are two of the most commonly used benchmarks for evaluating various object detection methods. The PASCAL VOC 2007 data set contains a total of 9,963 images from 20 object categories, including 5,011 images for training and validation and 4,952 images for testing, in which ground-truth bounding boxes of 20 object categories were manually labeled. The PASCAL VOC 2012 data set is an extension of the PASCAL VOC 2007 data set, which contains a total of 22,531 images, including 11,540 images for training and validation and 10,991 images for testing. However, no ground-truth labels are provided in the testing set. Therefore, all of the methods should be evaluated by submitting their test results to the PASCAL VOC evaluation server.

MS COCO [80] is a newer object detection benchmark proposed in 2014 with the goal of boosting the state of the art in object recognition by placing the question of object recognition in the context of the broader question of scene understanding. The data set is significantly larger in number of instances per category than the PASCAL VOC data set. To be specific, this data set contains more than 200,000 images and 80 object categories, where the training set consists of 80,000 images, the validation set consists of 40,000 images, and the test set consists of 80,000 images. To limit overfitting and give researchers more flexibility to test their methods, the test set is divided into three splits, including test-dev, test-standard, and test-challenge. Test-dev is used for debugging and validation experiments and allows for unlimited submission to the evaluation server. Test-standard is used to maintain a public leaderboard that is updated upon submission. Test-challenge is used for competition. Most of the published works report their detection results on the test-dev set.

Evaluation metrics of OD

The metrics for evaluating object proposals generated by the OD approaches typically are functions of intersection over union (IOU) (or Jaccard index) between the proposal locations and the corresponding ground-truth annotations. Specifically, the IOU between a certain proposal location p_i and the ground-truth annotation g_j is defined as

$$\mathrm{IOU}(p_i, g_j) = \frac{\mathrm{area}(p_i \cap g_j)}{\mathrm{area}(p_i \cap g_j)}.$$

Based on IOU, recall can be computed as the fraction of ground-truth bounding boxes covered by proposal locations above a certain IOU overlap threshold. Then, three evaluation metrics that are widely used for evaluating objectness detection approaches are as follows:

- *recall versus proposal curve*, which depicts recall for different number of proposals
- *recall versus overlap curve*, which illustrates the variation of recall under different IOU overlap criteria
- *average recall* (*AR*), which computes the area under the "recall versus overlap" curve in a range of overlap values (usually set as 0.5–1.0).

For more details of the aforementioned evaluation metrics, please refer to [33], [34], and [36].

Evaluation metrics of SOD

Usually three standard evaluation metrics are widely used for SOD. The first one is the precision-recall curve (PRC). Specifically, given a saliency map S and the corresponding ground-truth saliency mask G, we first normalize S to [0], [1]. Then, we use a threshold T to convert S to a binary mask M. Afterward, precision and recall can be computed under the threshold T. When varying T from zero to one, we can obtain a series of precision-recall value pairs. Thus, we can draw the PRC to evaluate the performance of a model by treating SOD as a classification task.

The second metric is the F measure score, which comprehensively considers both precision and recall. Usually, an adaptive threshold is first used to segment the saliency map S and obtain the precision and recall values; then, the F measure score is calculated as a weighted harmonic mean of them.

Table 1. The evaluation of OD approaches on the test set of the PASCAL VOC 2007 benchmark. AR@N indicates the AR scores obtained under *N* proposals.

Methods	AR@500	AR@1,000	AR@2,000	Run Time (s)
BING	25.4	27.3	28.4	0.2
OBJ	29.6	30.9	31.6	3
EB	45.5	50.2	53.8	0.3
GOP	28.8	49.7	54.3	1.7
SS	45.0	51.9	58.4	10
RPN	46.7	48.6	49.8	0.2
MCG	50.7	56.4	61.2	34

Although these two metrics are widely used, they fail to consider true negative pixels. To solve this problem, the mean absolute error (MAE) is also commonly used. MAE measures the average pixel-wise absolute difference between S and G. For more details of the aforementioned evaluation metrics, please refer to [20], [83], and [87].

Evaluation metrics of COD

Average precision (AP) and mAP over all object classes are two standard and widely used metrics for evaluating various object detection methods. They are designed to less prefer the methods with missing object instances, duplicate detections of one instance, and false positive detections. To be specific, the AP computes the average value of precision over different levels of recall, i.e., the area under the PRC, so the higher the AP value, the better the performance and vice versa. While precision measures the fraction of detections that are true positives, recall measures the fraction of positives that are correctly detected.

A detection output by a method is assigned to true positive if the IOU between the predicted bounding box and ground-truth bounding box exceed a predefined threshold. Otherwise the detection is considered as a false positive. In addition, if multiple detection outputs overlap with the same ground-truth object, only one is considered as true positive and the others are considered as false positives.

The area overlap threshold is generally set to be 0.5 for the PASCAL VOC data set [16]. For the COCO data set, there exists a new evaluation metric, i.e., mAP is averaged over ten different IOU thresholds, from 0.5 to 0.95 with a step of 0.05 (written as 0.5:0.95). This places a significantly larger emphasis on localization when compared to the PASCAL VOC metric, which only requires IOU of 0.5. For more details of the aforementioned evaluation metrics, please refer to [40], [58], [and 60].

Experimental comparison

Experimental comparison of OD methods

In this section, we compare the performance of different OD approaches on the test set of PASCAL VOC 2007 and the validation set of MS COCO, in terms of AR. Specifically, we compare seven approaches, i.e., the BING [67], OBJ [1], EB [35], GOP [30], SS [29], RPN [40], and MCG [86], on the test set of PASCAL VOC 2007 in Table 1 and seven approaches, i.e., MCG [86], DeepMask [33], DeepMaskZoom [33], DeepMask2 [34], SharpMask [34], SharpMaskZoom [34], and FastMask [36], on the validation set of MS COCO in Table 2. Here, DeepMaskZoom indicates zooming original images to multiple scales and then applying DeepMask on each scale. The same approach is also used for ShapMaskZoom. DeepMask2 is implemented based on 39-layer ResNet [27] with a revised head component according to the network architecture in SharpMask. DeepMask and DeepMaskZoom are built on the VGG net [25], while DeepMask2, SharpMask, SharpMaskZoom, and FastMask are built on the 39-layer ResNet [27].

In Table 1, we can observe that MCG is the best OD method among those not using any deep-learning technique. It is even better than RPN, which is one OD method built on the CNN. This is mainly because RPN is designed with relatively simple network architecture, and it is only a part of the entire framework of faster R-CNN, where the ultimate goal is COD rather than OD. We can observe in Table 2 that many OD approaches based on deep-learning models can outperform the MCG to a large extent (about 5–13%), which is mainly due to the more powerful feature representations and better learning frameworks. More specifically, in the deep-learning-based OD approaches, except for the concrete designed network architectures, the final detection performances are also highly related to the adopted backbone CNN models. For example, the OD methods using ResNet as the backbone CNN model (e.g., DeepMask2 and SharpMaskZoom) can usually outperform the ones using VGG net (e.g., DeepMask and DeepMaskZoom) by about 2–6%. In addition, by comparing the performance of MCG in Tables 1 and 2, we can observe that the MS COCO benchmark is more challenging than the PASCAL VOC 2007. Greater efforts need to be applied to this research area, as there is much room for improvement on the performance on both benchmarks.

In Tables 1 and 2, we also report the run time for the compared OD methods. From these tables, we can observe that the fastest OD methods are BING and RPN, which can process each image in only about 0.2 s. SS and MSG can obtain outperforming operation among the OD methods without using a deep model. However, they have much larger computation costs as compared with all other methods. The OD methods built on deep models can usually perform efficiently in testing, as their computation cost is about 0.2–2 s per image.

Experimental comparison of SOD methods

In this section, we report quantitative comparison results of some representative DNN-based SOD models. We selected seven models that were published in 2015 and 2016 and have released their codes or computed saliency maps: LEGS [46], DHS [20], DCL [48], ELD [47], MCDL [87], MDF [83], and RFCN [49]. The experimental results of the ECSSD [81], PASCAL-S [70], and HKU-IS [83] data sets are reported in Table 3 (in terms of F measure), Table 4 (in terms of MAE), and Figure 3 (in terms of the PRC). According to the results in terms of the F and MAE, DHS [20] is the best saliency model, while DCL [48] and RFCN [49] come in second.

Table 2. The evaluation of OD approaches on the validation set of the MS COCO benchmark. AR@N indicates the AR scores obtained under *N* proposals.

Methods	AR@10	AR@100	AR@1,000	Run Time (s)
MCG	10.1	24.6	39.8	34
DeepMask	15.3	31.3	44.6	1.6
DeepMaskZoom	15.0	32.6	48.2	—
DeepMask2	18.0	34.8	47.0	0.46
SharpMask	19.2	36.2	48.3	0.76
SharpMaskZoom	19.2	39.0	53.2	1.5
FastMask	22.6	43.1	57.4	0.26

Among these models, LEGS, MDF, MCDL, and ELD are based on the extracted local regions (i.e., superpixels or object proposals). These methods usually independently estimate the saliency scores of each local region. Thus, they have limitations in incorporating contextual information effectively and efficiently. Consequently, as shown in Table 3, such methods usually have more computational costs, but they cannot obtain the outperforming detection results. On the contrary, DHS, DCL, and RFCN are based on FCNs, which can simultaneously perform saliency inference for all pixels. Thus, as shown in Table 3, they are usually very fast—especially for DHS—which just needs one forward without any preprocessing or postprocessing. Besides, since FCNs take the whole image as input, large contexts can be effectively incorporated by the successive deep convolutional layers, which explains the better performances obtained by these approaches.

For deep understanding of the performance of the SOD algorithm, we implemented ablation experiments on the most state-of-the-art saliency model, DHS [20]. Different from the tasks of OD and COD, which usually have the standard separation of the training-test data, the SOD task does not have such

Table 3. SOD performance of seven representative models in terms of F-measure (the higher, the better) and MAE (the lower, the better) scores on five benchmark data sets. We also report the running time of all models.

Dataset	SOD		ECSSD		SED		PASCAL-S		HKU-IS		Running Time (in Seconds)
Metric	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	
DHS	0.7628	0.1041	0.8645	0.0588	0.8460	0.0657	0.7610	0.0989	0.8539	0.0519	0.04
RFCN	0.7517	0.1386	0.8560	0.0952	0.8182	0.1085	0.7733	0.1183	0.8573	0.0786	4.6
ELD	0.7058	0.1337	0.8136	0.0783	0.8041	0.0858	0.7201	0.1254	0.7781	0.0713	0.5
DCL	0.7409	0.1253	0.8620	0.0679	0.8389	0.0946	0.7124	0.1229	0.8748	0.0481	1.5
MCDL	0.6774	0.1544	0.7957	0.1002	0.8117	0.0955	0.6912	0.1465	0.7569	0.1020	2.38
MDF	0.6875	0.1628	0.7937	0.1050	0.7952	0.1181	0.6970	0.1500	0.7835	0.1292	8.00
LEGS	0.6661	0.1780	0.7828	0.1180	0.7918	0.1144	0.7002	0.1551	0.7278	0.1191	2.00

Table 4. An ablation study of a DHS model in terms of F-measure (the higher, the better) and MAE (the lower, the better) scores on five benchmark data sets. We show the influence of convolutional backbone architectures and training image numbers.

Backbone Network	Training Image Number	Model Name	SOD		ECSSD		SED		PASCAL-S		HKU-IS	
			F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE
VGG	3167	DHS-1/3	0.7495	0.1045	0.8450	0.0593	0.8424	0.0712	0.7455	0.1004	0.8488	0.0516
	6333	DHS-2/3	0.7572	0.1096	0.8557	0.0598	0.8583	0.0732	0.7541	0.1007	0.8531	0.0539
	9500	DHS	0.7628	0.1041	0.8645	0.0588	0.8460	0.0657	0.7610	0.0989	0.8539	0.0519
	9500	DHS-RCL2	0.7390	0.1112	0.8425	0.0673	0.8346	0.0704	0.7404	0.1032	0.8293	0.0596
ResNet50	9500	DHS-Res50	0.7421	0.1099	0.8477	0.0637	0.8315	0.0728	0.7477	0.1039	0.8326	0.0591
ResNet101	9500	DHS-Res101	0.7587	0.1041	0.8527	0.0589	0.8482	0.0670	0.7441	0.0989	0.8402	0.0546

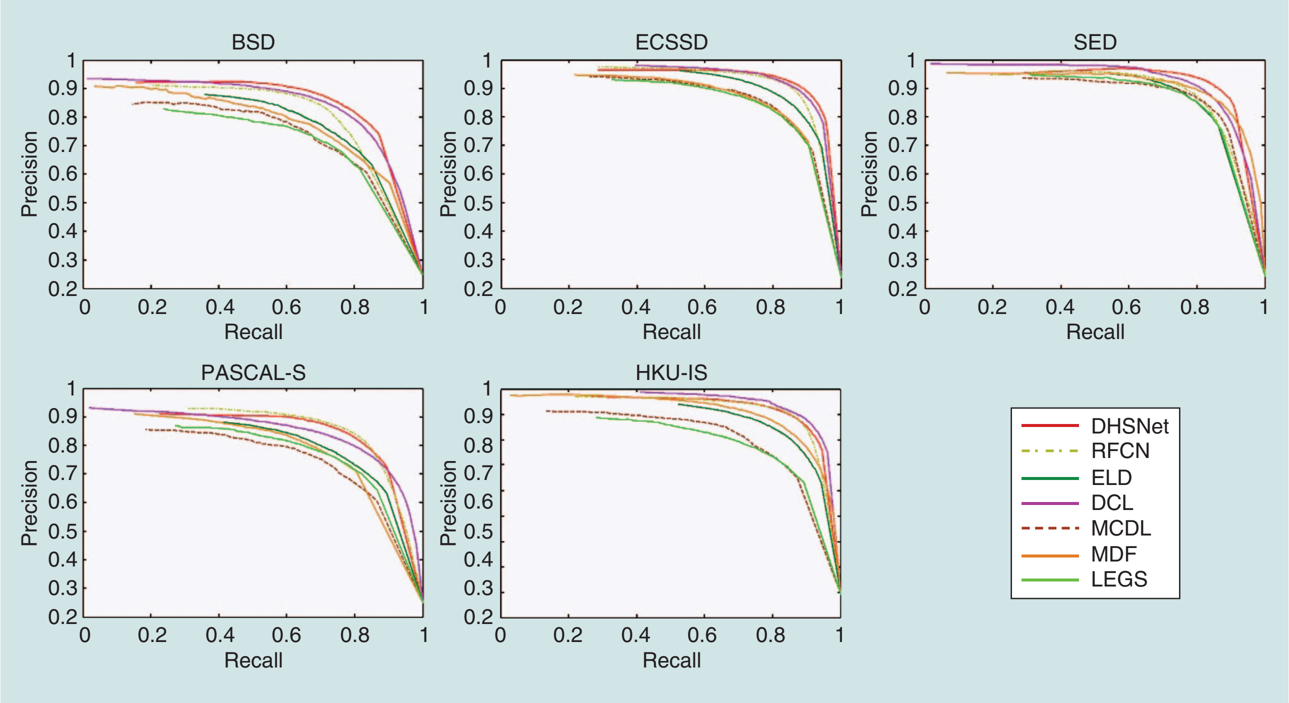

FIGURE 3. SOD results of seven representative models in terms of PRC.

a standard. Although most SOD algorithms implemented the training processes on the MSRA-10K and DUT-OMRON data sets and the test processes on other ones, they would obtain various performances when using different amounts of training data. To this end, we first analyze the influence of the number of training images. Specifically, we randomly selected one in three and two in three images from the original training set used in [20] (containing 9,500 images), generating two subtraining sets containing 3,167 and 6,333 images, respectively. By using these two subtraining sets, we obtained the performances of DHS-1/3 and DHS-2/3, as reported in Table 4. From the first three rows in Table 4, we can observe that, with the increase of the training image, the detection performance can be consistently improved. This indicates that we can improve the saliency model performance by using more training images. It is worth mentioning that, when comparing a model with other existing saliency models, one should also keep their training image number in a reasonable scope (fewer than 10,000) to ensure fair comparison.

Then, we analyze the influence of the backbone CNN model. The original DHS model used the VGG 16-layer network [25] as the backbone model. Here, we explored another two deeper CNN networks, i.e., ResNet50 and ResNet101 [27], as the backbone network of the DHS model. Note that, in the original DHS model, the authors used five step-wise refinements, which incorporate VGG feature maps from Conv4_3 to Conv1_2, thus recovering the saliency map from 28 × 28 to 224 × 224. However, the first convolutional layers of ResNet networks are all with stride 2, which means we can only use four step-wise refinements and recover the saliency map to 112 × 112. Thus, we also tested the results of the original DHS model with four refinement steps (named *DHS-RCL2*) for fair comparison. As the last four rows in Table 4 show, when using the same refinements steps, the deeper backbone model can obtain the better final performance. However, as the last refinement step plays a very importance role in boosting the final performance, using the ResNet as the backbone model cannot outperform the usage of the VGG net. As the network architectures similar to DHS are widely used in other modern detection approaches, the aforementioned analysis is important for us to balance the tradeoff brought by the ResNet.

Experimental comparison of COD methods

In this section, we select several representative CNN-based object detection methods that were published in the last three years (2015–2017) for performance comparison on the PASCAL VOC 2007 data set [79], PASCAL VOC 2012 data set [16], and MS COCO data set [80]. These methods include fast R-CNN [18], faster R-CNN [40], Inside-Outside Net (ION) [63], G-CNN [59], SSD [58], YOLO [57], and TOLOv2 [60]. Tables 5–7 report the detection results on PASCAL VOC 2007, PASCAL VOC 2012, and COCO data sets, respectively, by using the representative state-of-the-art methods. Table 8 presents the accuracy and speed comparison of various methods on the PASCAL VOC 2007.

As can be seen from Tables 5–7, regression-based COD methods, such as SSD [58] and TOLOv2 [60], could obtain comparatively even better accuracy compared with object proposal-based methods, such as fast R-CNN [18] and faster R-CNN [40]. Mainly, SSD512 [58] achieved the best accuracy on all three benchmarks (PASCAL VOC 2007, PASCAL VOC 2012, and COCO data sets). The computational cost comparison shown in Table 8 suggests that regression-based COD methods are much faster than object proposal-based methods with a comparable accuracy. Among them, YOLOv2 is the most competitive real-time detector and can also run at different resolutions for an easy tradeoff between speed and accuracy. The comprehensive comparison results in Tables 5–8 show that a regression-based method is a promising COD direction.

Recently, most detection frameworks, such as fast R-CNN [18], faster R-CNN [40], and SSD [58], rely on VGGNet-16 [25] as the backbone CNN model. Although VGGNet-16 is a powerful, accurate classification network, it is computationally expensive. The convolutional layers of VGGNet-16 require 30.69 billion floating point operations for a single pass over an image with a size of 224 × 224 pixels. The more

Table 5. Object detection results on the PASCAL VOC 2007 test set. The VGGNet-16 is the default CNN model for all methods if not following instructions. All methods (except for G-CNN) are trained on a union of VOC2007 trainval and VOC2012 trainval. The G-CNN method is trained on VOC2007 trainval set.

Method	aero	bike	bird	boat	bott	bus	car	cat	chair	cow	table	dog	horse	mbik	pson	plant	sheep	sofa	train	tv	mAP
Fast R-CNN	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	74.8	80.4	70.4	70.0
Faster R-CNN	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6	73.2
Faster R-CNN (ResNet-101)	79.8	80.7	76.2	68.3	55.9	85.1	85.3	89.8	56.7	87.8	69.4	88.3	88.9	80.9	78.4	41.7	78.6	79.8	85.3	72.0	76.4
ION	78.2	79.1	76.8	61.5	54.7	81.9	84.3	88.3	53.1	78.3	71.6	85.9	84.8	81.6	74.3	45.6	75.3	72.1	82.6	81.4	74.6
G-CNN	68.3	77.3	68.5	52.4	38.6	78.5	79.5	81.0	47.1	73.6	64.5	77.2	80.5	75.8	66.6	34.3	65.2	64.4	75.6	66.4	66.8
SSD300	75.5	80.2	72.3	66.3	47.6	83.0	84.2	86.1	54.7	78.3	73.9	84.5	85.3	82.6	76.2	48.6	73.9	76.0	83.4	74.0	74.3
SSD512	82.4	84.7	78.4	73.8	53.2	86.2	87.5	86.0	57.8	83.1	70.2	84.9	85.2	83.9	79.7	50.3	77.9	73.9	82.5	75.3	76.8

Table 6. Object detection results on the PASCAL VOC 2012 test set. The VGGNet-16 is the default CNN model for all methods (except for YOLO and YOLOv2) if not following instructions. All methods (except for ION and G-CNN) are trained on a union of VOC2007 trainval, VOC2007 test, and VOC2012 trainval. ION and G-CNN are trained on a union of VOC2007 trainval and VOC2012 trainval.

Method	aero	bike	bird	boat	bott	bus	car	cat	chair	cow	table	dog	horse	mbik	pson	plant	sheep	sofa	train	tv	mAP
Fast R-CNN	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	87.5	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2	68.4
Faster R-CNN	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5	70.4
Faster R-CNN (ResNet-101)	86.5	81.6	77.2	58.0	51.0	78.6	76.6	93.2	48.6	80.4	59.0	92.1	85.3	84.8	80.7	48.1	77.3	66.5	84.7	65.6	73.8
ION	86.9	84.5	75.2	58.2	57.7	80.5	78.3	90.4	54.4	79.9	60.5	88.4	83.0	83.0	81.2	50.7	77.3	67.6	83.5	72.3	74.7
G-CNN	82.0	76.1	69.3	49.9	40.1	75.2	69.5	86.3	42.3	72.3	50.8	84.7	77.8	77.2	68.0	38.1	68.4	59.8	79.1	61.9	66.4
SSD300	85.6	80.1	70.5	57.6	46.2	79.4	76.1	89.2	53.0	77.0	60.8	87.0	83.1	82.3	79.4	45.9	75.9	69.5	81.9	67.5	72.4
SSD512	87.4	82.3	75.8	59.0	52.6	81.7	81.5	90.0	55.4	79.0	59.8	88.4	84.3	84.7	83.3	50.2	78.0	66.3	86.3	72.0	74.9
YOLO	77.0	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	71.3	63.5	28.9	52.2	54.8	73.9	50.8	57.9
YOLOv2	86.3	82.0	74.8	59.2	51.8	79.8	76.5	90.6	52.1	78.2	58.5	89.3	82.5	83.4	81.3	49.1	77.2	62.4	83.8	68.7	73.4

Table 7. Object detection results on the MS COCO test-dev2015 in terms of the mAP score. The VGGNet-16 is the default CNN model for all methods (except for YOLOv2).

		IOU Threshold		
Method	Data	0.5:0.95	0.5	0.75
Fast R-CNN	train	20.5	39.9	19.4
Faster R-CNN	trainval	21.9	42.7	—
ION	train	23.6	43.2	23.6
SSD300	trainval35k	23.2	41.2	23.4
SSD512	trainval35k	26.8	46.5	27.8
YOLOv2	trainval35k	21.6	44.0	19.2

Table 8. Accuracy and speed comparison on PASCAL VOC 2007 test set. All methods are trained on union of VOC2007 trainval and VOC2012 trainval. Timings are all performed on a Geforce GTX Titan X. The results are from [60].

Method	mAP	FPS
Fast R-CNN	70.0	0.5
Faster R-CNN	73.2	7
Faster R-CNN (ResNet-101)	76.4	5
SSD300	74.3	46
SSD512	76.8	19
YOLO	63.4	45
YOLOv2 288×288	69.0	91
YOLOv2 352×352	73.7	81
YOLOv2 416×416	76.8	67
YOLOv2 480×480	77.8	59
YOLOv2 544×544	78.6	40

recently proposed YOLO [57] uses a custom network based on the GoogleNet architecture [26] to improve the detection speed. The YOLO network is indeed faster than VGGNet-16-based methods, using only 8.52 billion operations for a forward pass but with a slightly worse accuracy than VGG-Net-16. To further reduce the computational burden and meanwhile boost the accuracy, YOLOv2 [60] proposes a new classification model, named *Darknet-19*, which requires only 5.58 billion operations to process an image. The comprehensive comparison results of accuracy and speed show that the regression-based method is a promising COD direction by designing more practical model architectures such as YOLOv2 [60].

Discussions

Advantages brought by deep learning

Based on our analysis, the advantages brought by deep-learning-based object detectors can be summarized in the following four aspects.

Powerful feature representation

One of the most important reasons for developing deep-learning-based object detectors is the powerful feature representations that are built during the learning procedure. This has been clearly demonstrated in [10], where Girshick et al. compared the COD results of their proposed CNN-based object detectors with the DPM [54] baselines using one of the widely used HOG feature and its extended versions. The experimental results demonstrate that the CNN-based object detector outperforms those of the DPM baselines by more than 24.2% in terms of AP. In addition, Girshick et al. [10] analyzed the representation power of different network layers of the adopted CNN architecture. From their experimental results, we can observe that, with fine-tuning, the deeper layers can obtain better performance than the shallower layers, which indicates that feature representations extracted from the deep layers of CNN can effectively capture informative semantics for representing image regions. It is also worth mentioning that even the features extracted in layer "pool5," which use only 6% of the whole network parameters, can already obtain more than 10% performance gain (in terms of AP) as compared to the conventional object detection approaches. Similar experimental results can also be found in the studies of OD and SOD. For example, by using only simple CNN architectures to build features in their frameworks, RPN [40], DeepProposal [17], and DeepBox [2] can already outperform the existing OD methods obviously using conventional feature representations. In SOD, [83] and [20] point out that how to build real, meaningful feature representations is one of the most critical issues in SOD, and they have demonstrated that this problem can be effectively addressed by designing rational deep network architectures.

End-to-end learning

Another advantage of the deep-learning-based object detection approaches are due to their end-to-end learning frameworks. As we know, the conventional object detection approaches usually need separate computational blocks such as feature extraction and pattern classification (e.g., [35] and [67] for OD, [88] and [89] for SOD, and [54] and [90] for COD). The deep-learning-based approaches (e.g., [17] and [36] for OD, [20] and [87] for SOD, and [18] and [57] for COD) can obtain the desired object detection results from the original input images through only one unified CNN model. Such an end-to-end learning manner could bring two benefits when compared with the conventional approaches:

- It can largely reduce the complexity for choosing an optimal method from a number of candidates to each of the computational blocks in the conventional object detection approaches.
- Learning in such an end-to-end manner can determine the parameters of the entire model based on the learning objective.

Compared with the conventional ways (for designing handcrafted features), such a learning manner can dramatically reduce the loss of useful information during the whole system.

Multistage, multitask objective

Thanks to the end-to-end learning paradigm, the deep-learning-based modern object detection approaches can involve required learning objectives in multiple learning stages and with multiple learning tasks flexibly. For example, in OD, the deep network proposed by FastMask [36] contains the learning stages of semantic feature extraction and sliding-window-based proposal generation. The learning objective consists of three terms: the confidence loss, segmentation loss, and region attention loss. In SOD, LEGS [46] contains two cascade learning stages. The first one is the local estimation stage with the goal of learning local discriminative patch features. The second is the global search stage with the goal of exploiting the complex relationships among global saliency cues. DHS [20] also has two learning stages, where the first stage is for saliency inference while the second stage is for detail rendering. In COD, faster R-CNN [40] is designed with two cascade learning stages for learning to extract object proposal regions and recognize the object category of each extracted object proposal region, respectively. Fast R-CNN [18] adopts the multitask loss to simultaneously learn category-specific object detectors and bounding-box regression functions. Compared with the conventional object detection approaches, the most important advantage of such multistage and multitask objectives is that the corresponding learning procedure can consider all the designed learning objectives to determine the optimal parameters of the detection network, while the conventional approach can only train the main object detection model by considering one major learning objective and making use of other useful factors in the preprocessing or postprocessing stages.

Large-scale learning and knowledge transfer

In contrast to the learning models with shallow architectures where few parameters are preferable to avoid overfitting, the success of deep-learning models is mainly due to a large number of hidden neurons, which often results in millions of free parameters. Consequently, DNNs often need large-scale training data to achieve their full learning capability, which enables the deep models to capture much richer patterns from the training data than the shallow models. Besides the capability in knowledge mining, another advantage of deep-learning models is that they can conveniently transfer the learned knowledge to relevant task or scenario. This is mainly achieved by fine-tuning the deep models pretrained in the source domain by using the data in the target domain. For example, the most representative deep-learning-based COD approach [10] adopted the CNN model pretrained on ImageNet under the task of image classification. The experimental results indicate that directly using such a network can already obtain obvious improvement when compared with the previous state-of-the-art results, which indicates that patterns captured in large-scale learning could be powerful enough to complete a wide range of tasks. In addition, the experimental results in [10] show that the fine-tuned network can further achieve significant performance gain, which demonstrates the simple but effective knowledge transfer capability of the deep-learning model. Based on such capability, a large amount of the modern object detection approaches [2], [19], [34], [36], [42], [58], [59], [62] can benefit from a simple but effective pretraining phase.

Future research directions

Although deep-learning-based object detection approaches have achieved great success in this research field recently, there are still several challenging, yet interesting, research directions that need to be considered in the future.

Training object detectors with limited human annotation

Despite the significant performance gain obtained by the recent deep-learning-based object detection approaches, the problems in the research field of object detection are still largely underaddressed in practice because most of those approaches heavily rely on an unparalleled and tremendous amount of human-labeled training data. Under this circumstance, people are encumbered by the great burden of spending energy and time on tedious data annotation for training deep object detectors. Based on our statistics, one needs to spend approximately 15 s (with the help of some auxiliary tools like LabelMe) to draw a bounding-box annotation that can properly enclose the objects of interest. With this in mind, there might be hundreds of thousands of training images that need to be annotated manually, and each image might contain multiple objects from various categories. To alleviate this problem, weakly supervised object detection approaches [3], [13], [76], [99] have received wide interest in recent years. Nevertheless, the obtained performances are still far from satisfactory—they can achieve up to only 50% of performance obtained by the corresponding fully supervised object detection approaches. Thus, further efforts are still needed to address this issue.

Detection for unseen object categories

Most of the existing object detection approaches are evaluated for the images from the same set of object categories as in the training set. However, the ultimate goal of object detection is to detect all objects from any possible categories in given test images. Essentially, in real-world applications, we lack sufficient annotations for all object categories. The widely used PASCAL VOC and MS COCO benchmarks only contain 20 and 80 object categories, respectively, which are far from enough. The ILSVRC object detection benchmark contains 200 object categories, which, however, is still insufficient. In the absence of bounding-box-level annotations for many categories, one future direction is to establish zero-shot learning-based schemes (for object detection), where the combination of existing detectors and cross-concept/category mappings between these detectors can allow us to build object detectors for unseen classes. As an emerging branch of SOD, cosaliency detection [92]–[94] and event saliency detection [95] approaches could also be possible ways for detecting unseen objects/events, because they can learn from any given group of images/videos that contain the co-occurring but unknown objects/events.

Novel learning strategies to improve detection robustness

Another future direction is to increase the detection robustness to object categories that are trained with imbalanced data or noisy data. Here the imbalance issue mainly refers to the long-tailed distribution of sample numbers in different classes in object detection. The long-tailed property indicates the phenomena that a small number of object classes appear very often while most of other classes appear rarely. For example, in the PASCAL VOC and ImageNet object detection data sets, object categories such as *person* have many more samples than other object categories such as *sheep*. Some analysis and empirical results have shown that object categories with more samples would dominate the learned object detectors, leading to the insufficient learning of other object categories with fewer samples. Consequently, one future direction against this problem is to establish novel learning schemes to learn with more uniformly distributed sample numbers across different object categories. Thanks to some of the most recent generative learning models, such as generative adversarial networks (GANs), the samples of the object categories that are "in the tail" can be enriched by synthesizing usable data from latent noise vectors. On the contrary, large-scale human annotation would inevitably introduce noisy annotations such as missing labeling or incorrect labeling. To address the noise issue in human annotation, further research could design weighting-based learning mechanisms (such as the models based on self-paced learning [94], [96] and curriculum learning [97], [98]) to further improve the learning robustness when the human annotations of the learned object categories are noisy.

Unified learning framework for OD, SOD, and COD

The current research in the field of object detection has proposed some effective deep leaning-based frameworks, such as [40] and [41], for OD and COD at the same time. The experimental results have demonstrated that, by jointly optimizing the network parameters for the tasks of OD and category-specific detection, the network could additionally explore the underlying relationship between these tasks and capture the common and informative patterns that can benefit from both tasks. Essentially, as discussed previously, there is a rich relationship among the OD, SOD, and COD. Thus, it is of great interest to establish novel frameworks, especially the deep-learning-based frameworks, to simultaneously solve the common problems in these three directions. One possible way is to build a deep network that combines the blocks of attention modeling, proposal mining, and category recognizing into a unified learning framework. In this way, the shared informative patterns among all three tasks could be captured by the learned model, which can further improve the performance of each task.

Detection-based higher-level visual understanding

The appearance of the recent advanced object detection techniques has facilitated the development of some higher-level visual understanding tasks that have not been touched on before. One representative example of such tasks is called *image/video captioning*. The basic goal of this task is to automatically generate a sentence to describe the content of any given image/video. The object detection techniques can provide critical information of object location and category for interpreting what the things in the image/video scene are, where are they placed, and what they are doing with the interactive objects. Essentially, accurate object detection is the key to link the visual domain to the language domain. Along this line of research, there are still many unexplored yet interesting detection-based applications (higher-level visual understanding tasks) that form another branch of a future research direction.

Conclusions

In this article, we have reviewed the most recent progress in object detection, which is mainly based on advanced deep-learning techniques. Specifically, the modern approaches, benchmark data sets, and evaluation metrics were reviewed for OD, SOD, and COD—the three directions in object detection. We comprehensively analyzed the relationship among these directions and provided insightful discussions on the advantages from deep learning and offered some possible future directions.

Acknowledgment

Dingwen Zhang and Gong Chen are the corresponding authors of this article.

Authors

Junwei Han (junweihan2010@gmail.com) received his B.S., M.S., and Ph.D. degrees in pattern recognition and intelligent systems in 1999, 2001, and 2003, respectively, all from Northwestern Polytechnical University, Xi'an, China, where he is currently a professor. He was a research fellow at Nanyang Technological University, The Chinese University of Hong Kong, Dublin City University, and the University of Dundee from 2003 to 2010. His research interests include computer vision and brain-imaging analysis. He is an associate editor of *IEEE Transactions on Human-Machine Systems*, *Neurocomputing*, and *Machine Vision and Applications*.

Dingwen Zhang (zdw2006yyy@mail.nwpu.edu.cn) received his B.E. degree in automation from Northwestern Polytechnical University, Xi'an, China, in 2012, and he is currently pursuing his Ph.D. degree there. Since October 2015, he has been a visiting scholar at Carnegie Mellon University, Pittsburgh, Pennsylvania. His research interests include computer vision and multimedia processing.

Gong Cheng (chenggong1119@gmail.com) received his B.E. degree in automation from Xidian University, Xi'an, China, in 2007 and his M.S. and Ph.D. degrees in pattern recognition and machine intelligence from Northwestern Polytechnical University, Xi'an, China, in 2010 and 2013, respectively; he is currently an associate professor at the latter. His main research interests are in computer vision and object detection.

Nian Liu (liunian228@gmail.com) received his B.E. and M.E. degrees in automation from Northwestern Polytechnical

University (NPU), Xi'an, China, in 2012 and 2015, respectively. He is currently pursuing his Ph.D. degree in the School of Automation at NPU. His research interests include computer vision with a focus on saliency detection and deep learning.

Dong Xu (dong.xu@sydney.edu.au) received his B.E. and Ph.D. degrees in electronic engineering from the University of Science and Technology of China, Hefei, in 2001 and 2005, respectively. He is currently a professor with the School of Electrical and Information Engineering, the University of Sydney, Australia. He is a Senior Member of the IEEE and a fellow of the International Association of Pattern Recognition.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 34, no. 11, pp. 2189–2202, 2012.

[2] W. Kuo, B. Hariharan, and J. Malik, "Deepbox: Learning objectness with convolutional networks," in *Proc. IEEE Int. Conf. Computer Vision*, 2015, pp. 2479–2487.

[3] T. Deselaers, B. Alexe, and V. Ferrari, "Weakly supervised localization and learning with generic knowledge," *Int. J. Computer Vision*, vol. 100, no. 3, pp. 275–293, 2012.

[4] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 37, no. 9, pp. 1834–1848, 2015.

[5] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 33, no. 2, pp. 353–367, 2011.

[6] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 37, no. 3, pp. 569–582, 2015.

[7] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 35, no. 1, pp. 185–207, 2013.

[8] X. Li, T. Uricchio, L. Ballan, M. Bertini, C. G. Snoek, and A. D. Bimbo, "Socializing the semantic gap: A comparative survey on image tag assignment, refinement, and retrieval," *ACM Comput. Surveys*, vol. 49, no. 1, pp. 14, 2016.

[9] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2015, pp. 447–456.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[11] W. Ouyang, X. Wang, X. Zeng, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, and C.-C. Loy, "Deepid-net: Deformable deep convolutional neural networks for object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2015, pp. 2403–2412.

[12] J. Tighe, M. Niethammer, and S. Lazebnik, "Scene parsing with object instance inference using regions and per-exemplar detectors," *Int. J. Computer Vision*, vol. 112, no. 2, pp. 150–171, 2015.

[13] A. Prest, C. Schmid, and V. Ferrari, "Weakly supervised learning of interactions between humans and objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 34, no. 3, pp. 601–614, 2012.

[14] Y. LeCun, F. J. Huang, and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004, pp. 97–104.

[15] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Proc. Advances in Neural Information Processing Systems*, 2013, pp. 2553–2561.

[16] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *Int. J. Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.

[17] A. Ghodrati, A. Diba, M. Pedersoli, T. Tuytelaars, and L. Van Gool, "Deepproposal: Hunting objects by cascading deep convolutional layers," in *Proc. IEEE Int. Conf. Computer Vision*, 2015, pp. 2578–2586.

[18] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Computer Vision*, 2015, pp. 1440–1448.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," arXiv Preprint, arXiv:1506.02640, 2015.

[20] N. Liu and J. Han, "DHSNet: Deep hierarchical saliency network for salient object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 678–686.

[21] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybernet.*, vol. 36, no. 4, pp. 193–202, 1980.

[22] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.

[23] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv Preprint, arXiv:1409.1556, 2014.

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[29] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *Int. J. Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[30] P. Krähenbühl and V. Koltun, "Geodesic object proposals," in *Proc. European Conf. Computer Vision*, 2014, pp. 725–739.

[31] L. Bazzani, A. Bergamo, D. Anguelov, and L. Torresani, "Self-taught object localization with deep networks," in *Proc. IEEE Winter Conf. Applications of Computer Vision*, 2016, pp. 1–9.

[32] X. Hou, and L. Zhang, "Saliency detection: A spectral residual approach," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[33] P. O. Pinheiro, R. Collobert, and P. Dollar, "Learning to segment object candidates," in *Proc. Advances in Neural Information Processing Systems*, 2015, pp. 1990–1998.

[34] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *Proc. European Conf. Computer Vision*, 2016, pp. 75–91.

[35] C. L. Zitnick, and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. European Conf. Computer Vision*, 2014, pp. 391–405.

[36] H. Hu, S. Lan, Y. Jiang, Z. Cao, and F. Sha, "FastMask: Segment multi-scale object candidates in one shot," arXiv Preprint, arXiv:1612.08843, 2016.

[37] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2014, pp. 2147–2154.

[38] C. Szegedy, S. Reed, D. Erhan, D. Anguelov, and S. Ioffe, "Scalable, high-quality object detection," arXiv Preprint, arXiv:1412.1441, 2014.

[39] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[40] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[41] T. Kong, A. Yao, Y. Chen, and F. Sun, "HyperNet: Towards accurate region proposal generation and joint object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 845–853.

[42] H. Li, Y. Liu, W. Ouyang, and X. Wang, "Zoom out-and-in network with recursive training for object proposal," arXiv Preprint, arXiv:1702.05711, 2017.

[43] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A survey," arXiv Preprint, arXiv:1411.5878, 2014.

[44] A. Borji, H. R. Tavakoli, D. N. Sihite, and L. Itti, "Analysis of scores, data sets, and models in visual saliency prediction," in *Proc. IEEE Int. Conf. Computer Vision*, 2013, pp. 921–928.

[45] J. Han, D. Zhang, X. Hu, L. Guo, J. Ren, and F. Wu, "Background prior-based salient object detection via deep reconstruction residual," *IEEE Trans. Circuits Systems Video Technol.*, vol. 25, no. 8, pp. 1309–1321, 2015.

[46] L. Wang, H. Lu, X. Ruan, and M.-H. Yang, "Deep networks for saliency detection via local estimation and global search," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2015, pp. 3183–3192.

[47] G. Lee, Y.-W. Tai, and J. Kim, "Deep saliency with encoded low level distance map and high level features," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 660-668.

[48] G. Li, and Y. Yu, "Deep contrast learning for salient object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 660–668.

[49] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan, "Saliency detection with recurrent fully convolutional networks," in *Proc. European Conf. Computer Vision*, 2016, pp. 825–841.

[50] J. Yang, and M.-H. Yang, "Top-down visual saliency via joint CRF and dictionary learning," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2012, pp. 2296–2303.

[51] S. He, R. W. Lau, and Q. Yang, "Exemplar-driven top-down saliency detection via deep association," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 5723–5732.

[52] H. Cholakkal, J. Johnson, and D. Rajan, "Backtracking ScSPM image classifier for weakly supervised top-down saliency," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 5278–5287.

[53] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff, "Top-down neural attention by excitation backprop," in *Proc. European Conf. Computer Vision*, 2016, pp. 543–559.

[54] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 32, no. 9, pp. 1627–1645, 2010.

[55] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 37, no. 9, pp. 1904–1916, 2015.

[56] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," arXiv Preprint, arXiv:1312.6229, 2013.

[57] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[58] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. European Conf. Computer Vision*, 2016, pp. 21–37.

[59] M. Najibi, M. Rastegari, and L. S. Davis, "G-CNN: An iterative grid based object detector," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 2369–2377.

[60] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," arXiv Preprint, arXiv:1612.08242, 2016.

[61] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 761–769.

[62] G. Cheng, P. Zhou, and J. Han, "RIFD-CNN: Rotation-invariant and fisher discriminative convolutional neural networks for object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 2884–2893.

[63] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 2874–2883.

[64] S. Gidaris, and N. Komodakis, "Object detection via a multi-region and semantic segmentation-aware CNN model," in *Proc. IEEE Int. Conf. Computer Vision*, 2015, pp. 1134–1142.

[65] A. Shrivastava, and A. Gupta, "Contextual priming and feedback for faster R-CNN," in *Proc. European Conf. Computer Vision*, 2016, pp. 330–348.

[66] J. Feng, Y. Wei, L. Tao, C. Zhang, and J. Sun, "Salient object detection by composition," in *Proc. IEEE Int. Conf. Computer Vision*, 2011, pp. 1028–1035.

[67] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "BING: Binarized normed gradients for objectness estimation at 300fps," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2014, pp. 3286–3293.

[68] K.-Y. Chang, T.-L. Liu, H.-T. Chen, and S.-H. Lai, "Fusing generic objectness and visual saliency for salient object detection," in *Proc. IEEE Int. Conf. Computer Vision*, 2011, pp. 914–921.

[69] P. Jiang, H. Ling, J. Yu, and J. Peng, "Salient region detection by UFO: Uniqueness, focusness and objectness," in *Proc. IEEE Int. Conf. Computer Vision*, 2013, pp. 1976–1983.

[70] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, "The secrets of salient object segmentation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2014, pp. 280–287.

[71] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-svms for object detection and beyond," in *Proc. IEEE Int. Conf. Computer Vision*, 2011, pp. 89–96.

[72] M. Bar, "The proactive brain: Using analogies and associations to generate predictions," *Trends Cogn. Sci.*, vol. 11, no. 7, pp. 280–289, 2007.

[73] R. M. Nosofsky, "Attention, similarity, and the identification–categorization relationship," *J. Exp. Psychol.*, vol. 115, no. 1, pp. 39, 1986.

[74] A. Borji, "Boosting bottom-up and top-down visual features for saliency estimation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2012, pp. 438–445.

[75] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *Proc. IEEE Int. Conf. Computer Vision*, 2009, pp. 2106–2113.

[76] D. Zhang, D. Meng, L. Zhao, and J. Han, "Bridging saliency detection to weakly supervised object detection based on self-paced curriculum learning," in *Proc. Int. Joint Conf. Artificial Intelligence*, 2016, pp. 3538–3544.

[77] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 2921–2929.

[78] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[79] M. Everingham, A. Zisserman, C. K. Williams, L. Van Gool, M. Allan, C. M. Bishop, O. Chapelle, N. Dalal, T. Deselaers, and G. Dorkó, "The PASCAL visual object classes challenge 2007 (VOC2007) results," 2007. [Online]. Available: http://host.robots.ox.ac.uk/pascal/VOC/voc2007/workshop/index.html

[80] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. European Conf. Computer Vision*, 2014, pp. 740–755.

[81] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2013, pp. 1155–1162.

[82] V. Movahedi and J. H. Elder, "Design and perceptual validation of performance measures for salient object segmentation," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition Workshops*, 2010, pp. 49–56.

[83] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2015, pp. 5455–5463.

[84] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2013, pp. 3166–3173.

[85] M.-M. Cheng, N. J. Mitra, X. Huang, and S.-M. Hu, "Salientshape: Group saliency in image collections," *Visual Computer*, vol. 30, no. 4, pp. 443–453, 2014.

[86] J. Pont-Tuset, P. Arbelaez, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping for image segmentation and object proposal generation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 39, no. 1, pp. 128–140, 2017.

[87] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2015, pp. 1265–1274.

[88] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2013, pp. 2083–2090.

[89] P. Mehrani and O. Veksler, "Saliency segmentation based on learning and graph cut refinement," in *Proc. British Machine Vision Conf.*, 2010, pp. 1–12.

[90] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 36, no. 8, pp. 1532–1545, 2014.

[91] Q. Hou, M.-M. Cheng, X. Hu, Z. Tu, and A. Borji, "Deeply supervised salient object detection with short connections," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017, pp. 3203–3212.

[92] D. Zhang, J. Han, J. Han, and L. Shao, "Cosaliency detection based on intrasaliency prior transfer and deep intersaliency mining," *IEEE Trans. Neural Networks Learning Syst.*, vol. 27, no. 6, pp. 1163–1176, 2016.

[93] D. Zhang, J. Han, C. Li, J. Wang, and X. Li, "Detection of co-salient objects by looking deep and wide," *Int. J. Computer Vision*, vol. 120, no. 2, pp. 215–232, 2016.

[94] D. Zhang, D. Meng, and J. Han, "Co-saliency detection via a self-paced multiple-instance learning framework," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 39, no. 5, pp. 865–878, 2017.

[95] D. Zhang, J. Han, L. Jiang, S. Ye, and X. Chang, "Revealing event saliency in unconstrained video collection," *IEEE Trans. Image Processing*, vol. 26, no. 4, pp. 1746–1758, 2017.

[96] D. Zhang, L. Yang, D. Meng, D. Xu, and J. Han, "SPFTN: A self-paced fine-tuning network for segmenting objects in weakly labelled videos," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017, pp. 4429–4437.

[97] D. Zhang, J. Han, Y. Yang, and D. Huang, "Learning category-specific 3D shape models from weakly labeled 2D images," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017, pp. 4573–4581.

[98] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. Annu. Int. Conf. Machine Learning*, 2009, pp. 41–48.

[99] Y. Yuan, X. Liang, X. Wang, D.-Y. Yeung, and A. Gupta, "Temporal dynamic graph LSTM for action-driven video object detection," arXiv Preprint, arXiv:1708.00666, 2017.