Deep Learning for Generic Object Detection: A Survey

Li Liu [1,2] · **Wanli Ouyang** [3] · **Xiaogang Wang** [4] ·
Paul Fieguth [5] · **Jie Chen** [2] · **Xinwang Liu** [1] · **Matti Pietikäinen** [2]

Received: 12 September 2018

Abstract Generic object detection, aiming at locating object instances from a large number of predefined categories in natural images, is one of the most fundamental and challenging problems in computer vision. Deep learning techniques have emerged in recent years as powerful methods for learning feature representations directly from data, and have led to remarkable breakthroughs in the field of generic object detection. Given this time of rapid evolution, the goal of this paper is to provide a comprehensive survey of the recent achievements in this field brought by deep learning techniques. More than 250 key contributions are included in this survey, covering many aspects of generic object detection research: leading detection frameworks and fundamental subproblems including object feature representation, object proposal generation, context information modeling and training strategies; evaluation issues, specifically benchmark datasets, evaluation metrics, and state of the art performance. We finish by identifying promising directions for future research.

Keywords Object detection · deep learning · convolutional neural networks · object recognition

1 Introduction

As a longstanding, fundamental and challenging problem in computer vision, object detection has been an active area of research for several decades. The goal of object detection is to determine whether or not there are any instances of objects from the given categories (such as humans, cars, bicycles, dogs and cats) in some given image and, if present, to return the spatial location and extent of each object instance (*e.g.,* via a bounding box [53, 179]). As the cornerstone of image understanding and computer vision, object detection forms the basis for solving more complex or high level vision tasks such as segmentation, scene understanding, object tracking, image captioning, event detection, and activity recognition. Object detection has a wide range of applications in many areas of artificial intelligence and information technologies, including robot vision, consumer electronics, security, autonomous driving, human computer interaction, content based image retrieval, intelligent video surveillance, and augmented reality.

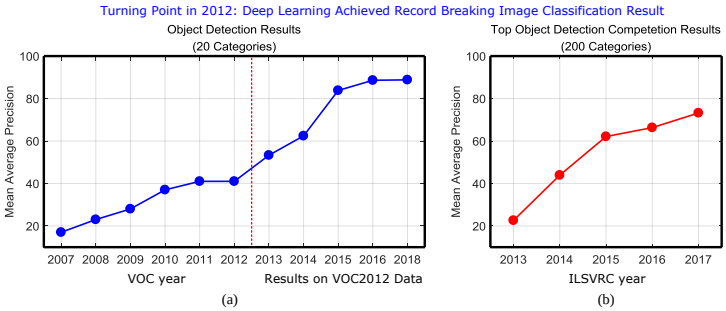


Fig. 1 Recent evolution of object detection performance. We can observe significant performance (mean average precision) improvement since deep learning entered the scene in 2012. The performance of the best detector has been steadily increasing by a significant amount on a yearly basis. (a) Results on the PASCAL VOC datasets: Detection results of winning entries in the VOC2007-2012 competitions (using only provided training data). (b) Top object detection competition results in ILSVRC2013-2017 (using only provided training data).

Recently, deep learning techniques [81, 116] have emerged as powerful methods for learning feature representations automatically from data. In particular, these techniques have provided significant improvement for object detection, a problem which has attracted enormous attention in the last five years, even though it has been studied for decades by psychophysicists, neuroscientists, and engineers.

Object detection can be grouped into one of two types [69, 240]: detection of specific instance and detection of specific categories. The first type aims at detecting instances of a particular object (such as Donald Trump's face, the Pentagon building, or my dog Penny), whereas the goal of the second type is to detect different instances of predefined object categories (for example humans,

✉ Li Liu (li.liu@oulu.fi)
Wanli Ouyang (wanli.ouyang@sydney.edu.au)
Xiaogang Wang (xgwang@ee.cuhk.edu.hk)
Paul Fieguth (pfieguth@uwaterloo.ca)
Jie Chen (jie.chen@oulu.fi)
Xinwang Liu (xinwangliu@nudt.edu.cn)
Matti Pietikäinen (matti.pietikainen@oulu.fi)
1 National University of Defense Technology, China
2 University of Oulu, Finland
3 University of Sydney, Australia
4 Chinese University of Hong Kong, China
5 University of Waterloo, Canada

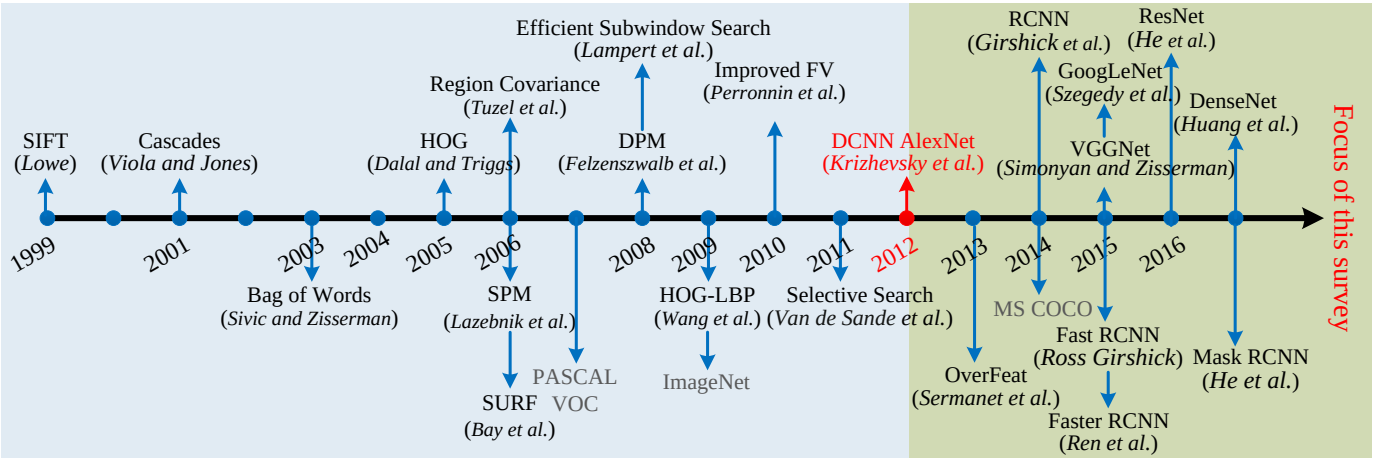


Fig. 2 Milestones of object detection and recognition, including feature representations [37, 42, 79, 109, 114, 139, 140, 166, 191, 194, 200, 213, 215], detection frameworks [56, 65, 183, 209, 213], and datasets [53, 129, 179]. The time period up to 2012 is dominated by handcrafted features. We see a turning point in 2012 with the development of DCNNs for image classification by Krizhevsky *et al.* [109]. Most listed methods are highly cited and won one of the major ICCV or CVPR prizes. See Section 2.3 for details.

cars, bicycles, and dogs). Historically, much of the effort in the field of object detection has focused on the detection of a single category (such as faces and pedestrians) or a few specific categories. In contrast, in the past several years the research community has started moving towards the challenging goal of building general purpose object detection systems whose breadth of object detection ability rivals that of humans.

However in 2012, Krizhevsky *et al.* [109] proposed a Deep Convolutional Neural Network (DCNN) called AlexNet which achieved record breaking image classification accuracy in the Large Scale Visual Recognition Challenge (ILSRVC) [179]. Since that time the research focus in many computer vision application areas has been on deep learning methods. A great many approaches based on deep learning have sprung up in generic object detection [65, 77, 64, 183, 176] and tremendous progress has been achieved, yet we are unaware of comprehensive surveys of the subject during the past five years. Given this time of rapid evolution, the focus of this paper is specifically that of generic object detection by deep learning, in order to gain a clearer panorama in generic object detection.

The generic object detection problem itself is defined as follows: Given an arbitrary image, determine whether there are any instances of semantic objects from predefined categories and, if present, to return the spatial location and extent. Object refers to a material thing that can be seen and touched. Although largely synonymous with object class detection, generic object detection places a greater emphasis on approaches aimed at detecting a broad range of natural categories, as opposed to object instances or specialized categories (*e.g.,* faces, pedestrians, or cars). Generic object detection has received significant attention, as demonstrated by recent progress on object detection competitions such as the PASCAL VOC detection challenge from 2006 to 2012 [53, 54], the ILSVRC large scale detection challenge since 2013 [179], and the MS COCO large scale detection challenge since 2015 [129]. The striking improvement in recent years is illustrated in Fig. 1.

1.1 Comparison with Previous Reviews

A number of notable object detection surveys have been published, as summarized in Table 1. These include many excellent surveys on the problem of *specific* object detection, such as pedestrian detection [51, 60, 48], face detection [226, 232], vehicle detection [196] and text detection [227]. Important contributions were also made by Ponce *et al.* [169], Dickinson [46], Galleguillos and Belongie [59], Grauman and Leibe [69], and Andreopoulos and Tsotsos [5].

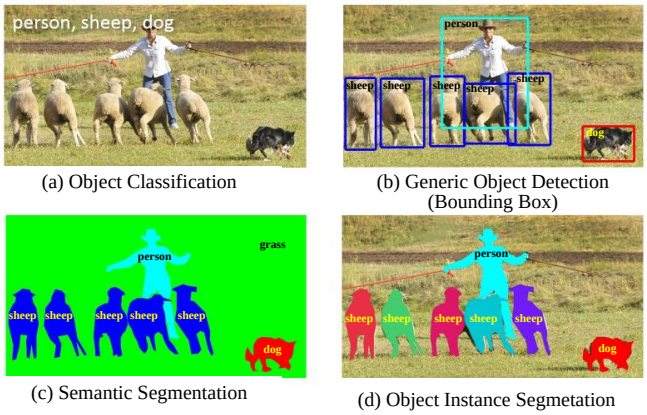
There are few recent surveys focusing directly on the problem of generic object detection, except for the work by Zhang *et al.* [240] who conducted a survey on the topic of object class detection. However, the research reviewed in [69], [5] and [240] is mostly that preceding 2012, and therefore before the more recent striking success of deep learning and related methods.

Deep learning allows computational models consisting of multiple hierarchical layers to learn fantastically complex, subtle, and abstract representations. In the past several years, deep learning has driven significant progress in a broad range of problems, such as visual recognition, object detection, speech recognition, natural language processing, medical image analysis, drug discovery and genomics. Among different types of deep neural networks, Deep Convolutional Neural Networks (DCNN) [115, 109, 116] have brought about breakthroughs in processing images, video, speech and audio. Given this time of rapid evolution, researchers have recently published surveys on different aspects of deep learning, including that of Bengio *et al.* [12], LeCun *et al.* [116], Litjens *et al.* [133], Gu *et al.* [71], and more recently in tutorials at ICCV and CVPR.

Although many deep learning based methods have been proposed for objection detection, we are unaware of comprehensive surveys of the subject during the past five years, the focus of this survey. A thorough review and summarization of existing work is essential for further progress in object detection, particularly for researchers wishing to enter the field. Extensive work on CNNs for specific object detection, such as face detection [120, 237, 92], pedestrian detection [238, 85], vehicle detection [247] and traffic sign detection [253] will not be included in our discussion.

Table 1 Summarization of a number of related surveys since 2000.

No.	Survey Title	Ref.	Year	Published	Content
1	Monocular Pedestrian Detection: Survey and Experiments	[51]	2009	PAMI	Evaluating three detectors with additional experiments integrating the detectors into full systems
2	Survey of Pedestrian Detection for Advanced Driver Assistance Systems	[60]	2010	PAMI	A survey of pedestrian detection for advanced driver assistance systems
3	Pedestrian Detection: An Evaluation of the State of The Art	[48]	2012	PAMI	Focus on a more thorough and detailed evaluation of detectors in individual monocular images
4	Detecting Faces in Images: A Survey	[226]	2002	PAMI	First survey of face detection from a single image
5	A Survey on Face Detection in the Wild: Past, Present and Future	[232]	2015	CVIU	A survey of face detection in the wild since 2000
6	On Road Vehicle Detection: A Review	[196]	2006	PAMI	A review of vision based onroad vehicle detection systems where the camera is mounted on the vehicle
7	Text Detection and Recognition in Imagery: A Survey	[227]	2015	PAMI	A survey of text detection and recognition in color imagery
8	Toward Category Level Object Recognition	[169]	2007	Book	Collects a series of representative papers on object categorization, detection, and segmentation
9	The Evolution of Object Categorization and the Challenge of Image Abstraction	[46]	2009	Book	A trace of the evolution of object categorization in the last four decades
10	Context based Object Categorization: A Critical Survey	[59]	2010	CVIU	A review of different ways of using contextual information for object categorization
11	50 Years of Object Recognition: Directions Forward	[5]	2013	CVIU	A review of the evolution of object recognition systems in the last five decades
12	Visual Object Recognition	[69]	2011	Tutorial	Covers fundamental and time tested approaches for both instance and category object recognition techniques
13	Object Class Detection: A Survey	[240]	2013	ACM CS	First survey of generic object detection methods before 2011
14	Feature Representation for Statistical Learning based Object Detection: A Review	[125]	2015	PR	A survey on feature representation methods in statistical learning based object detection, including handcrafted and a few deep learning based features
15	Salient Object Detection: A Survey	[17]	2014	arXiv	A survey for Salient object detection
16	Representation Learning: A Review and New Perspectives	[12]	2013	PAMI	A review of unsupervised feature learning and deep learning, covering advances in probabilistic models, autoencoders, manifold learning, and deep networks
17	Deep Learning	[116]	2015	Nature	An introduction to deep learning and its typical applications
18	A Survey on Deep Learning in Medical Image Analysis	[133]	2017	MIA	A survey of deep learning for image classification, object detection, segmentation, registration, and others in medical image analysis
19	Recent Advances in Convolutional Neural Networks	[71]	2017	PR	A broad survey of the recent advances in CNN and its applications in computer vision, speech and natural language processing
20	Tutorial: Tools for Efficient Object Detection	–	2015	ICCV15	A short course for object detection only covering recent milestones
21	Tutorial: Deep Learning for Objects and Scenes	–	2017	CVPR17	A high level summary of recent work on deep learning for visual recognition of objects and scenes
22	Tutorial: Instance Level Recognition	–	2017	ICCV17	A short course of recent advances on instance level recognition, including object detection, instance segmentation and human pose prediction
23	Tutorial: Visual Recognition and Beyond	–	2018	CVPR18	This tutorial covers methods and principles behind image classification, object detection, instance segmentation, and semantic segmentation.
24	**Deep Learning for Generic Object Detection**	–	**2018**	**Ours**	**A comprehensive survey of deep learning for generic object detection**



(a) Object Classification (b) Generic Object Detection (Bounding Box) (c) Semantic Segmentation (d) Object Instance Segmetation

Fig. 3 Recognition problems related to generic object detection. (a) Image level object classification, (b) bounding box level generic object detection, (c) pixel-wise semantic segmentation, (d) instance level semantic segmentation.

1.2 Categorization Methodology

The number of papers on generic object detection published since deep learning entering is just breathtaking. So many, in fact, that compiling a comprehensive review of the state of the art already exceeds the possibility of a paper like this one. It is necessary to establish some selection criteria, *e.g.* completeness of a paper and importance to the field. We have preferred to include top journal and conference papers. Due to limitations on space and our knowledge, we sincerely apologize to those authors whose works are not included in this paper. For surveys of efforts in related topics, readers are referred to the articles in Table 1. This survey mainly focuses on the major progress made in the last five years; but for completeness and better readability, some early related works are also included. We restrict ourselves to still pictures and leave video object detection as a separate topic.

The remainder of this paper is organized as follows. Related background, including the problem, key challenges and the progress made during the last two decades are summarized in Section 2. We describe the milestone object detectors in Section 3. Fundamental subproblems and relevant issues involved in designing object detectors are presented in Section 4. A summarization of popular databases and state of the art performance is given in 5. We conclude the paper with a discussion of several promising directions in Section 6.

2 Background

2.1 The Problem

Generic object detection (*i.e.,* generic object category detection), also called *object class detection* [240] or *object category detection*, is defined as follows. Given an image, the goal of generic object detection is to determine whether or not there are instances of objects from *many* predefined categories and, if present, to return the spatial location and extent of each instance. It places greater emphasis on detecting a broad range of natural categories, as opposed to specific object category detection where only a narrower predefined category of interest (*e.g.,* faces, pedestrians, or cars) may be present. Although thousands of objects occupy the visual world in which we live, currently the research community is primarily interested in the localization of highly structured objects (*e.g.,* cars, faces, bicycles and airplanes) and articulated (*e.g.,* humans, cows and horses) rather than unstructured scenes (such as sky, grass and cloud).

Typically, the spatial location and extent of an object can be defined coarsely using a bounding box, *i.e.,* an axis-aligned rectangle tightly bounding the object [53, 179], a precise pixel-wise segmentation mask, or a closed boundary [180, 129], as illustrated in Fig. 3. To our best knowledge, in the current literature, bounding boxes are more widely used for evaluating generic object detection algorithms [53, 179], and will be the approach we adopt in this survey as well. However the community is moving towards deep scene understanding (from image level object classification to single object localization, to generic object detection, and to pixel-wise object segmentation), hence it is anticipated that future challenges will be at the pixel level[129].

There are many problems closely related to that of generic object detection[1]. The goal of *object classification* or *object categorization* (Fig. 3 (a)) is to assess the presence of objects from a given number of object classes in an image; *i.e.,* assigning one or more object class labels to a given image, determining presence without the need of location. It is obvious that the additional requirement to locate the instances in an image makes detection a more challenging task than classification. The *object recognition* problem denotes the more general problem of finding and identifying objects of interest present in an image, subsuming the problems of object detection and object classification [53, 179, 156, 5].

[1] To our best knowledge, there is no universal agreement in the literature on the definitions of various vision subtasks. Often encountered terms such as detection, localization, recognition, classification, categorization, verification and identification, annotation, labeling and understanding are often differently defined [5].

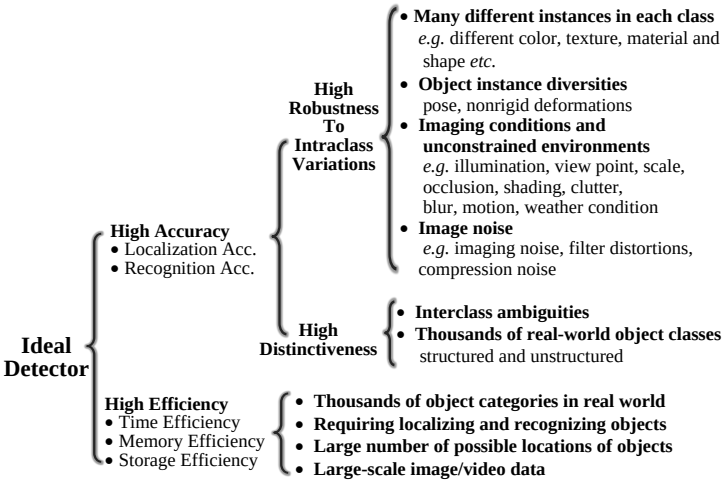


Fig. 4 Summary of challenges in generic object detection.

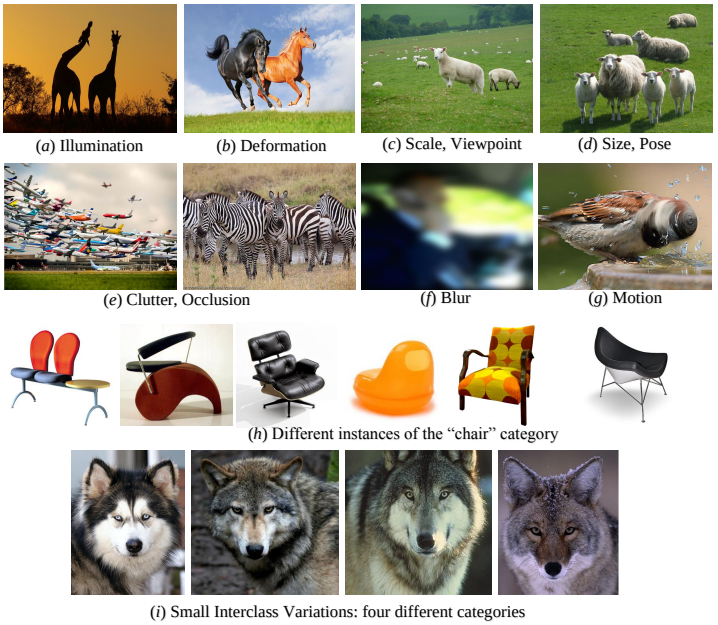


Fig. 5 Changes in imaged appearance of the same class with variations in imaging conditions (a-g). There is an astonishing variation in what is meant to be a single object class (h). In contrast, the four images in (i) appear very similar, but in fact are from four different object classes. Images from ImageNet [179] and MS COCO [129].

Generic object detection is closely related with *semantic image segmentation* (Fig. 3 (c)), which aims to assign each pixel in an image to a semantic class label. *Object instance segmentation* (Fig. 3 (d)) aims at distinguishing different instances of the same object class, while semantic segmentation does not distinguish different instances. *Generic object detection* also distinguishes different instances of the same object. Different from segmentation, object detection includes background region in the bounding box that might be useful for analysis.

2.2 Main Challenges

Generic object detection aims at localizing and recognizing a broad range of natural object categories. The ideal goal of generic object detection is to develop general-purpose object detection algorithms

achieving two competing goals: *high quality/accuracy* and *high efficiency*, as illustrated in Fig. 4. As illustrated in Fig. 5, high quality detection has to accurately localize and recognize objects in images or video frames, such that the large variety of object categories in the real world can be distinguished (*i.e.,* high distinctiveness), and that object instances from the same category, subject to intraclass appearance variations, can be localized and recognized (*i.e.,* high robustness). High efficiency requires the entire detection task to run at a sufficiently high frame rate with acceptable memory and storage usage. Despite several decades of research and significant progress, arguably the combined goals of accuracy and efficiency have not yet been met.

2.2.1 Accuracy related challenges

For accuracy, the challenge stems from 1) the vast range of intraclass variations and 2) the huge number of object categories.

We begin with intraclass variations, which can be divided into two types: intrinsic factors, and imaging conditions. For the former, each object category can have many different object instances, possibly varying in one or more of color, texture, material, shape, and size, such as the "chair" category shown in Fig. 5 (*h*). Even in a more narrowly defined class, such as human or horse, object instances can appear in different poses, with nonrigid deformations and different clothes.

For the latter, the variations are caused by changes in imaging conditions and unconstrained environments which may have dramatic impacts on object appearance. In particular, different instances, or even the same instance, can be captured subject to a wide number of differences: different times, locations, weather conditions, cameras, backgrounds, illuminations, viewpoints, and viewing distances. All of these conditions produce significant variations in object appearance, such as illumination, pose, scale, occlusion, background clutter, shading, blur and motion, with examples illustrated in Fig. 5 (*a*-*g*). Further challenges may be added by digitization artifacts, noise corruption, poor resolution, and filtering distortions.

In addition to *intra*class variations, the large number of object categories, on the order of $10^4 - 10^5$, demands great discrimination power of the detector to distinguish between subtly different *inter*-class variations, as illustrated in Fig. 5 (i)). In practice, current detectors focus mainly on structured object categories, such as the 20, 200 and 91 object classes in PASCAL VOC [53], ILSVRC [179] and MS COCO [129] respectively. Clearly, the number of object categories under consideration in existing benchmark datasets is much smaller than that can be recognized by humans.

2.2.2 Efficiency related challenges

The exponentially increasing number of images calls for efficient and scalable detectors. The prevalence of social media networks and mobile/wearable devices has led to increasing demands for analyzing visual data. However mobile/wearable devices have limited computational capabilities and storage space, in which case an efficient object detector is critical.

For efficiency, the challenges stem from the need to localize and recognize all object instances of very large number of object categories, and the very large number of possible locations and scales within a single image, as shown by the example in Fig. 5 (*c*). A further challenge is that of scalability: A detector should be able to handle unseen objects, unknown situations, and rapidly increasing image data. For example, the scale of ILSVRC [179] is already imposing limits on the manual annotations that are feasible to obtain. As the number of images and the number of categories grow even larger, it may become impossible to annotate them manually, forcing algorithms to rely more on weakly supervised training data.

2.3 Progress in the Past Two Decades

Early research on object recognition was based on template matching techniques and simple part based models [57], focusing on specific objects whose spatial layouts are roughly rigid, such as faces. Before 1990 the leading paradigm of object recognition was based on geometric representations [149, 169], with the focus later moving away from geometry and prior models towards the use of statistical classifiers (such as Neural Networks [178], SVM [159] and Adaboost [213, 222]) based on appearance features[150, 181]. This successful family of object detectors set the stage for most subsequent research in this field.

In the late 1990s and early 2000s object detection research made notable strides. The milestones of object detection in recent years are presented in Fig. 2, in which two main eras (SIFT *vs.* DCNN) are highlighted. The appearance features moved from global representations [151, 197, 205] to local representations that are invariant to changes in translation, scale, rotation, illumination, viewpoint and occlusion. Handcrafted local invariant features gained tremendous popularity, starting from the Scale Invariant Feature Transform (SIFT) feature [139], and the progress on various visual recognition tasks was based substantially on the use of local descriptors [145] such as Haar like features [213], SIFT [140], Shape Contexts [11], Histogram of Gradients (HOG) [42] and Local Binary Patterns (LBP) [153], covariance [206]. These local features are usually aggregated by simple concatenation or feature pooling encoders such as the influential and efficient Bag of Visual Words approach introduced by Sivic and Zisserman [194] and Csurka *et al.* [37], Spatial Pyramid Matching (SPM) of BoW models [114], and Fisher Vectors [166].

For years, the multistage handtuned pipelines of handcrafted local descriptors and discriminative classifiers dominated a variety of domains in computer vision, including object detection, until the significant turning point in 2012 when Deep Convolutional Neural Networks (DCNN) [109] achieved their record breaking results in image classification. The successful application of DCNNs to image classification [109] transferred to object detection, resulting in the milestone Region based CNN (RCNN) detector of Girshick *et al.* [65]. Since then, the field of object detection has dramatically evolved and many deep learning based approaches have been developed, thanks in part to available GPU computing resources and the availability of large scale datasets and challenges such as ImageNet [44, 179] and MS COCO [129]. With these new datasets, researchers can target more realistic and complex problems when detecting objects of hundreds categories from images with large intraclass variations and interclass similarities [129, 179].

The research community has started moving towards the challenging goal of building general purpose object detection systems whose ability to detect many object categories matches that of humans. This is a major challenge: according to cognitive scientists, human beings can identify around 3,000 entry level categories and 30,000 visual categories overall, and the number of categories distinguishable with domain expertise may be on the order of 10^5 [14]. Despite the remarkable progress of the past years, designing an accurate, robust, efficient detection and recognition system that approaches human-level performance on $10^4 - 10^5$ categories is undoubtedly an open problem.

3 Frameworks

There has been steady progress in object feature representations and classifiers for recognition, as evidenced by the dramatic change from handcrafted features [213, 42, 55, 76, 212] to learned DCNN features [65, 160, 64, 175, 40].

In contrast, the basic "sliding window" strategy [42, 56, 55] for localization remains to be the main stream, although with some endeavors in [113, 209]. However the number of windows is large and grows quadratically with the number of pixels, and the need to search over multiple scales and aspect ratios further increases the search space. The the huge search space results in high computational complexity. Therefore, the design of efficient and effective detection framework plays a key role. Commonly adopted strategies include cascading, sharing feature computation, and reducing per-window computation.

In this section, we review the milestone detection frameworks present in generic object detection since deep learning entered the field, as listed in Fig. 6 and summarized in Table 10. Nearly all detectors proposed over the last several years are based on one of these milestone detectors, attempting to improve on one or more aspects. Broadly these detectors can be organized into two main categories:

A. Two stage detection framework, which includes a pre-processing step for region proposal, making the overall pipeline two stage.
B. One stage detection framework, or region proposal free framework, which is a single proposed method which does not separate detection proposal, making the overall pipeline single-stage.

Section 4 will build on the following by discussing fundamental subproblems involved in the detection framework in greater detail, including DCNN features, detection proposals, context modeling, bounding box regression and class imbalance handling.

3.1 Region Based (Two Stage Framework)

In a region based framework, category-independent region proposals are generated from an image, CNN [109] features are extracted from these regions, and then category-specific classifiers are used to determine the category labels of the proposals. As can be observed from Fig. 6, DetectorNet [198], OverFeat [183], MultiBox [52] and RCNN [65] independently and almost simultaneously proposed using CNNs for generic object detection.

RCNN: Inspired by the breakthrough image classification results obtained by CNN and the success of selective search in region proposal for hand-crafted features [209], Girshick *et al.* were among the first to explore CNN for generic object detection and developed RCNN [65, 67], which integrates AlexNet [109] with the region proposal method selective search [209]. As illustrated in Fig. 7, training in an RCNN framework consists of multistage pipelines:

1. Class-agnostic region proposals, which are candidate regions that might contain objects, are obtained selective search [209];
2. Region proposals, which are cropped from the image and warped into the same size, are used as the input for finetuning a CNN model pre-trained using large-scale dataset such as ImageNet;
3. A set of class specific linear SVM classifiers are trained using fixed length features extracted with CNN, replacing the softmax classifier learned by finetuning.
4. Bounding box regression is learned for each object class with CNN features.

In spite of achieving high object detection quality, RCNN has notable drawbacks [64]:

1. Training is a multistage complex pipeline, which is inelegant, slow and hard to optimize because each individual stage must be trained separately.
2. Numerous region proposals which provide only rough localization need to be externally detected.
3. Training SVM classifiers and bounding box regression is expensive in both disk space and time, since CNN features are extracted independently from each region proposal in each image, posing great challenges for large-scale detection, especially very deep CNN networks such as AlexNet [109] and VGG [191].
4. Testing is slow, since CNN features are extracted per object proposal in each testing image.

SPPNet: During testing, CNN features extraction is the main bottleneck of the RCNN detection pipeline, which requires to extract CNN features from thousands of warped region proposals for an image. Noticing these obvious disadvantages, He *et al.* [77] introduced the traditional spatial pyramid pooling (SPP) [68, 114] into CNN architectures. Since convolutional layers accept inputs of arbitrary sizes, the requirement of fixed-sized images in CNNs is only due to the Fully Connected (FC) layers, He *et al.* found this fact and added an SPP layer on top of the last convolutional (CONV) layer to obtain features of fixed-length for the FC layers. With this SPPnet, RCNN obtains a significant speedup without sacrificing any detection quality because it only needs to run the convolutional layers once on the entire test image to generate fixed-length features for region proposals of arbitrary size. While SPPnet accelerates RCNN evaluation by orders of magnitude, it does not result in a comparable speedup of the detector training. Moreover, finetuning in SPPnet [77] is unable to update the convolutional layers before the SPP layer, which limits the accuracy of very deep networks.

Fast RCNN: Girshick [64] proposed Fast RCNN that addresses some of the disadvantages of RCNN and SPPnet, while improving on their detection speed and quality. As illustrated in Fig. 8, Fast RCNN enables end-to-end detector training (when ignoring

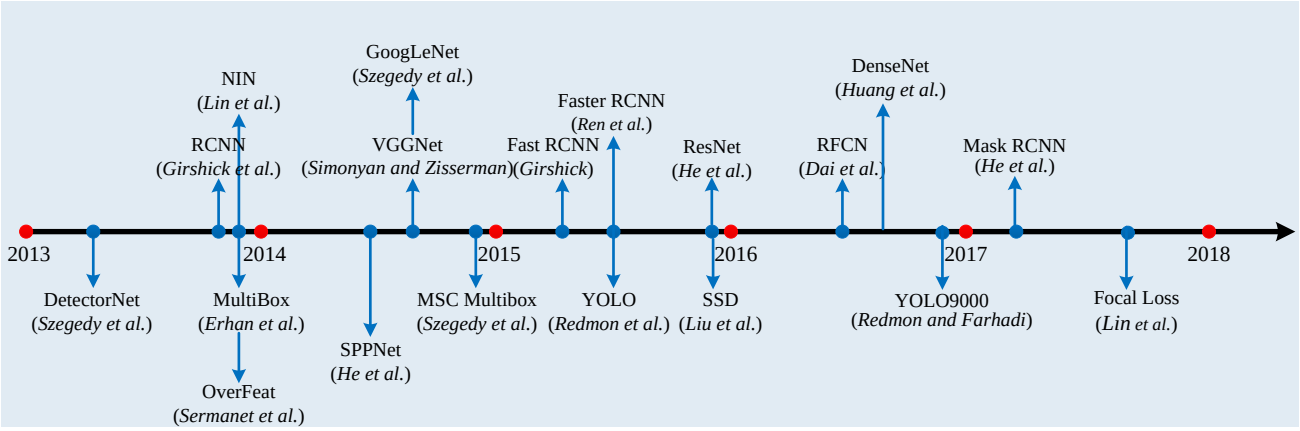


Fig. 6 Milestones in generic object detection based on the point in time of the first arXiv version.

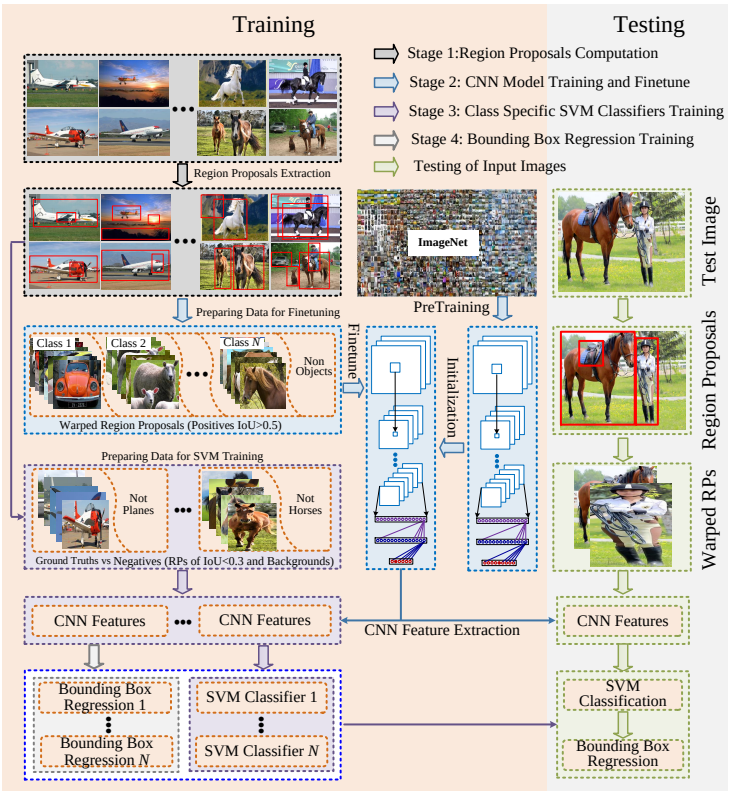


Fig. 7 Illustration of the milestone detecting framework RCNN [65, 67] in great detail.

the process of region proposal generation) by developing a streamlined training process that simultaneously learns a softmax classifier and class-specific bounding box regression using a multitask loss, rather than training a softmax classifier, SVMs, and BBRs in three separate stages as in RCNN/SPPnet. Fast RCNN employs the idea of sharing the computation of convolution across region proposals, and adds a Region of Interest (RoI) pooling layer between the last CONV layer and the first FC layer to extract a fixed-length feature for each region proposal (*i.e.* RoI). Essentially, RoI pooling uses warping at feature level for approximating warping at image level. The features after the RoI pooling layer are fed into a sequence of FC layers that finally branch into two sibling output layers: softmax probabilities for object category prediction and class-specific bounding box regression offsets for proposal refinement. Compared to RCNN/SPPnet, Fast RCNN improves the efficiency considerably – typically 3 times faster in training and 10 times faster in testing. In summary, Fast RCNN has attractive advantages of higher detection quality, a single-stage training process that updates all network layers, and no storage required for feature caching.

Faster RCNN [175, 176]: Although Fast RCNN significantly sped up the detection process, it still relies on external region proposals. Region proposal computation is exposed as the new bottleneck in Fast RCNN. Recent work has shown that CNNs have a remarkable ability to localize objects in CONV layers [243, 244, 36, 158, 75], an ability which is weakened in the FC layers. Therefore, the selective search can be replaced by the CNN in producing region proposals. The Faster RCNN framework proposed by Ren *et al.* [175, 176] proposed an efficient and accurate Region Proposal Network (RPN) to generating region proposals. They utilize single network to accomplish the task of RPN for region proposal and Fast RCNN for region classification. In Faster RCNN, the RPN and fast RCNN share large number of convolutional layers. The features from the last shared convolutional layer are used for region proposal and region classification from separate branches. RPN first initializes k $n \times n$ reference boxes (*i.e.* the so called *anchors*) of different scales and aspect ratios at each CONV feature map location. Each $n \times n$ anchor is mapped to a lower dimensional vector (such as 256 for ZF and 512 for VGG), which is fed into two sibling FC layers — an object category classification layer and a box regression layer. Different from Fast RCNN, the features used for regression in RPN have the same size. RPN shares CONV features with Fast RCNN, thus enabling highly efficient region proposal computation. RPN is, in fact, a kind of Fully Convolutional Network (FCN) [138, 185]; Faster RCNN is thus a purely CNN based framework without using handcrafted features. For the very deep VGG16 model [191], Faster RCNN can test at 5fps (including all steps) on a GPU, while achieving state of the art object detection accuracy on PASCAL VOC 2007 using 300 proposals per image. The initial Faster RCNN in [175] contains several alternating training steps. This was then simplified by one step joint training in [176].

Concurrent with the development of Faster RCNN, Lenc and Vedaldi [117] challenged the role of region proposal generation methods such as selective search, studied the role of region proposal generation in CNN based detectors, and found that CNNs contain sufficient geometric information for accurate object detec-

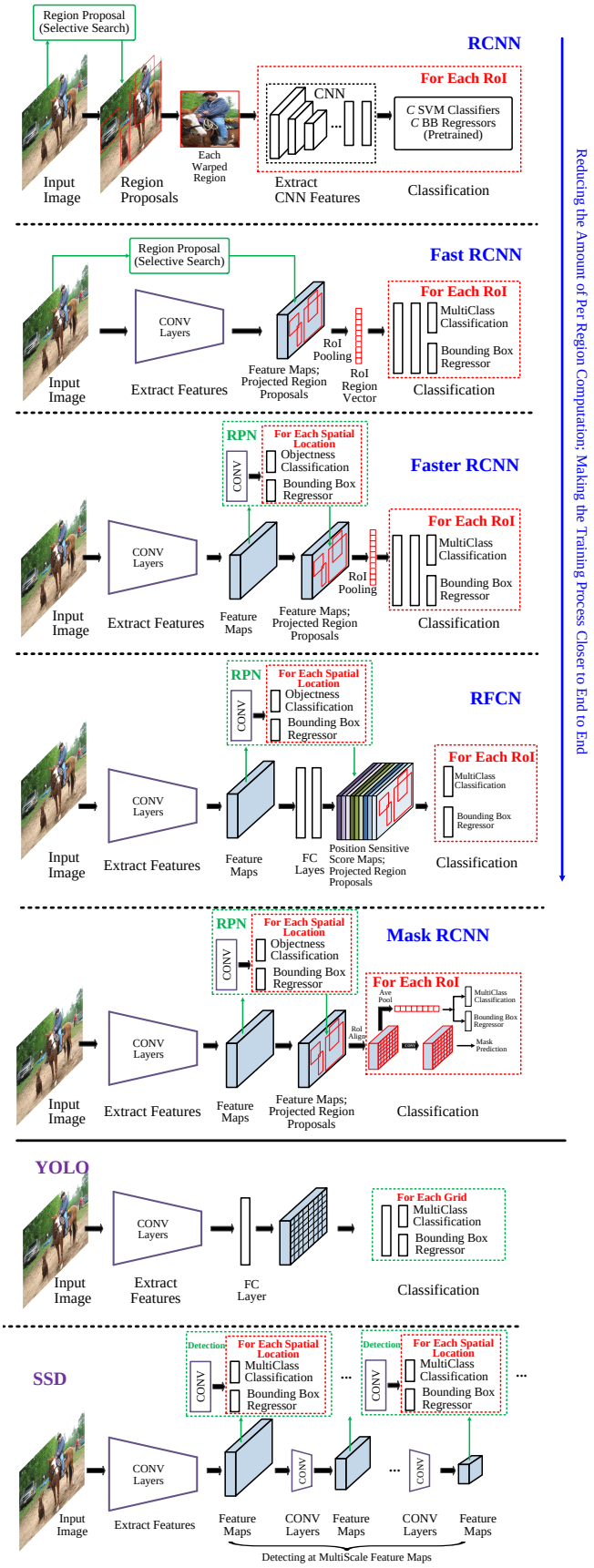


Fig. 8 High level diagrams of the leading frameworks for generic object detection. The properties of these methods are summarized in Table 10.

tion in the CONV rather than FC layers. They proved the possibility of building integrated, simpler, and faster object detectors that rely exclusively on CNNs, removing region proposal generation methods such as selective search.

RFCN (Region based Fully Convolutional Network): While Faster RCNN is an order of magnitude faster than Fast RCNN, the fact that the region-wise subnetwork still needs to be applied per RoI (several hundred RoIs per image) led Dai *et al.* [40] to propose the RFCN detector which is *fully convolutional* (no hidden FC layers) with almost all computation shared over the entire image. As shown in Fig. 8, RFCN differs from Faster RCNN only in the RoI subnetwork. In Faster RCNN, the computation after the RoI pooling layer cannot be shared. A natural idea is to minimize the amount of computation that cannot be shared, hence Dai *et al.* [40] proposed to use all CONV layers to construct a shared RoI subnetwork and RoI crops are taken from the last layer of CONV features prior to prediction. However, Dai *et al.* [40] found that this naive design turns out to have considerably inferior detection accuracy, conjectured to be that deeper CONV layers are more sensitive to category semantic and less sensitive to translation, whereas object detection needs localization representations that respect translation variance. Based on this observation, Dai *et al.* [40] constructed a set of position sensitive score maps by using a bank of specialized CONV layers as the FCN output, on top of which a position sensitive RoI pooling layer different from the more standard RoI pooling in [64, 175] is added. They showed that the RFCN with ResNet101 [79] could achieve comparable accuracy to Faster RCNN, often at faster running times.

Mask RCNN: Following the spirit of conceptual simplicity, efficiency, and flexibility, He *et al.* [80] proposed Mask RCNN to tackle pixel-wise object instance segmentation by extending Faster RCNN. Mask RCNN adopts the same two stage pipeline, with an identical first stage (RPN). In the second stage, in parallel to predicting the class and box offset, Mask RCNN adds a branch which outputs a binary mask for each RoI. The new branch is a Fully Convolutional Network (FCN) [138, 185] on top of a CNN feature map. In order to avoid the misalignments caused by the original RoI pooling (RoIPool) layer, a RoIAlign layer was proposed to preserve the pixel level spatial correspondence. With a backbone network ResNeXt101-FPN [223, 130], Mask RCNN achieved top results for the COCO object instance segmentation and bounding box object detection. It is simple to train, generalizes well, and adds only a small overhead to Faster RCNN, running at 5 FPS [80].

Light Head RCNN: In order to further speed up the detection speed of RFCN [40], Li *et al.* [128] proposed Light Head RCNN, making the head of the detection network as light as possible to reduce the RoI regionwise computation. In particular, Li *et al.* [128] applied a large kernel separable convolution to produce thin feature maps with small channel number and a cheap RCNN subnetwork, leading to an excellent tradeoff of speed and accuracy.

3.2 Unified Pipeline (One Stage Pipeline)

The region-based pipeline strategies of Section 3.1 have prevailed on detection benchmarks since RCNN [65]. The significant efforts introduced in Section 3.1 have led to faster and more accurate detectors, and the current leading results on popular benchmark datasets are all based on Faster RCNN [175]. In spite of

that progress, region-based approaches could be computationally expensive for mobile/wearable devices, which have limited storage and computational capability. Therefore, instead of trying to optimize the individual components of a complex region-based pipeline, researchers have begun to develop *unified* detection strategies.

Unified pipelines refer broadly to architectures that directly predict class probabilities and bounding box offsets from full images with a single feed forward CNN network in a monolithic setting that does not involve region proposal generation or post classification. The approach is simple and elegant because it completely eliminates region proposal generation and subsequent pixel or feature resampling stages, encapsulating all computation in a single network. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance.

DetectorNet: Szegedy *et al.* [198] were among the first to explore CNNs for object detection. DetectorNet formulated object detection a regression problem to object bounding box masks. They use AlexNet [109] and replace the final softmax classifier layer by a regression layer. Given an image window, they use one network to predict foreground pixels over a coarse grid, as well as four additional networks to predict the object's top, bottom, left and right halves. A grouping process then converts the predicted masks into detected bounding boxes. One needs to train a network per object type and mask type. It does not scale up to multiple classes. DetectorNet must take many crops of the image, and run multiple networks for each part on every crop.

OverFeat, proposed by Sermanet *et al.* [183], was one of the first modern one-stage object detectors based on fully convolutional deep networks. It is one of the most successful object detection frameworks, winning the ILSVRC2013 localization competition. OverFeat performs object detection in a multiscale sliding window fashion via a single forward pass through the CNN network, which (with the exception of the final classification/regressor layer) consists only of convolutional layers. In this way, they naturally share computation between overlapping regions. OverFeat produces a grid of feature vectors, each of which represents a slightly different context view location within the input image and can predict the presence of an object. Once an object is identified, the same features are then used to predict a single bounding box regressor. In addition, OverFeat leverages multiscale features to improve the overall performance by passing up to six enlarged scales of the original image through the network and iteratively aggregating them together, resulting in a significantly increased number of evaluated context views (final feature vectors). OverFeat has a significant speed advantage over RCNN [65], which was proposed during the same period, but is significantly less accurate because it is hard to train fully convolutional network at that stage. The speed advantage derives from sharing the computation of convolution between overlapping windows using fully convolutional network.

YOLO (You Only Look Once): Redmon *et al.* [174] proposed YOLO, a unified detector casting object detection as a regression problem from image pixels to spatially separated bounding boxes and associated class probabilities. The design of YOLO is illustrated in Fig. 8. Since the region proposal generation stage is completely dropped, YOLO directly predicts detections using a small set of candidate regions. Unlike region-based approaches, e.g. Faster RCNN, that predict detections based on features from local region, YOLO uses the features from entire image globally. In particular, YOLO divides an image into a $S \times S$ grid. Each grid predicts C class probabilities, B bounding box locations and confidences scores for those boxes. These predictions are encoded as an $S \times S \times (5B+C)$ tensor. By throwing out the region proposal generation step entirely, YOLO is fast by design, running in real time at 45 FPS and a fast version, *i.e.* Fast YOLO [174], running at 155 FPS. Since YOLO sees the entire image when making predictions, it implicitly encodes contextual information about object classes and is less likely to predict false positives on background. YOLO makes more localization errors resulting from the coarse division of bounding box location, scale and aspect ratio. As discussed in [174], YOLO may fail to localize some objects, especially small ones, possibly because the grid division is quite coarse, and because by construction each grid cell can only contain one object. It is unclear to what extent YOLO can translate to good performance on datasets with significantly more objects, such as the ILSVRC detection challenge.

YOLOv2 and YOLO9000: Redmon and Farhadi [173] proposed YOLOv2, an improved version of YOLO, in which the custom GoogLeNet [200] network is replaced with a simpler DarkNet19, plus utilizing a number of strategies drawn from existing work, such as batch normalization [78], removing the fully connected layers, and using good anchor boxes learned with *k*means and multiscale training. YOLOv2 achieved state of the art on standard detection tasks, like PASCAL VOC and MS COCO. In addition, Redmon and Farhadi [173] introduced YOLO9000, which can detect over 9000 object categories in real time by proposing a joint optimization method to train simultaneously on ImageNet and COCO with WordTree to combine data from multiple sources.

SSD (Single Shot Detector): In order to preserve real-time speed without sacrificing too much detection accuracy, Liu *et al.* [136] proposed SSD, which is faster than YOLO [174] and has accuracy competitive with state-of-the-art region-based detectors, including Faster RCNN [175]. SSD effectively combines ideas from RPN in Faster RCNN [175], YOLO [174] and multiscale CONV features [75] to achieve fast detection speed while still retaining high detection quality. Like YOLO, SSD predicts a fixed number of bounding boxes and scores for the presence of object class instances in these boxes, followed by an NMS step to produce the final detection. The CNN network in SSD is fully convolutional, whose early layers are based on a standard architecture, such as VGG [191] (truncated before any classification layers), which is referred as the base network. Then several auxiliary CONV layers, progressively decreasing in size, are added to the end of the base network. The information in the last layer with low resolution may be too coarse spatially to allow precise localization. SSD uses shallower layers with higher resolution for detecting small objects. For objects of different sizes, SSD performs detection over multiple scales by operating on multiple CONV feature maps, each of which predicts category scores and box offsets for bounding boxes of appropriate sizes. For a 300×300 input, SSD achieves 74.3% mAP on the VOC2007 test at 59 FPS on a Nvidia Titan X.

4 Fundamental SubProblems

In this section important subproblems are described, including feature representation, region proposal, context information mining, and training strategies. Each approach is reviewed with respect to its primary contribution.

4.1 DCNN based Object Representation

As one of the main components in any detector, good feature representations are of primary importance in object detection [46, 65, 62, 249]. In the past, a great deal of effort was devoted to designing local descriptors (*e.g.,* SIFT [139] and HOG [42]) and to explore approaches (*e.g.,* Bag of Words [194] and Fisher Vector [166]) to group and abstract the descriptors into higher level representations in order to allow the discriminative object parts to begin to emerge, however these feature representation methods required careful engineering and considerable domain expertise.

In contrast, deep learning methods (especially *deep* CNNs, or DCNNs), which are composed of multiple processing layers, can learn powerful feature representations with multiple levels of abstraction directly from raw images [12, 116]. As the learning procedure reduces the dependency of specific domain knowledge and complex procedures needed in traditional feature engineering [12, 116], the burden for feature representation has been transferred to the design of better network architectures.

The leading frameworks reviewed in Section 3 (RCNN [65], Fast RCNN [64], Faster RCNN [175], YOLO [174], SSD [136]) have persistently promoted detection accuracy and speed. It is generally accepted that the CNN representation plays a crucial role and it is the CNN architecture which is the engine of a detector. As a result, most of the recent improvements in detection accuracy have been achieved via research into the development of novel networks. Therefore we begin by reviewing popular CNN architectures used in Generic Object Detection, followed by a review of the effort devoted to improving object feature representations, such as developing invariant features to accommodate geometric variations in object scale, pose, viewpoint, part deformation and performing multiscale analysis to improve object detection over a wide range of scales.

4.1.1 Popular CNN Architectures

CNN architectures serve as network backbones to be used in the detection frameworks described in Section 3. Representative frameworks include AlexNet [110], ZFNet [234] VGGNet [191], GoogLeNet [200], Inception series [99, 201, 202], ResNet [79], DenseNet [94] and SENet [91], which are summarized in Table 2, and where the network improvement in object recognition can be seen from Fig. 9. A further review of recent CNN advances can be found in [71].

Briefly, a CNN has a hierarchical structure and is composed of a number of layers such as convolution, nonlinearity, pooling *etc*. From finer to coarser layers, the image repeatedly undergoes filtered convolution, and with each layer the receptive field (region of support) of these filters increases. For example, the pioneering AlexNet [110] has five convolutional layers and two Fully Connected (FC) layers, and where the first layer contains 96 filters of size $11 \times 11 \times 3$. In general, the first CNN layer extracts low level features (*e.g.* edges), intermediate layers extract features of increasing complexity, such as combinations of low level features, and later convolutional layers detect objects as combinations of earlier parts [234, 12, 116, 157].

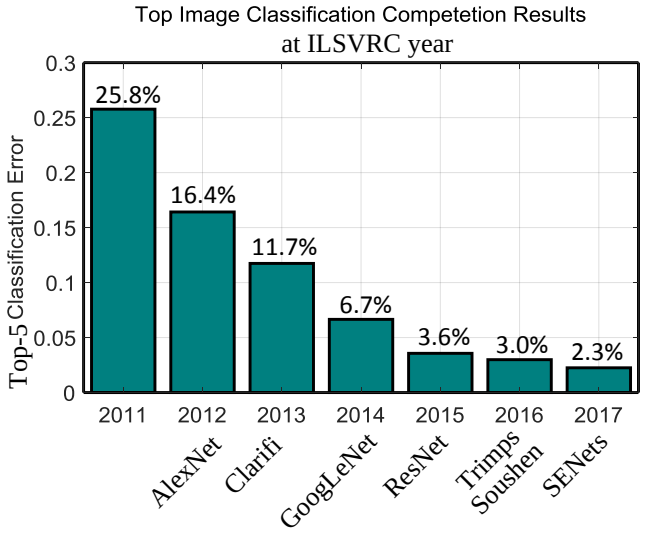


Fig. 9 Performance of winning entries in the ILSVRC competitions from 2011 to 2017 in the image classification task.

As can be observed from Table 2, the trend in architecture evolution is that networks are getting deeper: AlexNet consisted of 8 layers, VGGNet 16 layers, and more recently ResNet and DenseNet both surpassed the 100 layer mark, and it was VGGNet [191] and GoogLeNet [200], in particular, which showed that increasing depth can improve the representational power of deep networks. Interestingly, as can be observed from Table 2, networks such as AlexNet, OverFeat, ZFNet and VGGNet have an enormous number of parameters, despite being only few layers deep, since a large fraction of the parameters come from the FC layers. Therefore, newer networks like Inception, ResNet, and DenseNet, although having a very great network depth, have far fewer parameters by avoiding the use of FC layers.

With the use of Inception modules in carefully designed topologies, the parameters of GoogLeNet is dramatically reduced. Similarly ResNet demonstrated the effectiveness of skip connections for learning extremely deep networks with hundreds of layers, winning the ILSVRC 2015 classification task. Inspired by ResNet [79], InceptionResNets [202] combine the Inception networks with shortcut connections, claiming that shortcut connections can significantly accelerate the training of Inception networks. Extending ResNets, Huang *et al.* [94] proposed DenseNets which are built from dense blocks, where dense blocks connect each layer to every other layer in a feed-forward fashion, leading to compelling advantages such as parameter efficiency, implicit deep supervision, and feature reuse. Recently, Hu *et al.* [79] proposed an architectural unit termed the Squeeze and Excitation (SE) block which can be combined with existing deep architectures to boost their performance at minimal additional computational cost, by adaptively recalibrating channelwise feature responses by explicitly modeling the interdependencies between convolutional feature channels, leading to winning the ILSVRC 2017 classification task. Research

Table 2 DCNN architectures that are commonly used for generic object detection. Regarding the statistics for "#Paras" and "#Layers", we didn't consider the final FC prediction layer. "Test Error" column indicates the Top 5 classification test error on ImageNet1000. Explanations: OverFeat (accurate model), DenseNet201 (Growth Rate 32, DenseNet-BC), and ResNeXt50 (32*4d).

No.	DCNN Architecture	#Paras ($\times 10^6$)	#Layers (CONV+FC)	Test Error (Top 5)	First Used In	Highlights
1	AlexNet [110]	57	$5+2$	15.3%	[65]	The first DCNN; The historical turning point of feature representation from traditional to CNN; In the classification task of ILSVRC2012 competition, achieved a winning Top 5 test error rate of 15.3%, compared to 26.2% given by the second best entry.
2	OverFeat [183]	140	$6+2$	13.6%	[183]	Similar to AlexNet, differences including a smaller stride for CONV1 and 2, different filter size for some layers, more filters for some layers.
3	ZFNet (fast) [234]	58	$5+2$	14.8%	[77]	Highly similar to AlexNet, with a smaller filter size in CONV1 and a smaller stride for CONV1 and 2.
4	VGGNet16 [191]	134	$13+2$	6.8%	[64]	Increasing network depth significantly with small 3×3 convolution filters; Significantly better performance.
5	GoogLeNet [200]	6	22	6.7%	[200]	With the use of Inception module which concatenates feature maps produced by filters of different sizes, the network goes wider and parameters are much less than those of AlexNet *etc*.
6	Inception v2 [99]	12	31	4.8%	[88]	Faster training with the introduce of Batch Normalization.
7	Inception v3 [201]	22	47	3.6%		Going deeper with Inception building blocks in efficient ways.
8	YOLONet [174]	64	$24+1$	–	[174]	A network inspired by GoogLeNet used in YOLO detector.
9	ResNet50 [79]	23.4	49	3.6% (ResNets)	[79]	With the use of residual connections, substantially deeper but with fewer parameters than previous DCNNs (except for GoogLeNet).
10	ResNet101 [79]	42	100		[79]	
11	InceptionResNet v1 [202]	21	87	3.1% (Ensemble)		A residual version of Inception with similar computational cost of Inception v3, but with faster training process.
12	InceptionResNet v2 [202]	30	95		[96]	A costlier residual version of Inception, with significantly improved recognition performance.
13	Inception v4 [202]	41	75			A Inception variant without residual connections with roughly the same recognition performance as InceptionResNet v2, but significantly slower.
14	ResNeXt50 [223]	23	49	3.0%	[223]	Repeating a building block that aggregates a set of transformations with the same topology.
15	DenseNet201 [94]	18	200	–	[246]	Design dense block, which connects each layer to every other layer in a feed forward fashion; Alleviate the vanishing gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters.
16	DarkNet [173]	20	19	–	[173]	Similar to VGGNet, but with significantly less parameters due to the use of fewer filters at each layer.
17	MobileNet [88]	3.2	$27+1$	–	[88]	Light weight deep CNNs using depthwise separable convolutions for mobile applications.
18	SE ResNet50 [91]	26	50	2.3% (SENets)	[91]	Proposing a novel block called *Squeeze and Excitation* to model feature channel relationship; Can be flexibly used in all existing CNNs to improve recognition performance at minimal additional computational cost.

on CNN architectures remain active, and a numer of backbone networks are still emerging such as Dilated Residual Networks [230], Xception [35], DetNet [127], and Dual Path Networks (DPN) [31].

The training of a CNN requires a large labelled dataset with sufficient label and intraclass diversity. Unlike image classification, detection requires localizing (possibly many) objects from an image. It has been shown [161] that pretraining the deep model with a large scale dataset having object-level annotations (such as the ImageNet classification and localization dataset), instead of only image-level annotations, improves the detection performance. However collecting bounding box labels is expensive, especially for hundreds of thousands of categories. A common scenario is for a CNN to be pretrained on a large dataset (usually with a large number of visual categories) with image-level labels; the pretrained CNN can then be applied to a small dataset, directly, as a generic feature extractor [172, 8, 49, 228], which can support a wider range of visual recognition tasks. For detection, the pretrained network is typically finetuned[2] on a given detection dataset [49, 65, 67]. Several large scale image classification datasets are used for CNN pretraining; among them the ImageNet1000 dataset [44, 179] with 1.2 million images of 1000 object categories, or the Places dataset [245] which is much larger than ImageNet1000 but has fewer classes, or a recent hybrid dataset [245] combining the Places and ImageNet datasets.

Pretrained CNNs without finetuning were explored for object classification and detection in [49, 67, 1], where it was shown that features performance is a function of the extracted layer; for example, for AlexNet pretrained on ImageNet, FC6 / FC7 / Pool5 are in descending order of detection accuracy [49, 67]; finetuning a pretrained network can increase detection performance significantly [65, 67], although in the case of AlexNet the finetuning performance boost was shown to be much larger for FC6 and FC7 than for Pool5, suggesting that the Pool5 features are more general. Furthermore the relationship or similarity between the source

[2] Finetuning is done by initializing a network with weights optimized for a large labeled dataset like ImageNet and then updating the network's weights using the target-task training set.

and target datasets plays a critical role, for example that ImageNet based CNN features show better performance [243] on object related image datasets.

4.1.2 Methods For Improving Object Representation

Deep CNN based detectors such as RCNN [65], Fast RCNN [64], Faster RCNN [175] and YOLO [174], typically use the deep CNN architectures listed in 2 as the backbone network and use features from the top layer of the CNN as object representation, however detecting objects across a large *range* of scales is a fundamental challenge. A classical strategy to address this issue is to run the detector over a number of scaled input images (*e.g.,* an image pyramid) [56, 65, 77], which typically produces more accurate detection, however with obvious limitations of inference time and memory. In contrast, a CNN computes its feature hierarchy layer by layer, and the subsampling layers in the feature hierarchy lead to an inherent multiscale pyramid.

This inherent feature hierarchy produces feature maps of different spatial resolutions, but have inherent problems in structure [75, 138, 190]: the later (or higher) layers have a large receptive field and strong semantics, and are the most robust to variations such as object pose, illumination and part deformation, but the resolution is low and the geometric details are lost. On the contrary, the earlier (or lower) layers have a small receptive field and rich geometric details, but the resolution is high and is much less sensitive to semantics. Intuitively, semantic concepts of objects can emerge in different layers, depending on the size of the objects. So if a target object is small it requires fine detail information in earlier layers and may very well disappear at later layers, in principle making small object detection very challenging, for which tricks such as dilated convolutions [229] or atrous convolution [40, 27] have been proposed. On the other hand if the target object is large then the semantic concept will emerge in much later layers. Clearly it is not optimal to predict objects of different scales with features from only one layer, therefore a number of methods [190, 241, 130, 104] have been proposed to improve detection accuracy by exploiting multiple CNN layers, broadly falling into three types of **multiscale object detection**:

1. Detecting with combined features of multiple CNN layers [75, 103, 10];
2. Detecting at multiple CNN layers;
3. Combinations of the above two methods [58, 130, 190, 104, 246, 239].

(1) Detecting with combined features of multiple CNN layers seeks to combine features from multiple layers before making a prediction. Representative approaches include Hypercolumns [75], HyperNet [103], and ION [10]. Such feature combining is commonly accomplished via skip connections, a classic neural network idea that skips some layers in the network and feeds the output of an earlier layer as the input to a later layer, architectures which have recently become popular for semantic segmentation [138, 185, 75]. As shown in Fig. 10 (a), ION [10] uses skip pooling to extract RoI features from multiple layers, and then the object proposals generated by selective search and edgeboxes are classified by using the combined features. HyperNet [103], as shown in Fig. 10 (b), follows a similar idea and integrates deep, intermediate and shallow features to generate object proposals and predict objects via an end to end joint training strategy. This method extracts only 100 candidate regions in each image. The combined feature is more descriptive and is more beneficial for localization and classification, but at increased computational complexity.

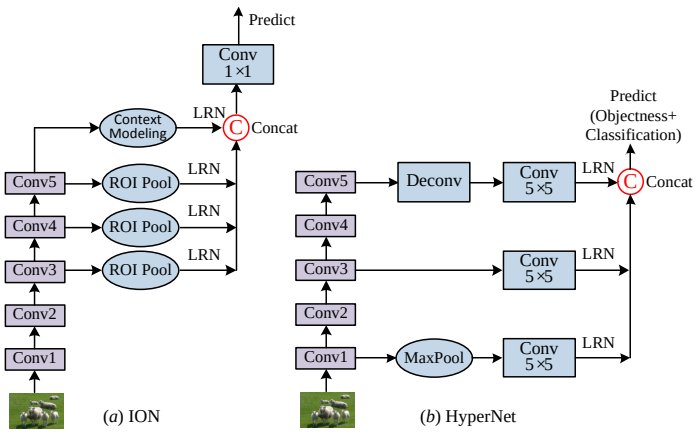


Fig. 10 Comparison of HyperNet and ION. LRN: Local Response Normalization

(2) Detecting at multiple CNN layers [138, 185] combines coarse to fine predictions from multiple layers by averaging segmentation probabilities. SSD [136] and MSCNN [20], RBFNet [135], and DSOD [186] combine predictions from multiple feature maps to handle objects of various sizes. SSD spreads out default boxes of different scales to multiple layers within a CNN and enforces each layer to focus on predicting objects of a certain scale. Liu *et al.* [135] proposed RFBNet which simply replaces the later convolution layers of SSD with a Receptive Field Block (RFB) to enhance the discriminability and robustness of features. The RFB is a multibranch convolutional block, similar to the Inception block [200], but combining multiple branches with different kernels and convolution layers [27]. MSCNN [20] applies deconvolution on multiple layers of a CNN to increase feature map resolution before using the layers to learn region proposals and pool features.

(3) Combination of the above two methods recognizes that, on the one hand, the utility of the hyper feature representation by simply incorporating skip features into detection like UNet [154], Hypercolumns [75], HyperNet [103] and ION [10] does not yield significant improvements due to the high dimensionality. On the other hand, it is natural to detect large objects from later layers with large receptive fields and to use earlier layers with small receptive fields to detect small objects; however, simply detecting objects from earlier layers may result in low performance because earlier layers possess less semantic information. Therefore, in order to combine the best of both worlds, some recent works propose to detect objects at multiple layers, and the feature of each detection layer is obtained by combining features from different layers. Representative methods include SharpMask [168], Deconvolutional Single Shot Detector (DSSD) [58], Feature Pyramid Network (FPN) [130], Top Down Modulation (TDM)[190], Reverse connection with Objectness prior Network (RON) [104], ZIP [122] (shown in Fig. 12), Scale Transfer Detection Network (STDN) [246], RefineDet [239] and StairNet [217], as shown in Table 3 and contrasted in Fig. 11.

Table 3 Summarization of properties of representative methods in improving DCNN feature representations for generic object detection. See Section 4.1.2 for more detail discussion. Abbreviations: Selective Search (SS), EdgeBoxes (EB), InceptionResNet (IRN). Detection results on VOC07, VOC12 and COCO were reported with mAP@IoU=0.5, and the other column results on COCO were reported with a new metric mAP@IoU=[0.5 : 0.05 : 0.95] which averages mAP over different IoU thresholds from 0.5 to 0.95 (written as [0.5:0.95]). Training data: "07"←VOC2007 trainval; "12"←VOC2012 trainval; "07+12"←union of 07 and VOC12 trainval; "07++12"←union of VOC07 trainval, VOC07 test, and VOC12 trainval; 07++12+CO←union of VOC07 trainval, VOC07 test, VOC12 trainval and COCO trainval. The COCO detection results were reported with COCO2015 Test-Dev, except for MPN [233] which reported with COCO2015 Test-Standard.

Group	Detector Name	Region Proposal	Backbone DCNN	Pipelined Used	mAP@IoU=0.5 VOC07	mAP@IoU=0.5 VOC12	mAP@IoU=0.5 COCO	mAP COCO	Published In	Full Name of Detector and Highlights
(1) Single detection with multilayer features	ION [10]	SS+EB MCG+RPN	VGG16	Fast RCNN	79.4 (07+12)	76.4 (07+12)	55.7	33.1	CVPR16	**(Inside Outside Network, ION)**; Use skip layer pooling to extract information at multiple scales; Features pooled from multilayers are normalized, concatenated, scaled, and dimension reduced; Won the Best Student Entry and 3rd overall in the 2015 MS COCO detection challenge.
	HyperNet [103]	RPN	VGG16	Faster RCNN	76.3 (07+12)	71.4 (07++12)	–	–	CVPR16	A good variant of Faster RCNN; Combine deep, intermediate and shallow layer features and compress them into a hyper feature; The hyper feature is used for both RPN and detection network.
	PVANet [102]	RPN	PVANet	Faster RCNN	84.9 (07+12+CO)	84.2 (07++12+CO)	–	–	NIPSW16	A newly designed deep but lightweight network with the principle "less channels with more layers"; Combine ideas from concatenated ReLU [184], Inception [200], and HyperNet [103].
(2) Detection at multiple layers	SDP+CRC [225]	EB	VGG16	Fast RCNN	69.4 (07)	–	–	–	CVPR16	**(Scale Dependent Pooling + Cascade Rejection Classifier, SDP+CRC)**; Utilize features in all CONV layers to reject easy negatives via CRC and then classify survived proposals using SDP which represents an object proposal with the convolutional features extracted from a layer corresponding to its scale.
	MSCNN [20]	RPN	VGG	Faster RCNN	Only Tested on KITTI				ECCV16	**(MultiScale CNN, MSCNN)**; Both proposal generation and detection are performed at multiple output layers; Propose to use feature upsampling; End to end learning.
	MPN [233]	SharpMask [168]	VGG16	Fast RCNN	–	–	51.9	33.2	BMVC16	**(MultiPath Network, MPN)**; Use skip connections, multiscale proposals and an integral loss function to improve Fast RCNN; Ranked 2nd in both the COCO15 detection and segmentation challenges; Need segmentation annotations for training.
	DSOD [186]	Free	DenseNet	SSD	77.7 (07+12)	72.2 (07++12)	47.3	29.3	ICCV17	**(Deeply Supervised Object Detection, DSOD)**; Combine ideas of DenseNet and SSD; Training from scratch on the target dataset without pretraining with other datasets like ImageNet.
	RFBNet [135]	Free	VGG16	SSD	82.2 (07+12)	81.2 (07++12)	55.7	34.4	CVPR18	**(Receptive Field Block, RBF)**; Proposed RFB to improve SSD; RBF is a multibranch convolutional block similar to the Inception block [200], but with dilated CONV layers.
(3) Combination of (1) and (2)	DSSD [58]	Free	ResNet101	SSD	81.5 (07+12)	80.0 (07++12)	53.3	33.2	2017	**(Deconvolutional Single Shot Detector, DSSD)**; Design a top-down network connected with lateral connections to supplement the bottom-up network, as shown in Fig. 11 (c1, c2).
	FPN [130]	RPN	ResNet101	Faster RCNN	–	–	59.1	36.2	CVPR17	**(Feature Pyramid Network, FPN)**; Exploit inherent pyramidal hierarchy of DCNN to construct feature pyramids with marginal extra cost, as shown in Fig. 11 (a1, a2); Widely used in detectors.
	TDM [190]	RPN	ResNet101 VGG16	Faster RCNN	–	–	57.7	36.8	CVPR17	**(Top Down Modulation, TDM)**; Integrate top-down features and bottom-up, feedforward features via the proposed block shown in Fig. 11 (b2); Result 36.8 was produced by InceptionResNetv2.
	RON [104]	RPN	VGG16	Faster RCNN	81.3 (07+12+CO)	80.7 (07++12+CO)	49.5	27.4	CVPR17	**(Reverse connection with Objectness prior Networks, RON)**; Effectively combine Faster RCNN and SSD; Design a block shown in Fig. 11 (d2) to perform multiscale object detection in DCNN.
	ZIP [122]	RPN	Inceptionv2	Faster RCNN	79.8 (07+12)	–	–	–	IJCV18	**(Zoom out and In network for object Proposals, ZIP)**; Generate proposals in a deep conv/deconv network with multilayers, as shown in Fig. 12; Proposed a map attention decision (MAD) unit to weight the feature channels input to RPN.
	STDN [246]	Free	DenseNet169	SSD	80.9 (07+12)	–	51.0	31.8	CVPR18	**(Scale Transferrable Detection Network, STDN)**; Proposed a efficient scale transfer module embedded into DenseNet; The scale transfer layer rearranges elements by expanding the width and height of the feature map with channel elements.
	RefineDet [239]	RPN	VGG16 ResNet101	Faster RCNN	83.8 (07+12)	83.5 (07++12)	62.9	41.8	CVPR18	Proposed an anchor refinement module to obtain better and less anchors; Designed a transfer connection block as shown in Fig. 11 (e2) to improve features for classification.
	StairNet [217]	–	VGG16	SSD	78.8 (07+12)	76.4 (07++12)	–	–	WACV18	Design a transfer connection block similar to those shown in Fig. 11 to improve feature combination.
(4) Model Geometric Transforms	DeepIDNet [160]	SS+ EB	AlexNet ZFNet OverFeat GoogLeNet	RCNN	69.0 (07)	–	–	25.6	CVPR15	Introduce a deformation constrained pooling layer to explore object part deformation; Also utilize context modeling, model averaging, and bounding box location refinement in the multistage detection pipeline; Highly engineered; Training not end to end;
	DCN [41]	RPN	ResNet101 IRN	RFCN	82.6 (07+12)	–	58.0	37.5	CVPR17	**(Deformable Convolutional Networks, DCN)**; Design efficient deformable convolution and deformable RoI pooling modules that can replace their plain counterparts in existing DCNNs.
	DPFCN [147]	AttractioNet [63]	ResNet	RFCN	83.3 (07+12)	81.2 (07++12)	59.1	39.1	IJCV18	**(Deformable Part based FCN, DPFCN)**; Design a deformable part based RoI pooling layer to explicitly select discriminative regions around object proposals by simultaneously optimizing latent displacements of all parts.

As can be observed from Fig. 11 (a1) to (e1), these methods have highly similar detection architectures which incorporate a top down network with lateral connections to supplement the standard bottom-up, feedforward network. Specifically, after a bottom-up pass the final high level semantic features are transmitted back by the top-down network to combine with the bottom-up features from intermediate layers after lateral processing. The combined features are further processed, then used for detection and also transmitted down by the top-down network. As can be seen from Fig. 11 (a2) to (e2), one main difference is the design of the Re-

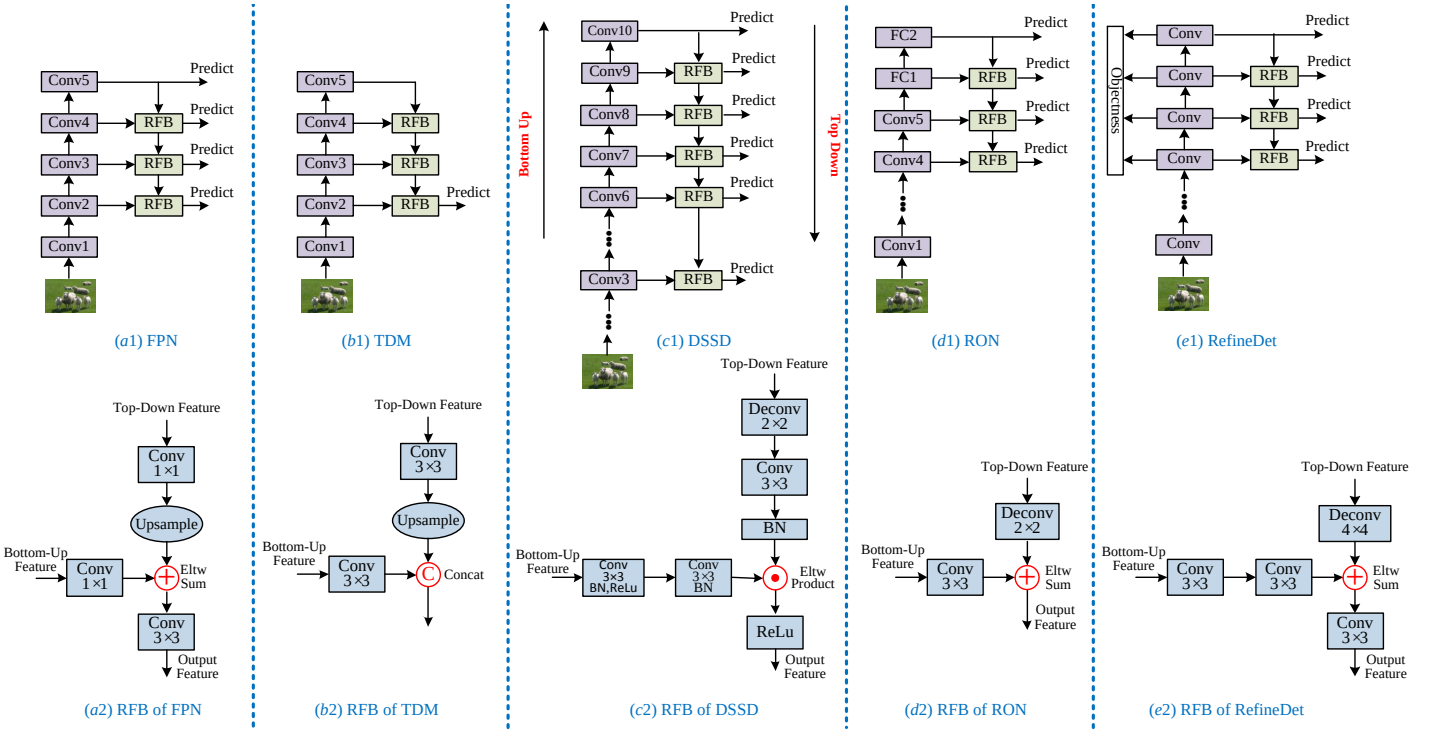


Fig. 11 Hourglass architectures: Conv1 to Conv5 are the main Conv blocks in backbone networks such as VGG or ResNet. Comparison of a number of Reverse Fusion Block (RFB) commonly used in recent approaches.

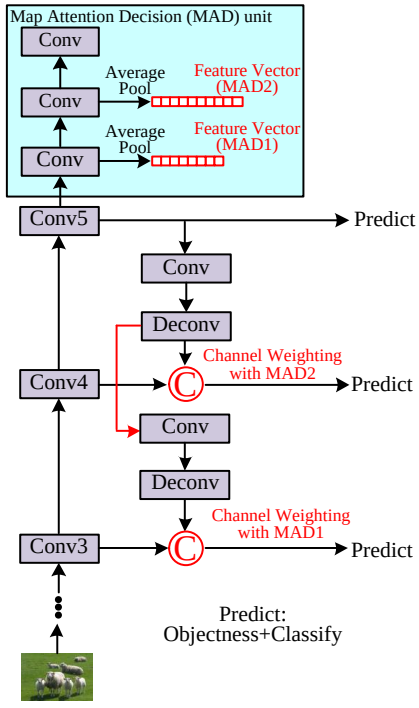


Fig. 12 ZIP is similar to the approaches in Fig. 11.

verse Fusion Block (RFB) which handles the selection of the lower layer filters and the combination of multilayer features. The top-down and lateral features are processed with small convolutions and combined with *elementwise sum* or *elementwise product* or *concatenation*. FPN shows significant improvement as a generic feature extractor in several applications including object detection [130, 131] and instance segmentation [80], *e.g.* using FPN in a basic Faster RCNN detector. These methods have to add additional layers to obtain multiscale features, introducing cost that can not be neglected. STDN [246] used DenseNet [94] to combine features of different layers and designed a scale transfer module to obtain feature maps with different resolutions. The scale transfer module module can be directly embedded into DenseNet with little additional cost.

(4) **Model Geometric Transformations.** DCNNs are inherently limited to model significant geometric transformations. An empirical study of the invariance and equivalence of DCNN representations to image transformations can be found in [118]. Some approaches have been presented to enhance the robustness of CNN representations, aiming at learning invariant CNN representations with respect to different types of transformations such as scale [101, 18], rotation [18, 32, 218, 248], or both [100].

Modeling Object Deformations: Before deep learning, Deformable Part based Models (DPMs) [56] have been very successful for generic object detection, representing objects by component parts arranged in a deformable configuration. This DPM modeling is less sensitive to transformations in object pose, viewpoint and nonrigid deformations because the parts are positioned accordingly and their local appearances are stable, motivating researchers [41, 66, 147, 160, 214] to explicitly model object composition to improve CNN based detection. The first attempts [66, 214] combined DPMs with CNNs by using deep features learned by AlexNet in DPM based detection, but without region proposals. To enable a CNN to enjoy the built-in capability of modeling the deformations of object parts, a number of approaches were proposed, including DeepIDNet [160], DCN [41] and DPFCN [147] (shown in Table 3). Although similar in spirit, deformations are computed in a different ways: DeepIDNet [161] designed a deformation constrained pooling layer to replace a regular max pooling layer to learn the shared visual patterns and their deformation

properties across different object classes, Dai *et al.* [41] designed a deformable convolution layer and a deformable RoI pooling layer, both of which are based on the idea of augmenting the regular grid sampling locations in the feature maps with additional position offsets and learning the offsets via convolutions, leading to Deformable Convolutional Networks (DCN), and in DPFCN [147], Mordan *et al.* proposed deformable part based RoI pooling layer which selects discriminative parts of objects around object proposals by simultaneously optimizing latent displacements of all parts.

4.2 Context Modeling

In the physical world visual objects occur in particular environments and usually coexist with other related objects, and there is strong psychological evidence [13, 9] that context plays an essential role in human object recognition. It is recognized that proper modeling of context helps object detection and recognition [203, 155, 27, 26, 47, 59], especially when object appearance features are insufficient because of small object size, occlusion, or poor image quality. Many different types of context have been discussed, in particular see surveys [47, 59]. Context can broadly be grouped into one of three categories [13, 59]:

1. Semantic context: The likelihood of an object to be found in some scenes but not in others;
2. Spatial context: Tthe likelihood of finding an object in some position and not others with respect to other objects in the scene;
3. Scale context: Objects have a limited set of sizes relative to other objects in the scene.

A great deal of work [28, 47, 59, 143, 152, 171, 162] preceded the prevalence of deep learning, however much of this work has not been explored in DCNN based object detectors [29, 90].

The current state of the art in object detection [175, 136, 80] detects objects without explicitly exploiting any contextual information. It is broadly agreed that DCNNs make use of contextual information implicitly [234, 242] since they learn hierarchical representations with multiple levels of abstraction. Nevertheless there is still value in exploring contextual information explicitly in DCNN based detectors [90, 29, 236], and so the following reviews recent work in exploiting contextual cues in DCNN based object detectors, organized into categories of *global* and *local* contexts, motivated by earlier work in [240, 59]. Representative approaches are summarized in Table 4.

Global context [240, 59] refers to image or scene level context, which can serve as cues for object detection (*e.g.,* a bedroom will predict the presence of a bed). In DeepIDNet [160], the image classification scores were used as contextual features, and concatenated with the object detection scores to improve detection results. In ION [10], Bell *et al.* proposed to use spatial Recurrent Neural Networks (RNNs) to explore contextual information across the entire image. In SegDeepM [250], Zhu *et al.* proposed a MRF model that scores appearance as well as context for each detection, and allows each candidate box to select a segment and score the agreement between them. In [188], semantic segmentation was used as a form of contextual priming.

Local context [240, 59, 171] considers local surroundings in object relations, the interactions between an object and its surrounding area. In general, modeling object relations is challenging, requiring reasoning about bounding boxes of different classes, locations, scales *etc*. In the deep learning era, research that explicitly models object relations is quite limited, with representative ones being Spatial Memory Network (SMN) [29], Object Relation Network [90], and Structure Inference Network (SIN) [137]. In SMN, spatial memory essentially assembles object instances back into a pseudo image representation that is easy to be fed into another CNN for object relations reasoning, leading to a new sequential reasoning architecture where image and memory are processed in parallel to obtain detections which further update memory. Inspired by the recent success of attention modules in natural language processing field [211], Hu *et al.* [90] proposed a lightweight ORN, which processes a set of objects simultaneously through interaction between their appearance feature and geometry. It does not require additional supervision and is easy to embed in existing networks. It has been shown to be effective in improving object recognition and duplicate removal steps in modern object detection pipelines, giving rise to the first fully end-to-end object detector. SIN [137] considered two kinds of context including scene contextual information and object relationships within a single image. It formulates object detection as a problem of graph structure inference, where given an image the objects are treated as nodes in a graph and relationships between objects are modeled as edges in such graph.

A wider range of methods has approached the problem more simply, normally by enlarging the detection window size to extract some form of local context. Representative approaches include MRCNN [62], Gated BiDirectional CNN (GBDNet) [235, 236], Attention to Context CNN (ACCNN) [123], CoupleNet [251], and Sermanet *et al.* [182].

In MRCNN [62] (Fig. 13 (a)), in addition to the features extracted from the original object proposal at the last CONV layer of the backbone, Gidaris and Komodakis proposed to extract features from a number of different regions of an object proposal (half regions, border regions, central regions, contextual region and semantically segmented regions), in order to obtain a richer and more robust object representation. All of these features are combined simply by concatenation.

Quite a number of methods, all closely related to MRCNN, have been proposed since. The method in [233] used only four contextual regions, organized in a foveal structure, where the classifier is trained jointly end to end. Zeng *et al.* proposed GBDNet [235, 236] (Fig. 13 (b)) to extract features from multiscale contextualized regions surrounding an object proposal to improve detection performance. Different from the naive way of learning CNN features for each region separately and then concatenating them, as in MRCNN, GBDNet can pass messages among features from different contextual regions, implemented through convolution. Noting that message passing is not always helpful but dependent on individual samples, Zeng *et al.* used gated functions to control message transmission, like in Long Short Term Memory (LSTM) networks [83]. Concurrent with GBDNet, Li *et al.* [123] presented ACCNN (Fig. 13 (c)) to utilize both global and local contextual information to facilitate object detection. To capture global context, a Multiscale Local Contextualized (MLC) subnetwork was pro-

Table 4 Summarization of detectors that exploit context information, similar to Table 3.

Group	Detector Name	Region Proposal	Backbone DCNN	Pipelined Used	mAP@IoU=0.5 VOC07	mAP@IoU=0.5 VOC12	mAP COCO	Published In	Full Name of Detector and Highlights
Global Context	SegDeepM [250]	SS+CMPC	VGG16	RCNN	VOC10	VOC12	–	CVPR15	Use an additional feature extracted from an enlarged object proposal as context information; Frame the detection problem as inference in a Markov Random Field.
	ION [10]	SS+EB	VGG16	Fast RCNN	80.1	77.9	33.1	CVPR16	**(Inside Outside Network, ION)**; Contextual information outside the region of interest is integrated using spatial recurrent neural networks.
	DeepIDNet [160]	SS+EB	AlexNet ZFNet	RCNN	69.0 (07)	–	–	CVPR15	Propose to use image classification scores as global contextual information to refine the detection scores of each object proposal.
	CPF [188]	RPN	VGG16	Faster RCNN	76.4 (07+12)	72.6 (07++12)	–	ECCV16	**(Contextual Priming and Feedback, CPF)**; Augment Faster RCNN with a semantic segmentation network; Use semantic segmentation to provide top down feedback.
Local Context	MRCNN [62]	SS	VGG16	SPPNet	78.2 (07+12)	73.9 (07+12)	–	ICCV15	**(MultiRegion CNN, MRCNN)**; Extract features from multiple regions surrounding or inside the object proposals; Intergrate the semantic segmentation-aware features.
	GBDNet [235]	CRAFT [224]	Inception v2 ResNet269	Fast RCNN	77.2 (07+12)	–	27.0	ECCV16	**(Gated BiDirectional CNN, GBDNet)**; Propose a GBDNet module to model the relations of multiscale contextualized regions surrounding an object proposal; GBDNet pass messages among features from different context regions through convolution between neighboring support regions in two directions; Gated functions are used to control message transmission.
	ACCNN[123]	SS	VGG16	Fast RCNN	72.0 (07+12)	70.6 (07++12)	–	TMM17	**(Attention to Context CNN, ACCNN)**; Propose to use multiple stacked LSTM layers to capture global context; Propose to encode features from multiscale contextualized regions surrounding an object proposal by feature concatenation. The global and local context feature are concatenated for recognition.
	CoupleNet[251]	RPN	ResNet101	RFCN	82.7 (07+12)	80.4 (07++12)	34.4	ICCV17	Improve on RFCN; Besides the main branch in the network head, propose an additional branch by concatenating features from multiscale contextualized regions surrounding an object proposal; Features from two branches are combined with elementwise sum.
	SMN [29]	RPN	VGG16	Faster RCNN	70.0 (07)	–	–	ICCV17	**(Spatial Memory Network, SMN)**; Propose a SMN to model object-object relationship efficiently and effectively; A sequential reasoning architecture.
	ORN [90]	RPN	ResNet101 +DCN	Faster RCNN	–	–	39.0	CVPR18	**(Object Relation Network, ORN)**; Propose an ORN to model the relations of a set of object proposals through interaction between their appearance feature and geometry; ORM does not require additional supervision and is easy to embed in existing networks.
	SIN [137]	RPN	VGG16	Faster RCNN	76.0 (07+12)	73.1 (07++12)	23.2	CVPR18	**(Structure Inference Network, SIN)**; Formulates object detection as a problem of graph structure inference, where given an image the objects are treated as nodes in a graph and relationships between the objects are modeled as edges in such graph.

posed, which recurrently generates an attention map for an input image to highlight useful global contextual locations, through multiple stacked LSTM layers. To encode local surroundings context, Li *et al.* [123] adopted a method similar to that in MRCNN [62]. As shown in Fig. 13 (d), CoupleNet [251] is conceptually similar to ACCNN [123], but built upon RFCN [40]. In addition to the original branch in RFCN [40], which captures object information with position sensitive RoI pooling, CoupleNet [251] added one branch to encode the global context information with RoI pooling.

4.3 Detection Proposal Methods

An object can be located at any position and scale in an image. During the heyday of handcrafted feature descriptors (*e.g.,* SIFT [140], HOG [42] and LBP [153]), the Bag of Words (BoW) [194, 37] and the DPM [55] used *sliding window* techniques [213, 42, 55, 76, 212]. However the number of windows is large and grows with the number of pixels in an image, and the need to search at multiple scales and aspect ratios further significantly increases the search space. Therefore, it is computationally too expensive to apply more sophisticated classifiers.

Around 2011, researchers proposed to relieve the tension between computational tractability and high detection quality by using *detection proposals*[3] [210, 209]. Originating in the idea of *objectness* proposed by [2], object proposals are a set of candidate regions in an image that are likely to contain objects. Detection proposals are usually used as a preprocessing step, in order to reduce the computational complexity by limiting the number of regions that need be evaluated by the detector. Therefore, a good detection proposal should have the following characteristics:

1. High recall, which can be achieved with only a few proposals;
2. The proposals match the objects as accurately as possible;
3. High efficiency.

The success of object detection based on detection proposals given by selective search [210, 209] has attracted broad interest [21, 7, 3, 33, 254, 50, 105, 144].

A comprehensive review of object proposal algorithms is outside the scope of this paper, because object proposals have applications beyond object detection [6, 72, 252]. We refer interested readers to the recent surveys [86, 23] which provides an in-depth

[3] We use the terminology *detection proposals*, *object proposals* and *region proposals* interchangeably.

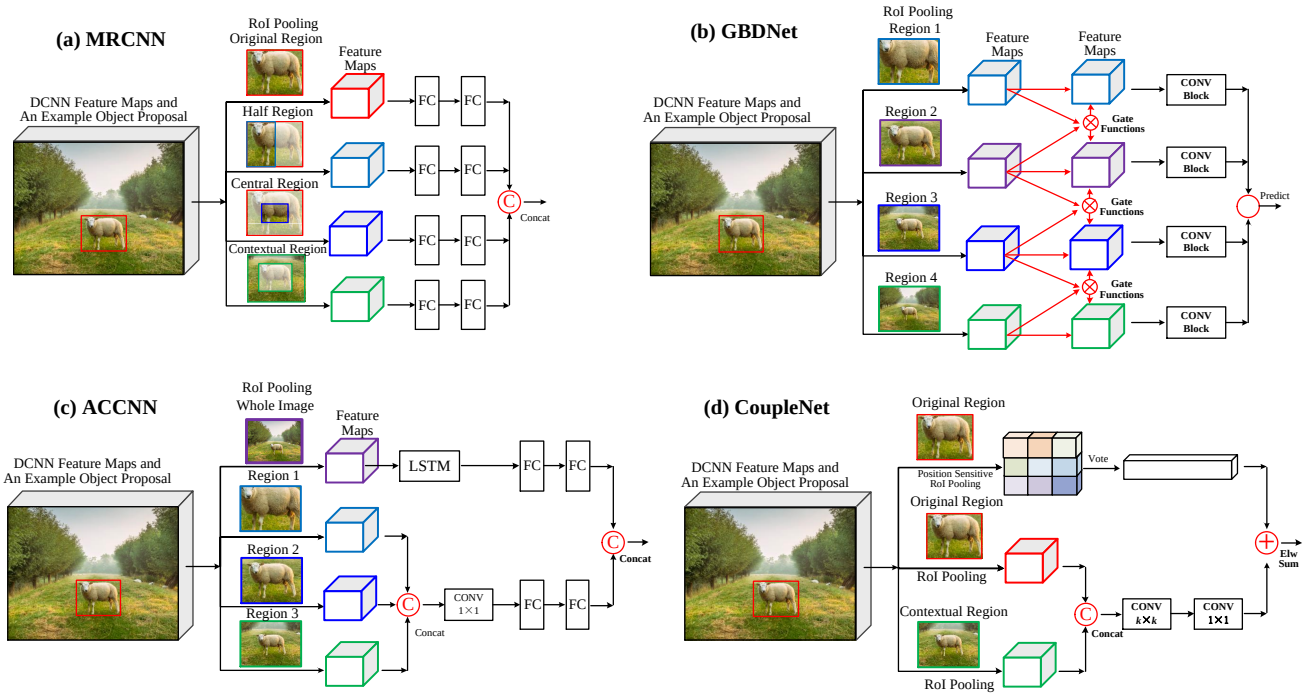


Fig. 13 Representative approaches that explore local surrounding contextual features: MRCNN [62], GBDNet [235, 236], ACCNN [123] and CoupleNet [251], see also Table 4.

analysis of many classical object proposal algorithms and their impact on detection performance. Our interest here is to review object proposal methods that are based on DCNNs, output class agnostic proposals, and related to generic object detection.

In 2014, the integration of object proposals [210, 209] and DCNN features [109] led to the milestone RCNN [65] in generic object detection. Since then, detection proposal algorithms have quickly become a standard preprocessing step, evidenced by the fact that all winning entries in the PASCAL VOC [53], ILSVRC [179] and MS COCO [129] object detection challenges since 2014 used detection proposals [65, 160, 64, 175, 236, 80].

Among object proposal approaches based on traditional low-level cues (*e.g.,* color, texture, edge and gradients), Selective Search [209], MCG [7] and EdgeBoxes [254] are among the more popular. As the domain rapidly progressed, traditional object proposal approaches [86] (*e.g.* selective search[209] and [254]), which were adopted as external modules independent of the detectors, became the bottleneck of the detection pipeline [175]. An emerging class of object proposal algorithms [52, 175, 111, 61, 167, 224] using DCNNs has attracted broad attention.

Recent DCNN based object proposal methods generally fall into two categories: *bounding box* based and *object segment* based, with representative methods summarized in Table 5.

Bounding Box Proposal Methods is best exemplified by the RPC method [175] of Ren *et al.*, illustrated in Fig. 14. RPN predicts object proposals by sliding a small network over the feature map of the last shared CONV layer (as shown in Fig. 14). At each sliding window location, it predicts k proposals simultaneously by using k anchor boxes, where each anchor box[4] is centered at some location in the image, and is associated with a particular scale and aspect ratio. Ren *et al.* [175] proposed to integrate RPN and Fast

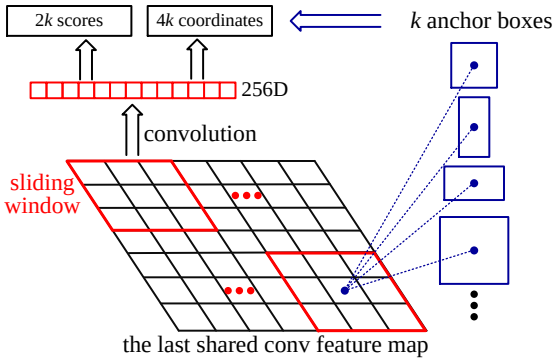


Fig. 14 Illustration of the Region Proposal Network (RPN) proposed in [175].

RCNN into a single network by sharing their convolutional layers. Such a design led to substantial speedup and the first end-to-end detection pipeline, Faster RCNN [175]. RPN has been broadly selected as the proposal method by many state of the art object detectors, as can be observed from Tables 3 and 4.

Instead of fixing *a priori* a set of anchors as MultiBox [52, 199] and RPN [175], Lu *et al.* [141] proposed to generate anchor locations by using a recursive search strategy which can adaptively guide computational resources to focus on subregions likely to contain objects. Starting with the whole image, all regions visited during the search process serve as anchors. For any anchor region encountered during the search procedure, a scalar zoom indicator is used to decide whether to further partition the region, and a set of bounding boxes with objectness scores are computed with a deep network called Adjacency and Zoom Network (AZNet). AZNet extends RPN by adding a branch to compute the scalar zoom indicator in parallel with the existing branch.

There is further work attempting to generate object proposals by exploiting multilayer convolutional features [103, 61, 224, 122].

[4] The terminology "an anchor box" or "an anchor" first appeared in [175].

Table 5 Summarization of object proposal methods using DCNN. The numbers in blue color denote the the number of object proposals. The detection results on COCO is mAP@IoU[0.5, 0.95], unless stated otherwise.

	Proposer Name	Backbone Network	Detector Tested	Recall@IoU (VOC07) 0.5	0.7	0.9	Detection Results (mAP) VOC07	VOC12	COCO	Published In	Full Name of Detector and Highlights
Bounding Box Object Proposal Methods	MultiBox1[52]	AlexNet	RCNN	–	–	–	29.0 (10) (12)	–	–	CVPR14	Among the first to explore DCNN for object proposals; Learns a class agnostic regressor on a small set of 800 predefined anchor boxes; Does not share features with the detection network.
	DeepBox [111]	VGG16	Fast RCNN	0.96 (1000)	0.84 (1000)	0.15 (1000)	–	–	37.8 (500) (IoU@0.5)	ICCV15	Propose a light weight CNN to learn to rerank proposals generated by EdgeBox; Can run at 0.26s per image; Not sharing features extracted for detection.
	RPN[175, 176]	VGG16	Faster RCNN	0.97 (300) 0.98 (1000)	0.79 (300) 0.84 (1000)	0.04 (300) 0.04 (1000)	73.2 (300) (07+12)	70.4 (300) (07++12)	21.9 (300)	NIPS15	(**Region Proposal Network, RPN**); First to generate object proposals by sharing full image convolutional features with the detection network; Firstly introduced the anchor boxes idea; Most widely used object proposal method; Greatly improved the detection speed and accuracy.
	DeepProposal[61]	VGG16	Fast RCNN	0.74 (100) 0.92 (1000)	0.58 (100) 0.80 (1000)	0.12 (100) 0.16 (1000)	53.2 (100) (07)	–	–	ICCV15	Generated proposals inside a DCNN in a multiscale manner; Selected the most promising object locations and refined their boxes in a coarse to fine cascading way; Used the off the shelf DCNN features for generating proposals; Sharing features with the detection network.
	CRAFT [224]	VGG16	Faster RCNN	0.98 (300)	0.90 (300)	0.13 (300)	75.7 (07+12)	71.3 (12)	–	CVPR16	(**Cascade Region proposal network And FasT rcnn, CRAFT**); Introduced a classification Network (*i.e.* two class Fast RCNN) cascade that comes after the RPN. Not sharing features extracted for detection.
	AZNet [141]	VGG16	Fast RCNN	0.91 (300)	0.71 (300)	0.11 (300)	70.4 (07)	–	22.3	CVPR16	(**Adjacency and Zoom Network, AZNet**); Generates anchor locations by using a recursive search strategy which can adaptively guide computational resources to focus on subregions likely to contain objects.
	ZIP [122]	Inception v2	Faster RCNN	0.85 (300) COCO	0.74 (300) COCO	0.35 (300) COCO	79.8 (07+12)	–	–	IJCV18	(**Zoom out and In network for object Proposals, ZIP**); Generate proposals in a deep conv/deconv network with multilayers; Proposed a map attention decision (MAD) unit to weight the feature channels input to RPN;
	DeNet[208]	ResNet101	Fast RCNN	0.82 (300)	0.74 (300)	0.48 (300)	77.1 (07+12)	73.9 (07++12)	33.8	ICCV17	A lot faster than Faster RCNN; Introduces a bounding box corner estimation for predict object proposals efficiently to replace RPN; Doesn't require predefined anchors.

	Proposer Name	Backbone Network	Detector Tested	Box Proposals (AR, COCO)	Segment Proposals (AR, COCO)	Published In	Highlights
Segment Proposal Methods	DeepMask [167]	VGG16	Fast RCNN	0.33 (100), 0.48(1000)	0.26 (100), 0.37(1000)	NIPS15	First to generate object mask proposals with DCNN; Slow inference time; Need segmentation annotations for training; Not sharing features with detection network; Achieved mAP of 69.9% (500) with Fast RCNN.
	InstanceFCN [38]	VGG16	–	–	0.32 (100), 0.39(1000)	ECCV16	(**Instance Fully Convolutional Networks, InstanceFCN**); Combine ideas of FCN [138] and DeepMask [167]; Introduce instance sensitive score maps; Need segmentation annotations to train their network.
	SharpMask [168]	MPN [233]	Fast RCNN	0.39 (100), 0.53(1000)	0.30 (100), 0.39(1000)	ECCV16	Leverages features at multiple convolutional layers by introducing a top down refinement module; Does not share features with detection network; Need segmentation annotations for training; Slow for real time use.
	FastMask[89]	ResNet39	–	0.43 (100), 0.57(1000)	0.32 (100), 0.41(1000)	CVPR17	Generate instance segment proposals efficiently in one shot manner similar to SSD [136], in order to make use of multiscale convolutional features in a deep network; Need segmentation annotations for training.
	ScaleNet [170]	ResNet	–	0.44 (100), 0.58(1000)	0.35 (100), 0.45(1000)	ICCV17	Extends SharpMask by explicitly adding a scale prediction phase; Proposed ScaleNet to estimate the distribution of the object scales for an input image. Performs well on supermarket datasets.

Concurrent with RPN [175], Ghodrati *et al.* [61] proposed DeepProposal which generates object proposals by using a cascade of multiple convolutional features, building an inverse cascade to select the most promising object locations and to refine their boxes in a coarse to fine manner. An improved variant of RPN, HyperNet [103] designs Hyper Features which aggregate multilayer convolutional features and shares them both in generating proposals and detecting objects via an end to end joint training strategy. Yang *et al.* proposed CRAFT [224] which also used a cascade strategy, first training an RPN network to generate object proposals and then using them to train another binary Fast RCNN network to further distinguish objects from background. Li *et al.* [122] proposed ZIP to improve RPN by leveraging a commonly used idea of predicting object proposals with multiple convolutional feature maps at different depths of a network to integrate both low level details and high level semantics. The backbone network used in ZIP is a "zoom out and in" network inspired by the conv and deconv structure [138].

Finally, recent work which deserves mention includes Deepbox [111], which proposed a light weight CNN to learn to rerank proposals generated by EdgeBox, and DeNet [208] which introduces a bounding box corner estimation to predict object proposals efficiently to replace RPN in a Faster RCNN style two stage detector.

Object Segment Proposal Methods [167, 168] aim to generate segment proposals that are likely to correspond to objects. Segment proposals are more informative than bounding box proposals, and take a step further towards object instance segmentation [74, 39, 126]. A pioneering work was DeepMask proposed by Pinheiro *et al.* [167], where segment proposals are learned directly from raw image data with a deep network. Sharing similarities with RPN, after a number of shared convolutional layers DeepMask splits the network into two branches to predict a class agnostic mask and an associated objectness score. Similar to the efficient sliding window prediction strategy in OverFeat [183], the trained DeepMask network is applied in a sliding window manner to an image (and its rescaled versions) during inference. More recently, Pinheiro *et al.* [168] proposed SharpMask by augmenting the DeepMask architecture with a refinement module, similar to the architectures shown in Fig. 11 (b1) and (b2), augmenting the feedforward network with a top-down refinement process. SharpMask can efficiently integrate the spatially rich information from early features with the strong semantic information encoded in later layers to generate high fidelity object masks.

Motivated by Fully Convolutional Networks (FCN) for semantic segmentation [138] and DeepMask [167], Dai *et al.* proposed InstanceFCN [38] for generating instance segment proposals. Similar to DeepMask, the InstanceFCN network is split into two branches,

howver the two branches are fully convolutional, where one branch generates a small set of instance sensitive score maps, followed by an assembling module that outputs instances, and the other branch for predicting the objectness score. Hu *et al.* proposed FastMask [89] to efficiently generate instance segment proposals in a one-shot manner similar to SSD [136], in order to make use of multiscale convolutional features in a deep network. Sliding windows extracted densely from multiscale convolutional feature maps were input to a scale-tolerant attentional head module to predict segmentation masks and objectness scores. FastMask is claimed to run at 13 FPS on a 800×600 resolution image with a slight trade off in average recall. Qiao *et al.* [170] proposed ScaleNet to extend previous object proposal methods like SharpMask [168] by explicitly adding a scale prediction phase. That is, ScaleNet estimates the distribution of object scales for an input image, upon which SharpMask searches the input image at the scales predicted by ScaleNet and outputs instance segment proposals. Qiao *et al.* [170] showed their method outperformed the previous state of the art on supermarket datasets by a large margin.

4.4 Other Special Issues

Aiming at obtaining better and more robust DCNN feature representations, data augmentation tricks are commonly used [22, 64, 65]. It can be used at training time, at test time, or both. Augmentation refers to perturbing an image by transformations that leave the underlying category unchanged, such as cropping, flipping, rotating, scaling and translating in order to generate additional samples of the class. Data augmentation can affect the recognition performance of deep feature representations. Nevertheless, it has obvious limitations. Both training and inference computational complexity increases significantly, limiting its usage in real applications. Detecting objects under a wide range of scale variations, and especially, detecting very small objects stands out as one of key challenges. It has been shown [96, 136] that image resolution has a considerable impact on detection accuracy. Therefore, among those data augmentation tricks, scaling (especially a higher resolution input) is mostly used, since high resolution inputs enlarge the possibility of small objects to be detected [96]. Recently, Singh *et al.* proposed advanced and efficient data argumentation methods SNIP [192] and SNIPER [193] to illustrate the scale invariance problem, as summarized in Table 6. Motivated by the intuitive understanding that small and large objects are difficult to detect at smaller and larger scales respectively, Singh *et al.* presented a novel training scheme named SNIP can reduce scale variations during training but without reducing training samples. SNIPER [193] is an approach proposed for efficient multiscale training. It only processes context regions around ground truth objects at the appropriate scale instead of processing a whole image pyramid. Shrivastava *et al.* [189] and Lin *et al.* explored approaches to handle the extreme foreground-background class imbalance issue [131]. Wang *et al.* [216] proposed to train an adversarial network to generate examples with occlusions and deformations that are difficult for the object detector to recognize. There are some works focusing on developing better methods for nonmaximum suppression [16, 87, 207].

5 Datasets and Performance Evaluation

5.1 Datasets

Datasets have played a key role throughout the history of object recognition research. They have been one of the most important factors for the considerable progress in the field, not only as a common ground for measuring and comparing performance of competing algorithms, but also pushing the field towards increasingly complex and challenging problems. The present access to large numbers of images on the Internet makes it possible to build comprehensive datasets of increasing numbers of images and categories in order to capture an ever greater richness and diversity of objects. The rise of large scale datasets with millions of images has paved the way for significant breakthroughs and enabled unprecedented performance in object recognition. Recognizing space limitations, we refer interested readers to several papers [53, 54, 129, 179, 107] for detailed description of related datasets.

Beginning with Caltech101 [119], representative datasets include Caltech256 [70], Scenes15 [114], PASCAL VOC (2007) [54], Tiny Images [204], CIFAR10 [108], SUN [221], ImageNet [44], Places [245], MS COCO [129], and Open Images [106]. The features of these datasets are summarized in Table 7, and selected sample images are shown in Fig. 15.

Earlier datasets, such as Caltech101 or Caltech256, were criticized because of the lack of intraclass variations that they exhibit. As a result, SUN [221] was collected by finding images depicting various scene categories, and many of its images have scene and object annotations which can support scene recognition and object detection. Tiny Images [204] created a dataset at an unprecedented scale, giving comprehensive coverage of all object categories and scenes, however its annotations were not manually verified, containing numerous errors, so two benchmarks (CIFAR10 and CIFAR100 [108]) with reliable labels were derived from Tiny Images.

PASCAL VOC [53, 54], a multiyear effort devoted to the creation and maintenance of a series of benchmark datasets for classification and object detection, creates the precedent for standardized evaluation of recognition algorithms in the form of annual competitions. Starting from only four categories in 2005, increasing to 20 categories that are common in everyday life, as shown in Fig. 15. ImageNet [44] contains over 14 million images and over 20,000 categories, the backbone of ILSVRC [44, 179] challenge, which has pushed object recognition research to new heights.

ImageNet has been criticized that the objects in the dataset tend to be large and well centered, making the dataset atypical of real world scenarios. With the goal of addressing this problem and pushing research to richer image understanding, researchers created the MS COCO database [129]. Images in MS COCO are complex everyday scenes containing common objects in their natural context, closer to real life, and objects are labeled using fully-segmented instances to provide more accurate detector evaluation. The Places database [245] contains 10 million scene images, labeled with scene semantic categories, offering the opportunity for data hungry deep learning algorithms to reach human level recognition of visual patterns. More recently, Open Images [106] is a dataset of about 9 million images that have been annotated with image level labels and object bounding boxes.

Table 6 Representative methods for training strategies and class imbalance handling. Results on COCO are reported with Test-Dev.

Detector Name	Region Proposal	Backbone DCNN	Pipelined Used	VOC07 Results	VOC12 Results	COCO Results	Published In	Full Name of Method and Highlights
MegDet [164]	RPN	ResNet50 +FPN	Faster RCNN	–	–	52.5	CVPR18	Allow training with much larger minibatch size (like 256) than before by introducing cross GPU batch normalization; Can finish the COCO training in 4 hours on 128 GPUs and achieved improved accuracy; Won COCO2017 detection challenge.
SNIP [193]	RPN	DPN [31] +DCN [41]	RFCN	–	–	48.3	CVPR18	**(Scale Normalization for Image Pyramids, SNIP)**; A novel scale normalized training scheme; Empirically examined the effect of upsampling for small object detection; During training, selectively back propagates the gradients of object instances by ignoring gradients arising from objects of extreme scales.
SNIPER [193]	RPN	ResNet101 +DCN	Faster RCNN	–	–	47.6	2018	**(Scale Normalization for Image Pyramids with Efficient Resampling, SNIPER);** An efficient multiscale data argumentation strategy.
OHEM [189]	SS	VGG16	Fast RCNN	78.9 (07+12)	76.3 (07++12)	22.4	CVPR16	**(Online Hard Example Mining, OHEM)**; A simple and effective OHEM algorithm to improve training of region based detectors.
RetinaNet [131]	free	ResNet101 +FPN	RetinaNet	–	–	39.1	ICCV17	Proposed a simple dense detector called RetinaNet; Proposed a novel Focal Loss which focuses training on a sparse set of hard examples; High detection speed and high detection accuracy.

Table 7 Popular databases for object recognition. Some example images from MNIST, Caltech101, CIFAR10, PASCAL VOC and ImageNet are shown in Fig. 15.

Dataset Name	Total Images	Categories	Images Per Category	Objects Per Image	Image Size	Started Year	Highlights
MNIST [115]	60,000	10	6,000	1	28×28	1998	Handwritten digits; Single object; Binary images; Blank backgrounds; Small image size.
Caltech101 [119]	9,145	101	$40 \sim 800$	1	300×200	2004	Relatively smaller number of training images; A single object centered on each image; No clutter in most images; Limited intraclass variations; Less applicable to real-world evaluation.
Caltech256 [70]	30,607	256	80+	1	300×200	2007	Similar to the Caltech101, a larger number of classes than the Caltech101.
Scenes15 [114]	4,485	15	$200 \sim 400$	–	256×256	2006	Collected for scene recognition.
Tiny Images [204]	79 millions+	53,464	–	–	32×32	2006	Largest number of images, largest number of categories, low resolution images, not manually verified, less suitable for algorithm evaluation; Two subsets CIFAR10 and CIFAR100 as popular benchmarks for object classification.
PASCAL VOC (2012) [54]	11,540	20	$303 \sim 4087$	2.4	470×380	2005	Covers only 20 categories that are common in everyday life; Large number of training images; Close to real-world applications; Significantly larger intraclass variations; Objects in scene context; Multiple objects in one image; Contains many difficult samples; Creates the precedent for standardized evaluation of recognition algorithms in the form of annual competitions.
SUN [221]	131,072	908	–	16.8	500×300	2010	A large number scene categories; The number of instances per object category exhibits the long tail phenomenon; Two benchmarks SUN397 and SUN2012 for scene recognition and object detection respectively.
ImageNet [179]	14 millions+	21,841	–	1.5	500×400	2009	Considerably larger number of object categories; More instances and more categories of objects per image; More challenging than PASCAL VOC; Popular subset benchmarks ImageNet1000; The backbone of ILSVRC challenge; Images are object-centric.
MS COCO [129]	328,000+	91	–	7.3	640×480	2014	Even closer to real world scenarios; Each image contains more instances of objects and richer object annotation information; Contains object segmentation notation data that is not available in the ImageNet dataset; The next major dataset for large scale object detection and instance segmentation.
Places [245]	10 millions+	434	–	–	256×256	2014	The largest labeled dataset for scene recognition; Four subsets Places365 Standard, Places365 Challenge, Places 205 and Places88 as benchmarks.
Open Images [106]	9 millions+	6000+	–	–	varied	2017	A dataset of about 9 million images that have been annotated with image level labels and object bounding boxes.

There are three famous challenges for generic object detection: PASCAL VOC [53, 54], ILSVRC [179] and MS COCO [129]. Each challenge consists of two components: (i) a publicly available dataset of images together with ground truth annotation and standardized evaluation software; and (ii) an annual competition and corresponding workshop. Statistics for the number of images and object instances in the training, validation and testing datasets [5] for the detection challenges is given in Table 8.

For the PASCAL VOC challenge, since 2009 the data consist of the previous years' images augmented with new images, allowing the number of images to grow each year and, more importantly, meaning that test results can be compared with the previous years' images.

ILSVRC [179] scales up PASCAL VOC's goal of standardized training and evaluation of detection algorithms by more than an order of magnitude in the number of object classes and images. The ILSVRC object detection challenge has been run annually from 2013 to the present.

[5] The annotations on the test set are not publicly released, except for PASCAL VOC2007.

Fig. 15 Some example images from MNIST, Caltech101, CIFAR10, PASCAL VOC and ImageNet. See Table 7 for summary of these datasets.

The COCO object detection challenge is designed to push the state of the art in generic object detection forward, and has been run annually from 2015 to the present. It features two object detection tasks: using either bounding box output or object instance segmentation output. It has fewer object categories than ILSVRC (80 in COCO versus 200 in ILSVRC object detection) but more instances per category (11000 on average compared to about 2600 in ILSVRC object detection). In addition, it contains object segmentation annotations which are not currently available in ILSVRC. COCO introduced several new challenges: (1) it contains objects at a wide range of scales, including a high percentage of small objects (*e.g.* smaller than 1% of image area [192]). (2) objects are less iconic and amid clutter or heavy occlusion, and (3) the evaluation metric (see Table 9) encourages more accurate object localization.

COCO has become the most widely used dataset for generic object detection, with the dataset statistics for training, validation and testing summarized in Table 8. Starting in 2017, the test set has only the *Dev* and *Challenge* splits, where the Test-Dev split is the default test data, and results in papers are generally reported on Test-Dev to allow for fair comparison.

2018 saw the introduction of the Open Images Object Detection Challenge, following in the tradition of PASCAL VOC, ImageNet and COCO, but at an unprecedented scale. It offers a broader range of object classes than previous challenges, and has two tasks: bounding box object detection of 500 different classes and visual relationship detection which detects pairs of objects in particular relations.

5.2 Evaluation Criteria

There are three criteria for evaluating the performance of detection algorithms: detection speed (Frames Per Second, FPS), precision, and recall. The most commonly used metric is *Average Precision* (AP), derived from precision and recall. AP is usually evaluated in a category specific manner, *i.e.*, computed for each object category separately. In generic object detection, detectors are usually tested in terms of detecting a number of object categories. To compare performance over all object categories, the *mean AP* (mAP) averaged over all object categories is adopted as the final measure of performance[6]. More details on these metrics can be found in [53, 54, 179, 84].

The standard outputs of a detector applied to a testing image $\mathbf{I}$ are the predicted detections $\{(b_j, c_j, p_j)\}_j$, indexed by j. A given detection (b, c, p) (omitting j for notational simplicity) denotes the

[6] In object detection challenges such as PASCAL VOC and ILSVRC, the winning entry of each object category is that with the highest AP score, and the winner of the challenge is the team that wins on the most object categories. The mAP is also used as the measure of a team's performance, and is justified since the ranking of teams by mAP was always the same as the ranking by the number of object categories won [179].

Table 8 Statistics of commonly used object detection datasets. Object statistics for VOC challenges list the nondifficult objects used in the evaluation (all annotated objects). For the COCO challenge, prior to 2017, the test set had four splits (*Dev*, *Standard*, *Reserve*, and *Challenge*), with each having about 20K images. Starting in 2017, test set has only the *Dev* and *Challenge* splits, with the other two splits removed.

Challenge	Object Classes	Number of Images			Number of Annotated Objects	
		Train	Val	Test	Train	Val
PASCAL VOC Object Detection Challenge						
VOC07	20	2,501	2,510	4,952	6,301(7,844)	6,307(7,818)
VOC08	20	2,111	2,221	4,133	5,082(6,337)	5,281(6,347)
VOC09	20	3,473	3,581	6,650	8,505(9,760)	8,713(9,779)
VOC10	20	4,998	5,105	9,637	11,577(13,339)	11,797(13,352)
VOC11	20	5,717	5,823	10,994	13,609(15,774)	13,841(15,787)
VOC12	20	5,717	5,823	10,991	13,609(15,774)	13,841(15,787)
ILSVRC Object Detection Challenge						
ILSVRC13	200	395,909	20,121	40,152	345,854	55,502
ILSVRC14	200	456,567	20,121	40,152	478,807	55,502
ILSVRC15	200	456,567	20,121	51,294	478,807	55,502
ILSVRC16	200	456,567	20,121	60,000	478,807	55,502
ILSVRC17	200	456,567	20,121	65,500	478,807	55,502
MS COCO Object Detection Challenge						
MS COCO15	80	82,783	40,504	81,434	604,907	291,875
MS COCO16	80	82,783	40,504	81,434	604,907	291,875
MS COCO17	80	118,287	5,000	40,670	860,001	36,781
MS COCO18	80	118,287	5,000	40,670	860,001	36,781
Open Images Object Detection Challenge						
OID18	500	1,743,042	41,620	125,436	12,195,144	–

Input: $\{(b_j, p_j)\}_{j=1}^{M}$: M predictions for image $\mathbf{I}$ for object class c, ranked by the confidence p_j in decreasing order;
$\mathcal{B} = \{b_k^g\}_{k=1}^{K}$: ground truth BBs on image $\mathbf{I}$ for object class c;
Output: $\boldsymbol{a} \in \mathbb{R}^M$: a binary vector indicating each (b_j, p_j) to be a TP or FP.
Initialize $\boldsymbol{a} = 0$;
for $j = 1, ..., M$ **do**
  Set $\mathcal{A} = \varnothing$ and $t = 0$;
  foreach *unmatched object* b_k^g *in* $\mathcal{B}$ **do**
    if $IOU(b_j, b_k^g) \geq \varepsilon$ *and* $IOU(b_j, b_k^g) > t$ **then**
      $\mathcal{A} = \{b_k^g\}$;
      $t = \text{IOU}(b_j, b_k^g)$;
    end
  end
  if $\mathcal{A} \neq \varnothing$ **then**
    Set $\boldsymbol{a}(i) = 1$ since object prediction (b_j, p_j) is a TP;
    Remove the matched GT box in $\mathcal{A}$ from $\mathcal{B}$, $\mathcal{B} = \mathcal{B} - \mathcal{A}$.
  end
end

Fig. 16 The algorithm for determining TPs and FPs by greedily matching object detection results to ground truth boxes.

predicted location (*i.e.,* the Bounding Box, BB) b with its predicted category label c and its confidence level p. A predicted detection (b, c, p) is regarded as a True Positive (TP) if

- The predicted class label c is the same as the ground truth label c_g.
- The overlap ratio IOU (Intersection Over Union) [53, 179]

$$\text{IOU}(b, b^g) = \frac{area(b \cap b^g)}{area(b \cup b^g)}, \quad (1)$$

between the predicted BB b and the ground truth one b^g is not smaller than a predefined threshold ε. Here $area(b \cap b^g)$ denotes the intersection of the predicted and ground truth BBs, and $area(b \cup b^g)$ their union. A typical value of ε is 0.5.

Otherwise, it is considered as a False Positive (FP). The confidence level p is usually compared with some threshold β to determine whether the predicted class label c is accepted.

Table 9 Summarization of commonly used metrics for evaluating object detectors.

Metric	Meaning	Definition and Description	
TP	True Positive	A true positive detection, per Fig. 16.	
FP	False Positive	A false positive detection, per Fig. 16.	
β	Confidence Threshold	A confidence threshold for computing $P(\beta)$ and $R(\beta)$.	
ε	IOU Threshold	VOC	Typically around 0.5
		ILSVRC	$\min(0.5, \frac{wh}{(w+10)(h+10)})$; $w \times h$ is the size of a GT box.
		MS COCO	Ten IOU thresholds $\varepsilon \in \{0.5 : 0.05 : 0.95\}$
$P(\beta)$	Precision	The fraction of correct detections out of the total detections returned by the detector with confidence of at least β.	
$R(\beta)$	Recall	The fraction of all N_c objects detected by the detector having a confidence of at least β.	
AP	Average Precision	Computed over the different levels of recall achieved by varying the confidence β.	
mAP	mean Average Precision	VOC	AP at a single IOU and averaged over all classes.
		ILSVRC	AP at a modified IOU and averaged over all classes.
		MS COCO	• AP_{coco}: mAP averaged over ten IOUs: $\{0.5 : 0.05 : 0.95\}$; • $AP_{coco}^{\text{IOU}=0.5}$: mAP at IOU=0.50 (PASCAL VOC metric); • $AP_{coco}^{\text{IOU}=0.75}$: mAP at IOU=0.75 (strict metric); • AP_{coco}^{small}: mAP for small objects of area smaller than 32^2; • $AP_{coco}^{\text{medium}}$: mAP for objects of area between 32^2 and 96^2; • AP_{coco}^{large}: mAP for large objects of area bigger than 96^2;
AR	Average Recall	The maximum recall given a fixed number of detections per image, averaged over all categories and IOU thresholds.	
AR	Average Recall	MS COCO	• $AR_{coco}^{\text{max}=1}$: AR given 1 detection per image; • $AR_{coco}^{\text{max}=10}$: AR given 10 detection per image; • $AR_{coco}^{\text{max}=100}$: AR given 100 detection per image; • AR_{coco}^{small}: AR for small objects of area smaller than 32^2; • $AR_{coco}^{\text{medium}}$: AR for objects of area between 32^2 and 96^2; • AR_{coco}^{large}: AR for large objects of area bigger than 96^2;

AP is computed separately for each of the object classes, based on *Precision* and *Recall*. For a given object class c and a testing image $\mathbf{I}_i$, let $\{(b_{ij}, p_{ij})\}_{j=1}^{M}$ denote the detections returned by a detector, ranked by the confidence p_{ij} in decreasing order. Let $\mathcal{B} = \{b_{ik}^g\}_{k=1}^{K}$ be the ground truth boxes on image $\mathbf{I}_i$ for the given object class c. Each detection (b_{ij}, p_{ij}) is either a TP or a FP, which can be determined via the algorithm[7] in Fig. 16. Based on the TP and FP detections, the precision $P(\beta)$ and recall $R(\beta)$ [53] can be computed as a function of the confidence threshold β, so by varying the confidence threshold different pairs (P, R) can be obtained, in principle allowing precision to be regarded as a function of recall, *i.e.* $P(R)$, from which the Average Precision (AP) [53, 179] can be found.

Table 9 summarizes the main metrics used in the PASCAL, ILSVRC and MS COCO object detection challenges.

5.3 Performance

A large variety of detectors has appeared in the last several years, and the introduction of standard benchmarks such as PASCAL VOC [53, 54], ImageNet [179] and COCO [129] has made it easier to compare detectors with respect to accuracy. As can be seen from our earlier discussion in Sections 3 and 4, it is difficult to objectively compare detectors in terms of accuracy, speed and memory alone, as they can differ in fundamental / contextual respects, including the following:

[7] It is worth noting that for a given threshold β, multiple detections of the same object in an image are not considered as all correct detections, and only the detection with the highest confidence level is considered as a TP and the rest as FPs.

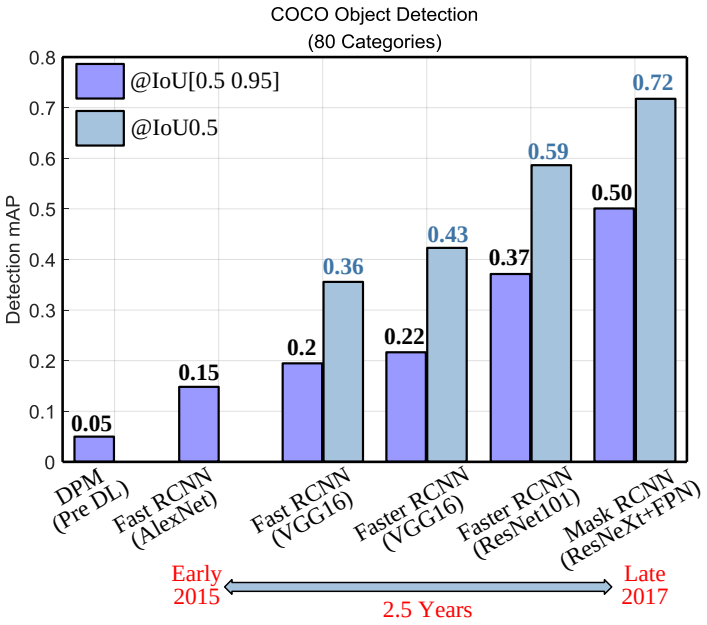


Fig. 17 Evolution of object detection performance on COCO (Test-Dev results). Results are quoted from [64, 80, 176] accordingly. The backbone network, the design of detection framework and the availability of good and large scale datasets are the three most important factors in detection.

- Meta detection frameworks, such as RCNN [65], Fast RCNN [64], Faster RCNN [175], RFCN [40], Mask RCNN [80], YOLO [174] and SSD [136];
- Backbone networks such as VGG [191], Inception [200, 99, 201], ResNet [79], ResNeXt [223], Xception [35] and DetNet [127] *etc.* listed in Table 2;
- Innovations such as multilayer feature combination [130, 190, 58], deformable convolutional networks [41], deformable RoI pooling [160, 41], heavier heads [177, 164], and lighter heads [128];
- Pretraining with datasets such as ImageNet [179], COCO [129], Places [245], JFT [82] and Open Images [106]
- Different detection proposal methods and different numbers of object proposals;
- Train/test data augmentation "tricks" such as multicrop, horizontal flipping, multiscale images and novel multiscale training strategies [192, 193] *etc*, mask tightening, and model ensembling.

Although it may be impractical to compare every recently proposed detector, it is nevertheless highly valuable to integrate representative and publicly available detectors into a common platform and to compare them in a unified manner. There has been very limited work in this regard, except for Huang's study [96] of the trade off between accuracy and speed of three main families of detectors (Faster RCNN [175], RFCN [40] and SSD [136]) by varying the backbone network, image resolution, and the number of box proposals *etc*.

As can be seen from Tables 3, 4, 5, 6 and Table 10, we have summarized the best reported performance of many methods on three widely used standard benchmarks. The results of these methods were reported on the same test benchmark, despite their differing in one or more of the aspects listed above.

Figs. 1 and 17 present a very brief overview of the state of the art, summarizing the best detection results of the PASCAL VOC, ILSVRC and MSCOCO challenges. More results can be found at detection challenge websites [98, 148, 163]. In summary, the backbone network, the detection framework design and the availability of large scale datasets are the three most important factors in detection. Furthermore ensembles of multiple models, the incorporation of context features, and data augmentation all help to achieve better accuracy.

In less than five years, since AlexNet [109] was proposed, the Top5 error on ImageNet classification [179] with 1000 classes has dropped from 16% to 2%, as shown in Fig. 9. However, the mAP of the best performing detector [164] (which is only trained to detect 80 classes) on COCO [129] has reached 73%, even at 0.5 IoU, illustrating clearly how object detection is much harder than image classification. The accuracy level achieved by the state of the art detectors is far from satisfying the requirements of general purpose practical applications, so there remains significant room for future improvement.

6 Conclusions

Generic object detection is an important and challenging problem in computer vision, and has received considerable attention. Thanks to remarkable development of deep learning techniques, the field of object detection has dramatically evolved. As a comprehensive survey on deep learning for generic object detection, this paper has highlighted the recent achievements, provided a structural taxonomy for methods according to their roles in detection, summarized existing popular datasets and evaluation criteria, and discussed performance for the most representative methods.

Despite the tremendous successes achieved in the past several years (*e.g.* detection accuracy improving significantly from 23% in ILSVRC2013 to 73% in ILSVRC2017), there remains a huge gap between the state-of-the-art and human-level performance, especially in terms of open world learning. Much work remains to be done, which we see focused on the following eight domains:

(1) Open World Learning: The ultimate goal is to develop object detection systems that are capable of accurately and efficiently recognizing and localizing instances of all object categories (thousands or more object classes [43]) in all open world scenes, competing with the human visual system. Recent object detection algorithms are learned with limited datasets [53, 54, 129, 179], recognizing and localizing the object categories included in the dataset, but blind, in principle, to other object categories outside the dataset, although ideally a powerful detection system should be able to recognize novel object categories [112, 73]. Current detection datasets [53, 179, 129] contain only dozens to hundreds of categories, which is significantly smaller than those which can be recognized by humans. To achieve this goal, new large-scale labeled datasets with significantly more categories for generic object detection will need to be developed, since the state of the art in CNNs require extensive data to train well. However collecting such massive amounts of data, particularly bounding box labels for object detection, is very expensive, especially for hundreds of thousands categories.

(2) Better and More Efficient Detection Frameworks: One of the factors for the tremendous successes in generic object detection has been the development of better detection frameworks, both region-based (RCNN [65], Fast RCNN [64], Faster RCNN [175], Mask RCNN [80]) and one-state detectors (YOLO [174], SSD [136]). Region-based detectors have the highest accuracy, but are too computationally intensive for embedded or real-time systems. One-stage detectors have the potential to be faster and simpler, but have not yet reached the accuracy of region-based detectors. One possible limitation is that the state of the art object detectors depend heavily on the underlying backbone network, which have been initially optimized for image classification, causing a learning bias due to the differences between classification and detection, such that one potential strategy is to learn object detectors from scratch, like the DSOD detector [186].

(3) Compact and Efficient Deep CNN Features: Another significant factor in the considerable progress in generic object detection has been the development of powerful deep CNNs, which have increased remarkably in depth, from several layers (*e.g.,* AlexNet [110]) to hundreds of layers (*e.g.,* ResNet [79], DenseNet [94]). These networks have millions to hundreds of millions of parameters, requiring massive data and power-hungry GPUs for training, again limiting their application to real-time / embedded applications. In response, there has been growing research interest in designing compact and lightweight networks [25, 4, 95, 88, 132, 231], network compression and acceleration [34, 97, 195, 121, 124], and network interpretation and understanding [19, 142, 146].

(4) Robust Object Representations: One important factor which makes the object recognition problem so challenging is the great variability in real-world images, including viewpoint and lighting changes, object scale, object pose, object part deformations, background clutter, occlusions, changes in appearance, image blur, image resolution, noise, and camera limitations and distortions. Despite the advances in deep networks, they are still limited by a lack of robustness to these many variations [134, 24], which significantly constrains the usability for real-world applications.

(5) Context Reasoning: Real-world objects typically coexist with other objects and environments. It has been recognized that contextual information (object relations, global scene statistics) helps object detection and recognition [155], especially in situations of small or occluded objects or poor image quality. There was extensive work preceding deep learning [143, 152, 171, 47, 59], however since the deep learning era there has been only very limited progress in exploiting contextual information [29, 62, 90]. How to efficiently and effectively incorporate contextual information remains to be explored, ideally guided by how humans are quickly able to guide their attention to objects of interest in natural scenes.

(6) Object Instance Segmentation: Continuing the trend of moving towards a richer and more detailed understanding image content (*e.g.,* from image classification to single object localization to object detection), a next challenge would be to tackle pixel-level object instance segmentation [129, 80, 93], as object instance segmentation can play an important role in many potential applications that require the precise boundaries of individual instances.

(7) Weakly Supervised or Unsupervised Learning: Current state of the art detectors employ fully-supervised models learned from labelled data with object bounding boxes or segmentation masks [54, 129, 179, 129], however such fully supervised learning has serious limitations, where the assumption of bounding box annotations may become problematic, especially when the number of object categories is large. Fully supervised learning is not scalable in the absence of fully labelled training data, therefore it is valuable to study how the power of CNNs can be leveraged in weakly supervised or unsupervised detection [15, 45, 187].

(8) 3D Object Detection: The progress of depth cameras has enabled the acquisition of depth information in the form of RGB-D images or 3D point clouds. The depth modality can be employed to help object detection and recognition, however there is only limited work in this direction [30, 165, 220], but which might benefit from taking advantage of large collections of high quality CAD models [219].

The research field of generic object detection is still far from complete; given the massive algorithmic breakthroughs over the past five years, we remain optimistic of the opportunities over the next five years.

7 Acknowledgments

The authors would like to thank the pioneer researchers in generic object detection and other related fields. The authors would also like to express their sincere appreciation to Professor Jiří Matas, the associate editor and the reviewers for their comments and suggestions. This work has been supported by the Center for Machine Vision and Signal Analysis at the University of Oulu (Finland) and the National Natural Science Foundation of China under Grant 61872379.

References

1. Agrawal P., Girshick R., Malik J. (2014) Analyzing the performance of multilayer neural networks for object recognition. In: ECCV, pp. 329–344 11
2. Alexe B., Deselaers T., Ferrari V. (2010) What is an object? In: CVPR, pp. 73–80 16
3. Alexe B., Deselaers T., Ferrari V. (2012) Measuring the objectness of image windows. IEEE TPAMI 34(11):2189–2202 16
4. Alvarez J., Salzmann M. (2016) Learning the number of neurons in deep networks. In: NIPS, pp. 2270–2278 24
5. Andreopoulos A., Tsotsos J. (2013) 50 years of object recognition: Directions forward. Computer Vision and Image Understanding 117(8):827–891 2, 3, 4
6. Arbeláez P., Hariharan B., Gu C., Gupta S., Bourdev L., Malik J. (2012) Semantic segmentation using regions and parts. In: CVPR, pp. 3378–3385 16
7. Arbeláez P., Pont-Tuset J., Barron J., Marques F., Malik J. (2014) Multiscale combinatorial grouping. In: CVPR, pp. 328–335 16, 17
8. Azizpour H., Razavian A., Sullivan J., Maki A., Carlsson S. (2016) Factors of transferability for a generic convnet representation. IEEE TPAMI 38(9):1790–1802 11
9. Bar M. (2004) Visual objects in context. Nature Reviews Neuroscience 5(8):617–629 15
10. Bell S., Lawrence Z., Bala K., Girshick R. (2016) Inside Outside Net: Detecting objects in context with skip pooling and recurrent neural networks. In: CVPR, pp. 2874–2883 12, 13, 15, 16
11. Belongie S., Malik J., Puzicha J. (2002) Shape matching and object recognition using shape contexts. IEEE TPAMI 24(4):509–522 5
12. Bengio Y., Courville A., Vincent P. (2013) Representation learning: A review and new perspectives. IEEE TPAMI 35(8):1798–1828 2, 3, 10
13. Biederman I. (1972) Perceiving real world scenes. IJCV 177(7):77–80 15

14. Biederman I. (1987) Recognition by components: a theory of human image understanding. Psychological review 94(2):115 6
15. Bilen H., Vedaldi A. (2016) Weakly supervised deep detection networks. In: CVPR, pp. 2846–2854 24
16. Bodla N., Singh B., Chellappa R., Davis L. S. (2017) SoftNMS improving object detection with one line of code. In: ICCV, pp. 5562–5570 19
17. Borji A., Cheng M., Jiang H., Li J. (2014) Salient object detection: A survey. arXiv: 14115878v1 1:1–26 3
18. Bruna J., Mallat S. (2013) Invariant scattering convolution networks. IEEE TPAMI 35(8):1872–1886 14
19. Bruna J., Mallat S. (2013) Invariant scattering convolution networks. IEEE transactions on pattern analysis and machine intelligence 35(8):1872–1886 24
20. Cai Z., Fan Q., Feris R., Vasconcelos N. (2016) A unified multiscale deep convolutional neural network for fast object detection. In: ECCV, pp. 354–370 12, 13
21. Carreira J., Sminchisescu C. (2012) CMPC: Automatic object segmentation using constrained parametric mincuts. IEEE TPAMI 34(7):1312–1328 16
22. Chatfield K., Simonyan K., Vedaldi A., Zisserman A. (2014) Return of the devil in the details: Delving deep into convolutional nets. In: BMVC 19
23. Chavali N., Agrawal H., Mahendru A., Batra D. (2016) Object proposal evaluation protocol is gameable. In: CVPR, pp. 835–844 16
24. Chellappa R. (2016) The changing fortunes of pattern recognition and computer vision. Image and Vision Computing 55:3–5 24
25. Chen G., Choi W., Yu X., Han T., Chandraker M. (2017) Learning efficient object detection models with knowledge distillation. In: NIPS 24
26. Chen L., Papandreou G., Kokkinos I., Murphy K., Yuille A. (2015) Semantic image segmentation with deep convolutional nets and fully connected CRFs. In: ICLR 15
27. Chen L., Papandreou G., Kokkinos I., Murphy K., Yuille A. (2018) DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE TPAMI 40(4):834–848 12, 15
28. Chen Q., Song Z., Dong J., Huang Z., Hua Y., Yan S. (2015) Contextualizing object detection and classification. IEEE TPAMI 37(1):13–27 15
29. Chen X., Gupta A. (2017) Spatial memory for context reasoning in object detection. In: ICCV 15, 16, 24
30. Chen X., Kundu K., Zhu Y., Berneshawi A. G., Ma H., Fidler S., Urtasun R. (2015) 3d object proposals for accurate object class detection. In: NIPS, pp. 424–432 24
31. Chen Y., Li J., Xiao H., Jin X., Yan S., Feng J. (2017) Dual path networks. In: NIPS, pp. 4467–4475 11, 20
32. Cheng G., Zhou P., Han J. (2016) RIFD-CNN: Rotation invariant and fisher discriminative convolutional neural networks for object detection. In: CVPR, pp. 2884–2893 14
33. Cheng M., Zhang Z., Lin W., Torr P. (2014) BING: Binarized normed gradients for objectness estimation at 300fps. In: CVPR, pp. 3286–3293 16
34. Cheng Y., Wang D., Zhou P., Zhang T. (2018) Model compression and acceleration for deep neural networks: The principles, progress, and challenges. IEEE Signal Processing Magazine 35(1):126–136 24
35. Chollet F. (2017) Xception: Deep learning with depthwise separable convolutions. In: CVPR, pp. 1800–1807 11, 23
36. Cinbis R., Verbeek J., Schmid C. (2017) Weakly supervised object localization with multi-fold multiple instance learning. IEEE TPAMI 39(1):189–203 7
37. Csurka G., Dance C., Fan L., Willamowski J., Bray C. (2004) Visual categorization with bags of keypoints. In: ECCV Workshop on statistical learning in computer vision 2, 5, 16
38. Dai J., He K., Li Y., Ren S., Sun J. (2016) Instance sensitive fully convolutional networks. In: ECCV, pp. 534–549 18
39. Dai J., He K., Sun J. (2016) Instance aware semantic segmentation via multitask network cascades. In: CVPR, pp. 3150–3158 18
40. Dai J., Li Y., He K., Sun J. (2016) RFCN: object detection via region based fully convolutional networks. In: NIPS, pp. 379–387 6, 8, 12, 16, 23, 30
41. Dai J., Qi H., Xiong Y., Li Y., Zhang G., Hu H., Wei Y. (2017) Deformable convolutional networks. In: ICCV 13, 14, 15, 20, 23
42. Dalal N., Triggs B. (2005) Histograms of oriented gradients for human detection. In: CVPR, vol 1, pp. 886–893 2, 5, 6, 10, 16
43. Dean T., Ruzon M., Segal M., Shlens J., Vijayanarasimhan S., Yagnik J. (2013) Fast, accurate detection of 100,000 object classes on a single machine. In: CVPR, pp. 1814–1821 23
44. Deng J., Dong W., Socher R., Li L., Li K., Li F. (2009) ImageNet: A large scale hierarchical image database. In: CVPR, pp. 248–255 5, 11, 19
45. Diba A., Sharma V., Pazandeh A. M., Pirsiavash H., Van Gool L. (2017) Weakly supervised cascaded convolutional networks. In: CVPR, vol 3, p. 9 24
46. Dickinson S., Leonardis A., Schiele B., Tarr M. (2009) The Evolution of Object Categorization and the Challenge of Image Abstraction in *Object Categorization: Computer and Human Vision Perspectives*. Cambridge University Press 2, 3, 10
47. Divvala S., Hoiem D., Hays J., Efros A., Hebert M. (2009) An empirical study of context in object detection. In: CVPR, pp. 1271–1278 15, 24
48. Dollar P., Wojek C., Schiele B., Perona P. (2012) Pedestrian detection: An evaluation of the state of the art. IEEE TPAMI 34(4):743–761 2, 3
49. Donahue J., Jia Y., Vinyals O., Hoffman J., Zhang N., Tzeng E., Darrell T. (2014) DeCAF: A deep convolutional activation feature for generic visual recognition. In: ICML, vol 32, pp. 647–655 11
50. Endres I., Hoiem D. (2010) Category independent object proposals 16
51. Enzweiler M., Gavrila D. M. (2009) Monocular pedestrian detection: Survey and experiments. IEEE TPAMI 31(12):2179–2195 2, 3
52. Erhan D., Szegedy C., Toshev A., Anguelov D. (2014) Scalable object detection using deep neural networks. In: CVPR, pp. 2147–2154 6, 17, 18
53. Everingham M., Gool L. V., Williams C., Winn J., Zisserman A. (2010) The pascal visual object classes (voc) challenge. IJCV 88(2):303–338 1, 2, 4, 5, 17, 19, 20, 21, 22, 23
54. Everingham M., Eslami S., Gool L. V., Williams C., Winn J., Zisserman A. (2015) The pascal visual object classes challenge: A retrospective. IJCV 111(1):98–136 2, 19, 20, 21, 22, 23, 24
55. Felzenszwalb P., McAllester D., Ramanan D. (2008) A discriminatively trained, multiscale, deformable part model. In: CVPR, pp. 1–8 6, 16
56. Felzenszwalb P., Girshick R., McAllester D., Ramanan D. (2010) Object detection with discriminatively trained part based models. IEEE TPAMI 32(9):1627–1645 2, 6, 12, 14
57. Fischler M., Elschlager R. (1973) The representation and matching of pictorial structures. IEEE Transactions on computers 100(1):67–92 5
58. Fu C.-Y., Liu W., Ranga A., Tyagi A., Berg A. C. (2017) DSSD: Deconvolutional single shot detector. In: arXiv preprint arXiv:1701.06659 12, 13, 23
59. Galleguillos C., Belongie S. (2010) Context based object categorization: A critical survey. Computer Vision and Image Understanding 114:712–722 2, 3, 15, 24
60. Geronimo D., Lopez A. M., Sappa A. D., Graf T. (2010) Survey of pedestrian detection for advanced driver assistance systems. IEEE TPAMI 32(7):1239–1258 2, 3
61. Ghodrati A., Diba A., Pedersoli M., Tuytelaars T., Van Gool L. (2015) DeepProposal: Hunting objects by cascading deep convolutional layers. In: ICCV, pp. 2578–2586 17, 18
62. Gidaris S., Komodakis N. (2015) Object detection via a multiregion and semantic segmentation aware CNN model. In: ICCV, pp. 1134–1142 10, 15, 16, 17, 24
63. Gidaris S., Komodakis N. (2016) Attend refine repeat: Active box proposal generation via in out localization. In: BMVC 13
64. Girshick R. (2015) Fast R-CNN. In: ICCV, pp. 1440–1448 2, 6, 8, 10, 11, 12, 17, 19, 23, 24, 30
65. Girshick R., Donahue J., Darrell T., Malik J. (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR, pp. 580–587 2, 5, 6, 7, 8, 9, 10, 11, 12, 17, 19, 23, 24, 30
66. Girshick R., Iandola F., Darrell T., Malik J. (2015) Deformable part models are convolutional neural networks. In: CVPR, pp. 437–446 14
67. Girshick R., Donahue J., Darrell T., Malik J. (2016) Region-based convolutional networks for accurate object detection and segmentation. IEEE TPAMI 38(1):142–158 6, 7, 11
68. Grauman K., Darrell T. (2005) The pyramid match kernel: Discriminative classification with sets of image features. In: ICCV, vol 2, pp. 1458–1465 6

69. Grauman K., Leibe B. (2011) Visual object recognition. Synthesis lectures on artificial intelligence and machine learning 5(2):1–181 1, 2, 3
70. Griffin G., Holub A., Perona P. (2007) Caltech 256 object category dataset. In: California Institute of Technology Technique Report 7694 19, 20
71. Gu J., Wang Z., Kuen J., Ma L., Shahroudy A., Shuai B., Liu T., Wang X., Wang G., Cai J., Chen T. (2017) Recent advances in convolutional neural networks. Pattern Recognition pp. 1–24 2, 3, 10
72. Guillaumin M., Küttel D., Ferrari V. (2014) Imagenet autoannotation with segmentation propagation. International Journal of Computer Vision 110(3):328–348 16
73. Hariharan B., Girshick R. B. (2017) Low shot visual recognition by shrinking and hallucinating features. In: ICCV, pp. 3037–3046 23
74. Hariharan B., Arbeláez P., Girshick R., Malik J. (2014) Simultaneous detection and segmentation. In: ECCV, pp. 297–312 18
75. Hariharan B., Arbeláez P., Girshick R., Malik J. (2016) Object instance segmentation and fine grained localization using hypercolumns. IEEE TPAMI 7, 9, 12
76. Harzallah H., Jurie F., Schmid C. (2009) Combining efficient object localization and image classification. In: ICCV, pp. 237–244 6, 16
77. He K., Zhang X., Ren S., Sun J. (2014) Spatial pyramid pooling in deep convolutional networks for visual recognition. In: ECCV, pp. 346–361 2, 6, 11, 12, 30
78. He K., Zhang X., Ren S., Sun J. (2015) Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In: ICCV, pp. 1026–1034 9
79. He K., Zhang X., Ren S., Sun J. (2016) Deep residual learning for image recognition. In: CVPR, pp. 770–778 2, 8, 10, 11, 23, 24
80. He K., Gkioxari G., Dollár P., Girshick R. (2017) Mask RCNN. In: ICCV 8, 14, 15, 17, 23, 24, 30
81. Hinton G., Salakhutdinov R. (2006) Reducing the dimensionality of data with neural networks. science 313(5786):504–507 1
82. Hinton G., Vinyals O., Dean J. (2015) Distilling the knowledge in a neural network. arXiv:150302531 23
83. Hochreiter S., Schmidhuber J. (1997) Long short term memory. Neural Computation 9(8):1735–1780 15
84. Hoiem D., Chodpathumwan Y., Dai Q. (2012) Diagnosing error in object detectors. In: ECCV, pp. 340–353 21
85. Hosang J., Omran M., Benenson R., Schiele B. (2015) Taking a deeper look at pedestrians. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4073–4082 2
86. Hosang J., Benenson R., Dollr P., Schiele B. (2016) What makes for effective detection proposals? IEEE TPAMI 38(4):814–829 16, 17
87. Hosang J., Benenson R., Schiele B. (2017) Learning nonmaximum suppression. In: ICCV 19
88. Howard A., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M., Adam H. (2017) Mobilenets: Efficient convolutional neural networks for mobile vision applications. In: CVPR 11, 24
89. Hu H., Lan S., Jiang Y., Cao Z., Sha F. (2017) FastMask: Segment multiscale object candidates in one shot. In: CVPR, pp. 991–999 18, 19
90. Hu H., Gu J., Zhang Z., Dai J., Wei Y. (2018) Relation networks for object detection. In: CVPR 15, 16, 24
91. Hu J., Shen L., Sun G. (2018) Squeeze and excitation networks. In: CVPR 10, 11
92. Hu P., Ramanan D. (2017) Finding tiny faces. In: CVPR, pp. 1522–1530 2
93. Hu R., Dollár P., He K., Darrell T., Girshick R. (2018) Learning to segment every thing. In: CVPR 24
94. Huang G., Liu Z., Weinberger K. Q., van der Maaten L. (2017) Densely connected convolutional networks. In: CVPR 10, 11, 14, 24
95. Huang G., Liu S., van der Maaten L., Weinberger K. (2018) CondenseNet: An efficient densenet using learned group convolutions. In: CVPR 24
96. Huang J., Rathod V., Sun C., Zhu M., Korattikara A., Fathi A., Fischer I., Wojna Z., Song Y., Guadarrama S., Murphy K. (2017) Speed/accuracy trade-offs for modern convolutional object detectors 11, 19, 23
97. Hubara I., Courbariaux M., Soudry D., ElYaniv R., Bengio Y. (2016) Binarized neural networks. In: NIPS, pp. 4107–4115 24
98. ILSVRC detection challenge results (2018) http://www.image-net.org/challenges/LSVRC/ 23
99. Ioffe S., Szegedy C. (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448–456 10, 11, 23
100. Jaderberg M., Simonyan K., Zisserman A., et al. (2015) Spatial transformer networks. In: NIPS, pp. 2017–2025 14
101. Kim A., Sharma A., Jacobs D. (2014) Locally scale invariant convolutional neural networks. In: NIPS 14
102. Kim K., Hong S., Roh B., Cheon Y., Park M. (2016) PVANet: Deep but lightweight neural networks for real time object detection. In: NIPSW 13
103. Kong T., Yao A., Chen Y., Sun F. (2016) HyperNet: towards accurate region proposal generation and joint object detection. In: CVPR, pp. 845–853 12, 13, 17, 18
104. Kong T., Sun F., Yao A., Liu H., Lu M., Chen Y. (2017) RON: Reverse connection with objectness prior networks for object detection. In: CVPR 12, 13
105. Krähenbühl1 P., Koltun V. (2014) Geodesic object proposals. In: ECCV 16
106. Krasin I., Duerig T., Alldrin N., Ferrari V., AbuElHaija S., Kuznetsova A., Rom H., Uijlings J., Popov S., Kamali S., Malloci M., PontTuset J., Veit A., Belongie S., Gomes V., Gupta A., Sun C., Chechik G., Cai D., Feng Z., Narayanan D., Murphy K. (2017) OpenImages: A public dataset for large scale multilabel and multiclass image classification. Dataset available from https://storagegoogleapiscom/openimages/web/indexhtml 19, 20, 23
107. Krishna R., Zhu Y., Groth O., Johnson J., Hata K., Kravitz J., Chen S., Kalantidis Y., Li L., Shamma D., Bernstein M., FeiFei L. (2017) Visual genome: Connecting language and vision using crowdsourced dense image annotations. IJCV 123(1):32–73 19
108. Krizhevsky A. (2009) Learning multiple layers of features from tiny images. Master's thesis, University of Toronto 19
109. Krizhevsky A., Sutskever I., Hinton G. (2012) ImageNet classification with deep convolutional neural networks. In: NIPS, pp. 1097–1105 2, 5, 6, 9, 17, 23
110. Krizhevsky A., Sutskever I., Hinton G. (2012) ImageNet classification with deep convolutional neural networks. In: NIPS, pp. 1097–1105 10, 11, 24
111. Kuo W., Hariharan B., Malik J. (2015) DeepBox: Learning objectness with convolutional networks. In: ICCV, pp. 2479–2487 17, 18
112. Lake B., Salakhutdinov R., Tenenbaum J. (2015) Human level concept learning through probabilistic program induction. Science 350(6266):1332–1338 23
113. Lampert C. H., Blaschko M. B., Hofmann T. (2008) Beyond sliding windows: Object localization by efficient subwindow search. In: CVPR, pp. 1–8 6
114. Lazebnik S., Schmid C., Ponce J. (2006) Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR, vol 2, pp. 2169–2178 2, 5, 6, 19, 20
115. LeCun Y., Bottou L., Bengio Y., Haffner P. (1998) Gradient based learning applied to document recognition. Proceedings of the IEEE 86(11):2278–2324 2, 20
116. LeCun Y., Bengio Y., Hinton G. (2015) Deep learning. Nature 521:436–444 1, 2, 3, 10
117. Lenc K., Vedaldi A. (2015) R-CNN minus R. In: BMVC15 7, 30
118. Lenc K., Vedaldi A. (2018) Understanding image representations by measuring their equivariance and equivalence. IJCV 14
119. Li F., Fergus R., Perona P. (2004) Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: CVPR Workshop on Generative Model Based Vision 19, 20
120. Li H., Lin Z., Shen X., Brandt J., Hua G. (2015) A convolutional neural network cascade for face detection. In: CVPR, pp. 5325–5334 2
121. Li H., Kadav A., Durdanovic I., Samet H., Graf H. P. (2017) Pruning filters for efficient convnets. In: ICLR 24
122. Li H., Liu Y., Ouyang W., XiaogangWang (2018) Zoom out and in network with map attention decision for region proposal and object detection. IJCV 12, 13, 17, 18
123. Li J., Wei Y., Liang X., Dong J., Xu T., Feng J., Yan S. (2017) Attentive contexts for object detection. IEEE Transactions on Multimedia 19(5):944–954 15, 16, 17
124. Li Q., Jin S., Yan J. (2017) Mimicking very efficient network for object detection. In: CVPR, pp. 7341–7349 24

125. Li Y., Wang S., Tian Q., Ding X. (2015) Feature representation for statistical learning based object detection: A review. Pattern Recognition 48(11):3542–3559 3
126. Li Y., Qi H., Dai J., Ji X., Wei Y. (2017) Fully convolutional instance aware semantic segmentation. In: CVPR, pp. 4438–4446 18
127. Li Z., Peng C., Yu G., Zhang X., Deng Y., Sun J. (2018) DetNet: A backbone network for object detection. In: ECCV 11, 23
128. Li Z., Peng C., Yu G., Zhang X., Deng Y., Sun J. (2018) Light head RCNN: In defense of two stage object detector. In: CVPR 8, 23
129. Lin T., Maire M., Belongie S., Hays J., Perona P., Ramanan D., Dollár P., Zitnick L. (2014) Microsoft COCO: Common objects in context. In: ECCV, pp. 740–755 2, 4, 5, 17, 19, 20, 22, 23, 24
130. Lin T., Dollár P., Girshick R., He K., Hariharan B., Belongie S. (2017) Feature pyramid networks for object detection. In: CVPR 8, 12, 13, 14, 23
131. Lin T., Goyal P., Girshick R., He K., Dollár P. (2017) Focal loss for dense object detection. In: ICCV 14, 19, 20
132. Lin X., Zhao C., Pan W. (2017) Towards accurate binary convolutional neural network. In: NIPS, pp. 344–352 24
133. Litjens G., Kooi T., Bejnordi B., Setio A., Ciompi F., Ghafoorian M., J. van der Laak B. v., Sánchez C. (2017) A survey on deep learning in medical image analysis. Medical Image Analysis 42:60–88 2, 3
134. Liu L., Fieguth P., Guo Y., Wang X., Pietikäinen M. (2017) Local binary features for texture classification: Taxonomy and experimental study. Pattern Recognition 62:135–160 24
135. Liu S., Huang D., Wang Y. (2018) Receptive field block net for accurate and fast object detection. In: CVPR 12, 13
136. Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C., Berg A. (2016) SSD: single shot multibox detector. In: ECCV, pp. 21–37 9, 10, 12, 15, 18, 19, 23, 24, 30
137. Liu Y., Wang R., Shan S., Chen X. (2018) Structure Inference Net: Object detection using scene level context and instance level relationships. In: CVPR, pp. 6985–6994 15, 16
138. Long J., Shelhamer E., Darrell T. (2015) Fully convolutional networks for semantic segmentation. In: CVPR, pp. 3431–3440 7, 8, 12, 18
139. Lowe D. (1999) Object recognition from local scale invariant features. In: ICCV, vol 2, pp. 1150–1157 2, 5, 10
140. Lowe D. (2004) Distinctive image features from scale-invariant keypoints. IJCV 60(2):91–110 2, 5, 16
141. Lu Y., Javidi T., Lazebnik S. (2016) Adaptive object detection using adjacency and zoom prediction. In: CVPR, pp. 2351–2359 17, 18
142. Mahendran A., Vedaldi A. (2016) Visualizing deep convolutional neural networks using natural preimages. International Journal of Computer Vision 120(3):233–255 24
143. Malisiewicz T., Efros A. (2009) Beyond categories: The visual memex model for reasoning about object relationships. In: NIPS 15, 24
144. Manen S., Guillaumin M., Van Gool L. (2013) Prime object proposals with randomized prim's algorithm. In: CVPR, pp. 2536–2543 16
145. Mikolajczyk K., Schmid C. (2005) A performance evaluation of local descriptors. IEEE TPAMI 27(10):1615–1630 5
146. Montavon G., Samek W., Müller K.-R. (2018) Methods for interpreting and understanding deep neural networks. Digital Signal Processing 73:1–15 24
147. Mordan T., Thome N., Henaff G., Cord M. (2018) End to end learning of latent deformable part based representations for object detection. IJCV pp. 1–21 13, 14, 15
148. MS COCO detection leaderboard (2018) http://cocodataset.org/#detection-leaderboard 23
149. Mundy J. (2006) Object recognition in the geometric era: A retrospective. in book Toward Category Level Object Recognition edited by J Ponce, M Hebert, C Schmid and A Zisserman pp. 3–28 5
150. Murase H., Nayar S. (1995) Visual learning and recognition of 3D objects from appearance. IJCV 14(1):5–24 5
151. Murase H., Nayar S. (1995) Visual learning and recognition of 3d objects from appearance. IJCV 14(1):5–24 5
152. Murphy K., Torralba A., Freeman W. (2003) Using the forest to see the trees: a graphical model relating features, objects and scenes. In: NIPS 15, 24
153. Ojala T., Pietikäinen M., Maenpää T. (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE TPAMI 24(7):971–987 5, 16
154. Olaf Ronneberger T. B. Philipp Fischer (2015) UNet: convolutional networks for biomedical image segmentation. In: MICCAI, pp. 234–241 12
155. Oliva A., Torralba A. (2007) The role of context in object recognition. Trends in cognitive sciences 11(12):520–527 15, 24
156. Opelt A., Pinz A., Fussenegger M., Auer P. (2006) Generic object recognition with boosting. IEEE TPAMI 28(3):416–431 4
157. Oquab M., Bottou L., Laptev I., Sivic J. (2014) Learning and transferring midlevel image representations using convolutional neural networks. In: CVPR, pp. 1717–1724 10
158. Oquab M., Bottou L., Laptev I., Sivic J. (2015) Is object localization for free? weakly supervised learning with convolutional neural networks. In: CVPR, pp. 685–694 7
159. Osuna E., Freund R., Girosit F. (1997) Training support vector machines: an application to face detection. In: CVPR, pp. 130–136 5
160. Ouyang W., Wang X., Zeng X., Qiu S., Luo P., Tian Y., Li H., Yang S., Wang Z., Loy C.-C., et al. (2015) DeepIDNet: Deformable deep convolutional neural networks for object detection. In: CVPR, pp. 2403–2412 6, 13, 14, 15, 16, 17, 23
161. Ouyang W., Zeng X., Wang X., Qiu S., Luo P., Tian Y., Li H., Yang S., Wang Z., Li H., Wang K., Yan J., Loy C. C., Tang X. (2017) DeepIDNet: Object detection with deformable part based convolutional neural networks. IEEE TPAMI 39(7):1320–1334 11, 14
162. Parikh D., Zitnick C., Chen T. (2012) Exploring tiny images: The roles of appearance and contextual information for machine and human object recognition. IEEE TPAMI 34(10):1978–1991 15
163. PASCAL VOC detection leaderboard (2018) http://host.robots.ox.ac.uk:8080/leaderboard/main_bootstrap.php 23
164. Peng C., Xiao T., Li Z., Jiang Y., Zhang X., Jia K., Yu G., Sun J. (2018) MegDet: A large minibatch object detector. In: CVPR 20, 23
165. Pepik B., Benenson R., Ritschel T., Schiele B. (2015) What is holding back convnets for detection? In: German Conference on Pattern Recognition, pp. 517–528 24
166. Perronnin F., Sánchez J., Mensink T. (2010) Improving the fisher kernel for large scale image classification. In: ECCV, pp. 143–156 2, 5, 10
167. Pinheiro P., Collobert R., Dollar P. (2015) Learning to segment object candidates. In: NIPS, pp. 1990–1998 17, 18
168. Pinheiro P., Lin T., Collobert R., Dollár P. (2016) Learning to refine object segments. In: ECCV, pp. 75–91 12, 13, 18, 19
169. Ponce J., Hebert M., Schmid C., Zisserman A. (2007) Toward Category Level Object Recognition. Springer 2, 3, 5
170. Qiao S., Shen W., Qiu W., Liu C., Yuille A. (2017) ScaleNet: guiding object proposal generation in supermarkets and beyond. In: ICCV 18, 19
171. Rabinovich A., Vedaldi A., Galleguillos C., Wiewiora E., Belongie S. (2007) Objects in context. In: ICCV 15, 24
172. Razavian R., Azizpour H., Sullivan J., Carlsson S. (2014) CNN features off the shelf: an astounding baseline for recognition. In: CVPR Workshops, pp. 806–813 11
173. Redmon J., Farhadi A. (2017) YOLO9000: Better, faster, stronger. In: CVPR 9, 11, 30
174. Redmon J., Divvala S., Girshick R., Farhadi A. (2016) You only look once: Unified, real-time object detection. In: CVPR, pp. 779–788 9, 10, 11, 12, 23, 24, 30
175. Ren S., He K., Girshick R., Sun J. (2015) Faster R-CNN: Towards real time object detection with region proposal networks. In: NIPS, pp. 91–99 6, 7, 8, 9, 10, 12, 15, 17, 18, 23, 24, 30
176. Ren S., He K., Girshick R., Sun J. (2017) Faster RCNN: Towards real time object detection with region proposal networks. IEEE TPAMI 39(6):1137–1149 2, 7, 18, 23
177. Ren S., He K., Girshick R., Zhang X., Sun J. (2017) Object detection networks on convolutional feature maps. IEEE TPAMI 23
178. Rowley H., Baluja S., Kanade T. (1998) Neural network based face detection. IEEE TPAMI 20(1):23–38 5
179. Russakovsky O., Deng J., Su H., Krause J., Satheesh S., Ma S., Huang Z., Karpathy A., Khosla A., Bernstein M., Berg A., Li F. (2015) ImageNet large scale visual recognition challenge. IJCV 115(3):211–252 1, 2, 4, 5, 11, 17, 19, 20, 21, 22, 23, 24
180. Russell B., Torralba A., Murphy K., Freeman W. (2008) LabelMe: A database and web based tool for image annotation. IJCV 77(1-3):157–173 4

181. Schmid C., Mohr R. (1997) Local grayvalue invariants for image retrieval. IEEE TPAMI 19(5):530–535 5
182. Sermanet P., Kavukcuoglu K., Chintala S., LeCun Y. (2013) Pedestrian detection with unsupervised multistage feature learning. In: CVPR, pp. 3626–3633 15
183. Sermanet P., Eigen D., Zhang X., Mathieu M., Fergus R., LeCun Y. (2014) OverFeat: Integrated recognition, localization and detection using convolutional networks. In: ICLR 2, 6, 9, 11, 18, 30
184. Shang W., Sohn K., Almeida D., Lee H. (2016) Understanding and improving convolutional neural networks via concatenated rectified linear units. In: ICML, pp. 2217–2225 13
185. Shelhamer E., Long J., Darrell T. (????) Fully convolutional networks for semantic segmentation. IEEE TPAMI 7, 8, 12
186. Shen Z., Liu Z., Li J., Jiang Y., Chen Y., Xue X. (2017) DSOD: Learning deeply supervised object detectors from scratch. In: ICCV 12, 13, 24
187. Shi Z., Yang Y., Hospedales T., Xiang T. (2017) Weakly supervised image annotation and segmentation with objects and attributes. IEEE TPAMI 39(12):2525–2538 24
188. Shrivastava A., Gupta A. (2016) Contextual priming and feedback for Faster RCNN. In: ECCV, pp. 330–348 15, 16
189. Shrivastava A., Gupta A., Girshick R. (2016) Training region based object detectors with online hard example mining. In: CVPR, pp. 761–769 19, 20
190. Shrivastava A., Sukthankar R., Malik J., Gupta A. (2017) Beyond skip connections: Top down modulation for object detection. In: CVPR 12, 13, 23
191. Simonyan K., Zisserman A. (2015) Very deep convolutional networks for large scale image recognition. In: ICLR 2, 6, 7, 9, 10, 11, 23
192. Singh B., Davis L. (2018) An analysis of scale invariance in object detection-SNIP. In: CVPR 19, 21, 23
193. Singh B., Najibi M., Davis L. S. (2018) SNIPER: Efficient multiscale training. arXiv:180509300 19, 20, 23
194. Sivic J., Zisserman A. (2003) Video google: A text retrieval approach to object matching in videos. In: International Conference on Computer Vision (ICCV), vol 2, pp. 1470–1477 2, 5, 10, 16
195. Song Han W. J. D. Huizi Mao (2016) Deep Compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In: ICLR 24
196. Sun Z., Bebis G., Miller R. (2006) On road vehicle detection: A review. IEEE TPAMI 28(5):694–711 2, 3
197. Swain M., Ballard D. (1991) Color indexing. IJCV 7(1):11–32 5
198. Szegedy C., Toshev A., Erhan D. (2013) Deep neural networks for object detection. In: NIPS, pp. 2553–2561 6, 9
199. Szegedy C., Reed S., Erhan D., Anguelov D., Ioffe S. (2014) Scalable, high quality object detection. In: arXiv preprint arXiv:1412.1441 17
200. Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A. (2015) Going deeper with convolutions. In: CVPR, pp. 1–9 2, 9, 10, 11, 12, 13, 23
201. Szegedy C., Vanhoucke V., Ioffe S., Shlens J., Wojna Z. (2016) Rethinking the inception architecture for computer vision. In: CVPR, pp. 2818–2826 10, 11, 23
202. Szegedy C., Ioffe S., Vanhoucke V., Alemi A. (2017) Inception v4, inception resnet and the impact of residual connections on learning. AAAI pp. 4278–4284 10, 11
203. Torralba A. (2003) Contextual priming for object detection. IJCV 53(2):169–191 15
204. Torralba A F. W. Fergus R (2008) 80 million tiny images: A large dataset for non-parametric object and scene recognition. IJCV 30(11):1958–1970 19, 20
205. Turk M. A., Pentland A. (1991) Face recognition using eigenfaces. In: CVPR, pp. 586–591 5
206. Tuzel O., Porikli F., Meer P. (2006) Region covariance: A fast descriptor for detection and classification. In: ECCV, pp. 589–600 5
207. Tychsen-Smith L., Petersson L. (2018) Improving object localization with fitness nms and bounded iou loss. In: CVPR 19
208. TychsenSmith L., Petersson L. (2017) DeNet: scalable real time object detection with directed sparse sampling. In: ICCV 18
209. Uijlings J., van de Sande K., Gevers T., Smeulders A. (2013) Selective search for object recognition. IJCV 104(2):154–171 2, 6, 16, 17
210. Van de Sande K., Uijlings J., Gevers T., Smeulders A. (2011) Segmentation as selective search for object recognition. In: ICCV, pp. 1879–1886 16, 17
211. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser Ł., Polosukhin I. (2017) Attention is all you need. In: NIPS, pp. 6000–6010 15
212. Vedaldi A., Gulshan V., Varma M., Zisserman A. (2009) Multiple kernels for object detection. In: ICCV, pp. 606–613 6, 16
213. Viola P., Jones M. (2001) Rapid object detection using a boosted cascade of simple features. In: CVPR, vol 1, pp. 1–8 2, 5, 6, 16
214. Wan L., Eigen D., Fergus R. (2015) End to end integration of a convolution network, deformable parts model and nonmaximum suppression. In: CVPR, pp. 851–859 14
215. Wang X., Han T., Yan S. (2009) An HOG-LBP human detector with partial occlusion handling. In: International Conference on Computer Vision, pp. 32–39 2
216. Wang X., Shrivastava A., Gupta A. (2017) A Fast RCNN: Hard positive generation via adversary for object detection. In: CVPR 19
217. Woo S., Hwang S., Kweon I. (2018) StairNet: Top down semantic aggregation for accurate one shot detection. In: WACV, pp. 1093–1102 12, 13
218. Worrall D. E., Garbin S. J., Turmukhambetov D., Brostow G. J. (2017) Harmonic networks: Deep translation and rotation equivariance. In: CVPR, vol 2 14
219. Wu Z., Song S., Khosla A., Yu F., Zhang L., Tang X., Xiao J. (2015) 3D ShapeNets: A deep representation for volumetric shapes. In: CVPR, pp. 1912–1920 24
220. Xiang Y., Mottaghi R., Savarese S. (2014) Beyond PASCAL: A benchmark for 3D object detection in the wild. In: WACV, pp. 75–82 24
221. Xiao J., Ehinger K., Hays J., orralba A., Oliva A. (2014) SUN Database: Exploring a large collection of scene categories. IJCV 119(1):3–22 19, 20
222. Xiao R., Zhu L., Zhang H. (2003) Boosting chain learning for object detection. In: ICCV, pp. 709–715 5
223. Xie S., Girshick R., Dollár P., Tu Z., He K. (2017) Aggregated residual transformations for deep neural networks. In: CVPR 8, 11, 23
224. Yang B., Yan J., Lei Z., Li S. (2016) CRAFT objects from images. In: CVPR, pp. 6043–6051 16, 17, 18
225. Yang F., Choi W., Lin Y. (2016) Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers. In: CVPR, pp. 2129–2137 13
226. Yang M., Kriegman D., Ahuja N. (2002) Detecting faces in images: A survey. IEEE TPAMI 24(1):34–58 2, 3
227. Ye Q., Doermann D. (2015) Text detection and recognition in imagery: A survey. IEEE TPAMI 37(7):1480–1500 2, 3
228. Yosinski J., Clune J., Bengio Y., Lipson H. (2014) How transferable are features in deep neural networks? In: NIPS, pp. 3320–3328 11
229. Yu F., Koltun V. (2016) Multiscale context aggregation by dilated convolutions 12
230. Yu F., Koltun V., Funkhouser T. (2017) Dilated residual networks. In: CVPR, vol 2, p. 3 11
231. Yu R., Li A., Chen C., Lai J., et al. (2018) NISP: Pruning networks using neuron importance score propagation. CVPR 24
232. Zafeiriou S., Zhang C., Zhang Z. (2015) A survey on face detection in the wild: Past, present and future. Computer Vision and Image Understanding 138:1–24 2, 3
233. Zagoruyko S., Lerer A., Lin T., Pinheiro P., Gross S., Chintala S., Dollár P. (2016) A multipath network for object detection. In: BMVC 13, 15, 18
234. Zeiler M., Fergus R. (2014) Visualizing and understanding convolutional networks. In: ECCV, pp. 818–833 10, 11, 15
235. Zeng X., Ouyang W., Yang B., Yan J., Wang X. (2016) Gated bidirectional cnn for object detection. In: ECCV, pp. 354–369 15, 16, 17
236. Zeng X., Ouyang W., Yan J., Li H., Xiao T., Wang K., Liu Y., Zhou Y., Yang B., Wang Z., Zhou H., Wang X. (2017) Crafting gbdnet for object detection. IEEE TPAMI 15, 17
237. Zhang K., Zhang Z., Li Z., Qiao Y. (2016) Joint face detection and alignment using multitask cascaded convolutional networks. IEEE SPL 23(10):1499–1503 2
238. Zhang L., Lin L., Liang X., He K. (2016) Is faster RCNN doing well for pedestrian detection? In: ECCV, pp. 443–457 2
239. Zhang S., Wen L., Bian X., Lei Z., Li S. (2018) Single shot refinement neural network for object detection. In: CVPR 12, 13
240. Zhang X., Yang Y., Han Z., Wang H., Gao C. (2013) Object class detection: A survey. ACM Computing Surveys 46(1):10:1–10:53 1, 2, 3, 4,

15
241. Zhang Z., Qiao S., Xie C., Shen W., Wang B., Yuille A. (2018) Single shot object detection with enriched semantics. In: CVPR 12
242. Zheng S., Jayasumana S., Romera-Paredes B., Vineet V., Su Z., Du D., Huang C., Torr P. (2015) Conditional random fields as recurrent neural networks. In: ICCV, pp. 1529–1537 15
243. Zhou B., Khosla A., Lapedriza A., Oliva A., Torralba A. (2015) Object detectors emerge in deep scene CNNs. In: ICLR 7, 12
244. Zhou B., Khosla A., Lapedriza A., Oliva A., Torralba A. (2016) Learning deep features for discriminative localization. In: CVPR, pp. 2921–2929 7
245. Zhou B., Lapedriza A., Khosla A., Oliva A., Torralba A. (2017) Places: A 10 million image database for scene recognition. IEEE Trans Pattern Analysis and Machine Intelligence 11, 19, 20, 23
246. Zhou P., Ni B., Geng C., Hu J., Xu Y. (2018) Scale transferrable object detection. In: CVPR 11, 12, 13, 14
247. Zhou Y., Liu L., Shao L., Mellor M. (2016) DAVE: A unified framework for fast vehicle detection and annotation. In: ECCV, pp. 278–293 2
248. Zhou Y., Ye Q., Qiu Q., Jiao J. (2017) Oriented response networks. In: CVPR, pp. 4961–4970 14
249. Zhu X., Vondrick C., Fowlkes C., Ramanan D. (2016) Do we need more training data? IJCV 119(1):76–92 10
250. Zhu Y., Urtasun R., Salakhutdinov R., Fidler S. (2015) SegDeepM: Exploiting segmentation and context in deep neural networks for object detection. In: CVPR, pp. 4703–4711 15, 16
251. Zhu Y., Zhao C., Wang J., Zhao X., Wu Y., Lu H. (2017) CoupleNet: Coupling global structure with local parts for object detection. In: ICCV 15, 16, 17
252. Zhu Y., Zhou Y., Ye Q., Qiu Q., Jiao J. (2017) Soft proposal networks for weakly supervised object localization. In: ICCV, pp. 1841–1850 16
253. Zhu Z., Liang D., Zhang S., Huang X., Li B., Hu S. (2016) Traffic sign detection and classification in the wild. In: CVPR, pp. 2110–2118 2
254. Zitnick C., Dollár P. (2014) Edge boxes: Locating object proposals from edges. In: ECCV, pp. 391–405 16, 17

Table 10 Summarization of properties and performance of milestone detection frameworks for generic object detection. See Section 3 for detail discussion. The architectures of some methods listed in this table are illustrated in Fig. 8. The properties of the backbone DCNNs can be found in Table 2.

	Detector Name	RP	Backbone DCNN	Input ImgSize	VOC07 Results	VOC12 Results	Speed (FPS)	Published In	Source Code	Highlights and Disadvantages
Region based (Section 3.1)	RCNN [65]	SS	AlexNet	Fixed	58.5 (07)	53.3 (12)	< 0.1	CVPR14	Caffe Matlab	**Highlights:** First to integrate CNN with RP methods; Dramatic performance improvement over previous state of the art; ILSVRC2013 detection result 31.4% mAP. **Disadvantages:** Multistage pipeline of sequentially-trained (External RP computation, CNN finetuning, Each warped RP passing through CNN, SVM and BBR training); Training is expensive in space and time; Testing is slow.
	SPPNet [77]	SS	ZFNet	Arbitrary	60.9 (07)	–	< 1	ECCV14	Caffe Matlab	**Highlights:** First to introduce SPP into CNN architecture; Enable convolutional feature sharing; Accelerate RCNN evaluation by orders of magnitude without sacrificing performance; Faster than OverFeat; ILSVRC2013 detection result 35.1% mAP. **Disadvantages:** Inherit disadvantages of RCNN except the speedup; Does not result in much speedup of training; Finetuning not able to update the CONV layers before SPP layer.
	Fast RCNN [64]	SS	AlexNet VGGM VGG16	Arbitrary	70.0 (VGG) (07+12)	68.4 (VGG) (07++12)	< 1	ICCV15	Caffe Python	**Highlights:** First to enable end to end detector training (when ignoring the process of RP generation); Design a RoI pooling layer (a special case of SPP layer); Much faster and more accurate than SPPNet; No disk storage required for feature caching; **Disadvantages:** External RP computation is exposed as the new bottleneck; Still too slow for real time applications.
	Faster RCNN [175]	RPN	ZFnet VGG	Arbitrary	73.2 (VGG) (07+12)	70.4 (VGG) (07++12)	< 5	NIPS15	Caffe Matlab Python	**Highlights:** Propose RPN for generating nearly cost free and high quality RPs instead of selective search; Introduce translation invariant and multiscale anchor boxes as references in RPN; Unify RPN and Fast RCNN into a single network by sharing CONV layers; An order of magnitude faster than Fast RCNN without performance loss; Can run testing at 5 FPS with VGG16. **Disadvantages:** Training is complex, not a streamlined process; Still fall short of real time.
	RCNN⊖R [117]	New	ZFNet +SPP	Arbitrary	59.7 (07)	–	< 5	BMVC15	–	**Highlights:** Replace selective search with static RPs; Prove the possibility of building integrated, simpler and faster detectors that rely exclusively on CNN. **Disadvantages:** Fall short of real time; Decreased accuracy from not having good RPs.
	RFCN [40]	RPN	ResNet101	Arbitrary	80.5 (07+12) 83.6 (07+12+CO)	77.6 (07++12) 82.0 (07++12+CO)	< 10	NIPS16	Caffe Matlab	**Highlights:** Fully convolutional detection network; Minimize the amount of regionwise computation; Design a set of position sensitive score maps using a bank of specialized CONV layers; Faster than Faster RCNN without sacrificing much accuracy. **Disadvantages:** Training is not a streamlined process; Still fall short of real time.
	Mask RCNN [80]	RPN	ResNet101 ResNeXt101	Arbitrary	50.3 (ResNeXt101) (COCO Result)		< 5	ICCV17	Caffe Matlab Python	**Highlights:** A simple, flexible, and effective framework for object instance segmentation; Extends Faster RCNN by adding another branch for predicting an object mask in parallel with the existing branch for BB prediction; Feature Pyramid Network (FPN) is utilized; Achieved outstanding performance. **Disadvantages:** Fall short of real time applications.
Unified (Section 3.2)	OverFeat [183]	–	AlexNet like	Arbitrary	–	–	< 0.1	ICLR14	c++	**Highlights:** Enable convolutional feature sharing; Multiscale image pyramid CNN feature extraction; Win the ISLVRC2013 localization competition; Significantly faster than RCNN; ILSVRC2013 detection result 24.3% mAP. **Disadvantages:** Multistage pipeline of sequentially-trained (classifier model training, class specific localizer model finetuning); Single bounding box regressor; Cannot handle multiple object instances of the same class in an image; Too slow for real time applications.
	YOLO [174]	–	GoogLeNet like	Fixed	66.4 (07+12)	57.9 (07++12)	< 25 (VGG)	CVPR16	DarkNet	**Highlights:** First efficient unified detector; Drop RP process completely; Elegant and efficient detection framework; Significantly faster than previous detectors; YOLO runs at 45 FPS and Fast YOLO at 155 FPS; **Disadvantages:** Accuracy falls far behind state of the art detectors; Struggle to localize small objects.
	YOLOv2[173]	–	DarkNet	Fixed	78.6 (07+12)	73.5 (07++12)	< 50	CVPR17	DarkNet	**Highlights:** Propose a faster DarkNet19; Use a number of existing strategies to improve both speed and accuracy; Achieve high accuracy and high speed; YOLO9000 can detect over 9000 object categories in real time. **Disadvantages:** Not good at detecting small objects.
	SSD [136]	–	VGG16	Fixed	76.8 (07+12) 81.5 (07+12+CO)	74.9 (07++12) 80.0 (07++12+CO)	< 60	ECCV16	Caffe Python	**Highlights:** First accurate and efficient unified detector; Effectively combine ideas from RPN and YOLO to perform detection at multiscale CONV layers; Faster and significantly more accurate than YOLO; Can run at 59 FPS; **Disadvantages:** Not good at detecting small objects.

Abbreviations in this table: Region Proposal (RP), Selective Search (SS), Region Proposal Network (RPN), RCNN⊖R represents "RCNN minus R" and used a trivial RP method. Training data: "07"←VOC2007 trainval; "12"←VOC2012 trainval; "07+12"←union of 07 and VOC12 trainval; "07++12"←union of VOC07 trainval, VOC07 test, and VOC12 trainval; 07++12+CO←union of VOC07 trainval, VOC07 test, VOC12 trainval and COCO trainval. The "Speed" column roughly estimates the detection speed with a single Nvidia Titan X GPU.