# 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** Object detection, one of the most fundamental and challenging problems in computer vision, seeks to locate object instances from a large number of predefined categories in natural images. Deep learning techniques have emerged as a powerful strategy for learning feature representations directly from data and have led to remarkable breakthroughs in the field of generic object detection. Given this period 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 300 research contributions are included in this survey, covering many aspects of generic object detection: detection frameworks, object feature representation, object proposal generation, context modeling, training strategies, and evaluation metrics. We finish the survey 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 (as illustrated in Fig. 1) for several decades [74]. 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 or 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 [66, 230]). As the cornerstone of image understanding and computer vision, object detection forms the basis for solving 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, including robot vision, consumer electronics, security, autonomous driving, human computer interaction, content based image retrieval, intelligent video surveillance, and augmented reality.

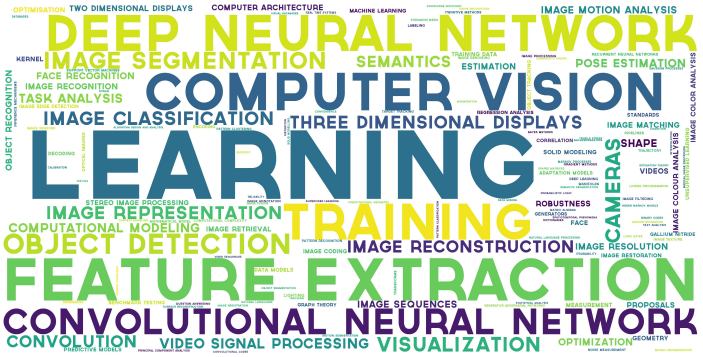



**Fig. 1** Most frequent keywords for each detection challenge. The size of each word is proportional to the frequency of that keyword. We can see that object detection has received significant attention in recent years. All keywords are extracted from ICCV and CVPR conference papers from 2016 to 2018.

Recently, deep learning techniques [102, 145] have emerged as powerful methods for learning feature representations automatically from data. In particular, these techniques have specifically provided significant improvement in object detection, as illustrated in Fig. 3.

As illustrated in Fig. 2, object detection can be grouped into one of two types [88, 302]: detection of specific instances versus the detection of broad categories. The first type aims to detect instances of a particular object (such as Donald Trump's face, the Eiffel Tower, or a neighbor's dog), essentially a matching problem. The goal of the second type is to detect (usually previously unseen) instances of some predefined object categories (for example humans, cars, bicycles, and dogs). Historically, much of the effort in the field of object detection has focused on the detection of a single category (typically faces and pedestrians) or a few specific categories. In contrast, in the past several years the research community has started moving towards the more challenging goal of

✉ 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

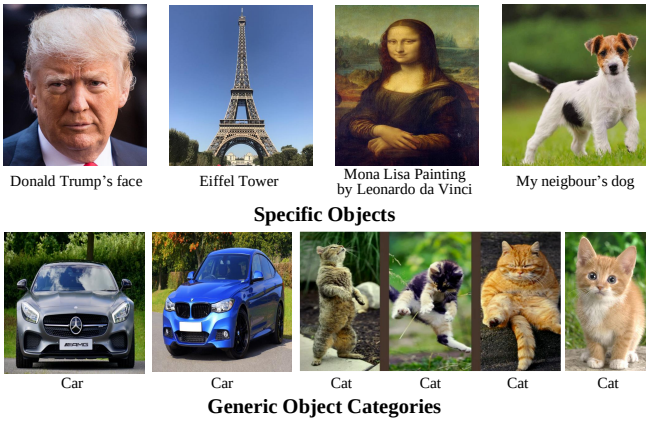



**Fig. 2** Object detection includes localizing instances of a *particular* object (top) as well as generalizing to detecting object *categories* in general (bottom). This survey focuses on recent advances for the latter problem of generic object detection.

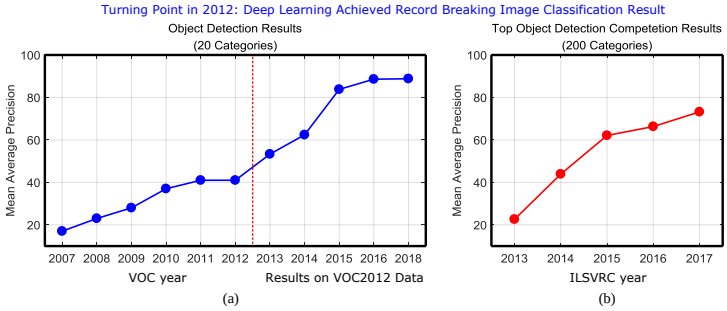



**Fig. 3** An overview of recent object detection performance. We can observe a significant improvement in performance (measured as mean average precision) since the arrival of deep learning in 2012. (a) Results on the PASCAL VOC datasets: Detection results of winning entries in the VOC2007-2012 competitions, and (b) Top object detection competition results in ILSVRC2013-2017 (results in both panels use only the provided training data).

building general purpose object detection systems whose breadth of object detection ability rivals that of humans.

In 2012, Krizhevsky *et al.* [136] proposed a Deep Convolutional Neural Network (DCNN) called AlexNet which achieved record breaking image classification accuracy in the Large Scale Visual Recognition Challenge (ILSVRC) [230]. Since that time the research focus in many computer vision domains has been on deep learning methods, indeed including the domain of generic object detection [83, 96, 82, 235, 226]. Although tremendous progress has been achieved, illustrated in Fig. 3, we are unaware of comprehensive surveys of this subject over the past five years. Given the exceptionally rapid rate of progress, this article attempts to track recent advances and summarize their achievements, in order to gain a clearer picture of the current panorama in generic object detection.

### 1.1 Comparison with Previous Reviews

Many 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 [64, 77, 57], face detection [287, 294], vehicle detection [251] and text detection [288]. There are comparatively few recent surveys focusing directly on the problem of generic object detection, except for the work by Zhang *et al.* [302] who conducted a survey on the topic of object class detection. However, the research reviewed in [88], [5] and [302] is mostly preceding 2012, and therefore before the recent striking success and dominance of deep learning and related methods.

Deep learning allows computational models to learn fantastically complex, subtle, and abstract representations, driving 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, DCNNs [144, 136, 145] have brought about breakthroughs in processing images, video, speech and audio. To be sure, there have been many published surveys on deep learning, including that of Bengio *et al.* [13], LeCun *et al.* [145], Litjens *et al.* [166], Gu *et al.* [89], and more recently in tutorials at ICCV and CVPR.

Although many deep learning based methods have been proposed for object detection, we are unaware of a comprehensive recent survey. A thorough review and summary of existing work is essential for further progress in object detection, particularly for researchers wishing to enter the field. Since our focus is on *generic* object detection, the extensive work on DCNNs for *specific* object detection, such as face detection [150, 299, 112], pedestrian detection [300, 105], vehicle detection [314] and traffic sign detection [321] will not be considered.

### 1.2 Scope

The number of papers on generic object detection based on deep learning is breathtaking. So many, in fact, that compiling any comprehensive review of the state of the art is beyond the scope of any reasonable–length paper. As a result, it is necessary to establish selection criteria, such that we have limited our focus to top journal and conference papers. Due to these limitations, we sincerely apologize to those authors whose works are not included in this paper. For surveys of work in related topics, readers are referred to the articles in Table 1. This survey focuses on major progress of the last five years, and we restrict our attention to still pictures, leaving the important subject of video object detection as a separate topic.

The main goal of this paper is to offer a comprehensive survey of deep learning based generic object detection techniques and to present some degree of taxonomy, a high level perspective and organization, primarily on the basis of popular datasets, evaluation metrics, context modeling, and detection proposal methods. The intent is that our categorization be helpful for readers to have an accessible understanding of similarities and differences between a wide variety of strategies. The proposed taxonomy gives researchers a framework to understand current research and to identify open challenges for future research.

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. A brief introduction to deep learning is given in Section 3. Popular datasets and evaluation criteria are summarized in 4. We describe the milestone object detection frameworks in Section 5. From Section 6 to Section 9, fundamental subproblems and relevant issues involved in designing object detectors are discussed. Finally, in

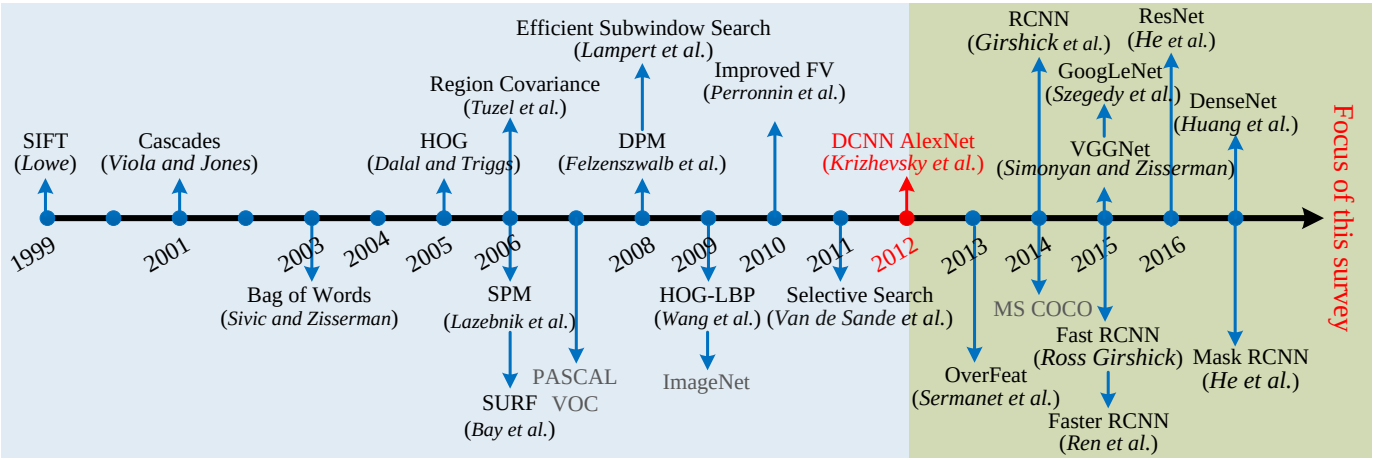
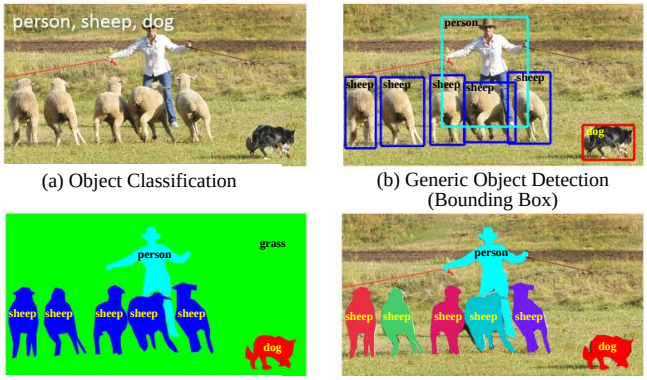



**Fig. 4** Milestones of object detection and recognition, including feature representations [45, 50, 98, 136, 143, 174, 175, 208, 244, 247, 256, 269, 272], detection frameworks [72, 83, 235, 264, 269], and datasets [66, 162, 230]. The time period up to 2012 is dominated by handcrafted features, a turning point in 2012 with the development of DCNNs for image classification by Krizhevsky *et al.* [136], with methods after 2012 dominated by deep networks. Most listed methods are highly cited and won a major ICCV or CVPR prize. See Section 2.3 for details.

**Table 1** Summary of related object detection surveys since 2000.

| No. | Survey Title | Ref. | Year | Venue | Content |
|---|---|---|---|---|---|
| 1 | Monocular Pedestrian Detection: Survey and Experiments | [64] | 2009 | PAMI | An evaluation of three pedestrian detectors |
| 2 | Survey of Pedestrian Detection for Advanced Driver Assistance Systems | [77] | 2010 | PAMI | A survey of pedestrian detection for advanced driver assistance systems |
| 3 | Pedestrian Detection: An Evaluation of the State of The Art | [57] | 2012 | PAMI | Focus on a more thorough and detailed evaluation of detectors in individual monocular images |
| 4 | Detecting Faces in Images: A Survey | [287] | 2002 | PAMI | First survey of face detection from a single image |
| 5 | A Survey on Face Detection in the Wild: Past, Present and Future | [294] | 2015 | CVIU | A survey of face detection in the wild since 2000 |
| 6 | On Road Vehicle Detection: A Review | [251] | 2006 | PAMI | A review of vision based onroad vehicle detection systems |
| 7 | Text Detection and Recognition in Imagery: A Survey | [288] | 2015 | PAMI | A survey of text detection and recognition in color imagery |
| 8 | Toward Category Level Object Recognition | [211] | 2007 | Book | Representative papers on object categorization, detection, and segmentation |
| 9 | The Evolution of Object Categorization and the Challenge of Image Abstraction | [54] | 2009 | Book | A trace of the evolution of object categorization in the last four decades |
| 10 | Context based Object Categorization: A Critical Survey | [76] | 2010 | CVIU | A review of 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 | [88] | 2011 | Tutorial | Instance and category object recognition techniques |
| 13 | Object Class Detection: A Survey | [302] | 2013 | ACM CS | Survey of generic object detection methods before 2011 |
| 14 | Feature Representation for Statistical Learning based Object Detection: A Review | [156] | 2015 | PR | Feature representation methods in statistical learning based object detection, including handcrafted and deep learning based features |
| 15 | Salient Object Detection: A Survey | [19] | 2014 | arXiv | A survey for salient object detection |
| 16 | Representation Learning: A Review and New Perspectives | [13] | 2013 | PAMI | Unsupervised feature learning and deep learning, probabilistic models, autoencoders, manifold learning, and deep networks |
| 17 | Deep Learning | [145] | 2015 | Nature | An introduction to deep learning and its typical applications |
| 18 | A Survey on Deep Learning in Medical Image Analysis | [166] | 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 | [89] | 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** | **Ours** | **2019** | **(TBD)** | **A comprehensive survey of deep learning for generic object detection** |

Section 10, we conclude the paper with an overall discussion of object detection including SoA performance and an analysis of several future research directions.




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

**Fig. 5** 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.

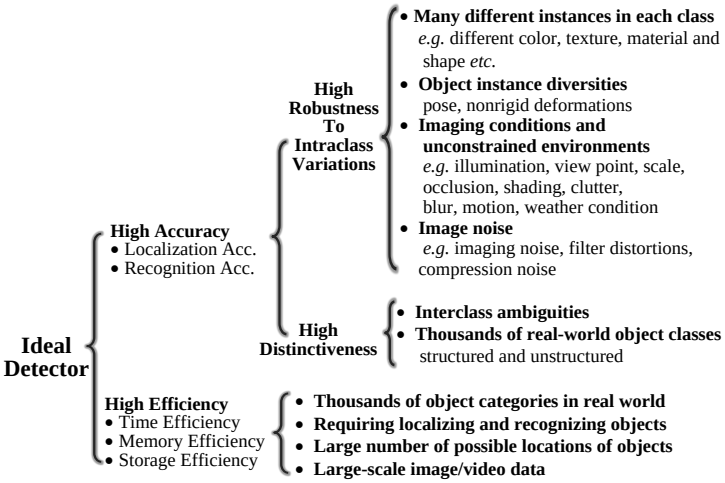



**Fig. 6** Taxonomy of challenges in generic object detection.

## 2 Generic Object Detection

### 2.1 The Problem

*Generic object detection* (*i.e.,* generic object category detection), also called *object class detection* [302] 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 predefined categories (usually *many* categories, *e.g.,* 200 categories in the ILSVRC object detection challenge) and, if present, to return the spatial location and extent of each instance. A greater emphasis is placed 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 objects (*e.g.,* humans, cows and horses) rather than unstructured scenes (such as sky, grass and cloud).

The spatial location and extent of an object can be defined coarsely using a bounding box (an axis-aligned rectangle tightly bounding the object) [66, 230], a precise pixelwise segmentation mask [302], or a closed boundary [162, 231], as illustrated in Fig. 5. To the best of our knowledge, in the current literature, for the evaluation of generic object detection algorithms, it is bounding boxes which are most widely used [66, 230], and which will be the approach we adopt in this survey. However as the research community moves towards deeper scene understanding (from image level object classification to single object localization, to generic object detection, and to pixel-wise object segmentation), it is anticipated that future challenges will be at the pixel level [162].

There are many problems closely related to that of generic object detection[1]. The goal of *object classification* or *object categorization* (Fig. 5 (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. 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 identifying/localizing all the objects present in an image, subsuming the problems of object detection and classification [66, 230, 194, 5]. Generic object detection is closely related to *semantic image segmentation* (Fig. 5 (c)), which aims to assign each pixel in an image to a semantic class label. *Object instance segmentation* (Fig. 5 (d)) aims to distinguish different instances of the same object class, while semantic segmentation does not distinguish different instances.


**Fig. 7** Changes in imaged appearance of the same class with variations in imaging conditions (a-h). There is an astonishing variation in what is meant to be a single object class (i). In contrast, the four images in (j) appear very similar, but in fact are from four different object classes. Most images are from ImageNet [230] and MS COCO [162].

[1] To the best of our 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, identification, annotation, labeling, and understanding are often differently defined [5].

## 2.2 Main Challenges

The ideal goal of generic object detection is to develop a general-purpose algorithm that the achieves two competing goals of *high quality/accuracy* and *high efficiency*, as illustrated in Fig. 6. As illustrated in Fig. 7, 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 real time with acceptable memory and storage demands.

### *2.2.1 Accuracy related challenges*

The accuracy 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. In terms of intrinsic factors, 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. 7 (*i*). Even in a more narrowly defined class, such as human or horse, object instances can appear in different poses, with nonrigid deformations or different clothes.

For imaging conditions, the variations are caused by changes in imaging, since unconstrained environments can have dramatic impacts on object appearance. In particular, a given instance can be captured subject to a wide number of differences: sunlight (dawn, day, dusk), locations, weather conditions, cameras, backgrounds, illuminations, occlusion conditions, and viewing distances. All of these conditions produce significant variations in object appearance, such as illumination, pose, scale, occlusion, clutter, shading, blur and motion, with examples illustrated in Fig. 7 (*a-h*). 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. 7 (j)). In practice, current detectors focus mainly on structured object categories, such as the 20, 200 and 91 object classes in PASCAL VOC [66], ILSVRC [230] and MS COCO [162] 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 and scalability related challenges*

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, thus computational complexity grows with the very large number of object categories, and with the very large number of possible locations and scales within a single image, as shown by the example in Fig. 7 (*c*, *d*).

A further challenge is that of scalability: A detector should be able to handle unseen objects, unknown situations, and rapidly increasing amounts of image data. For example, the scale of ILSVRC [230] is already straining the limits of the manual annotations that can feasibly be obtained. 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 [74], 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 [186, 211], with the focus later moving away from geometry and prior models towards the use of statistical classifiers (such as Neural Networks [229], SVM [197] and Adaboost [269, 283]) based on appearance features [187, 232]. 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. 4, in which two main eras (SIFT *vs.* DCNN) are highlighted. The appearance features moved from global representations [188, 253, 260] to local representations that are designed to be 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 [174], and the progress on various visual recognition tasks was based substantially on the use of local descriptors [183] such as Haar like features [269], SIFT [175], Shape Contexts [12], Histogram of Gradients (HOG) [50] and Local Binary Patterns (LBP) [192], covariance [261]. 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 [247] and Csurka *et al.* [45], Spatial Pyramid Matching (SPM) of BoW models [143], and Fisher Vectors [208].

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 DCNNs [136] achieved their record breaking results in image classification.

The use of CNN for detection and localization [229] can be traced back to 1990s. CNNs with a small number of hidden layers have been used for object detection for the last two decades [265, 229, 234]. Until recently, they were successful in restricted domains such as face detection. Recently, deeper CNN have led to record-breaking improvements in the detection of more general object categories. This shift came about when the successful application of DCNN to image classification [136] was transferred to object detection, resulting the milestone RCNN detector of Girshick *et al.* [83]. Since then lots of research in object detection builds on the rapidly evolving RCNN line of work.

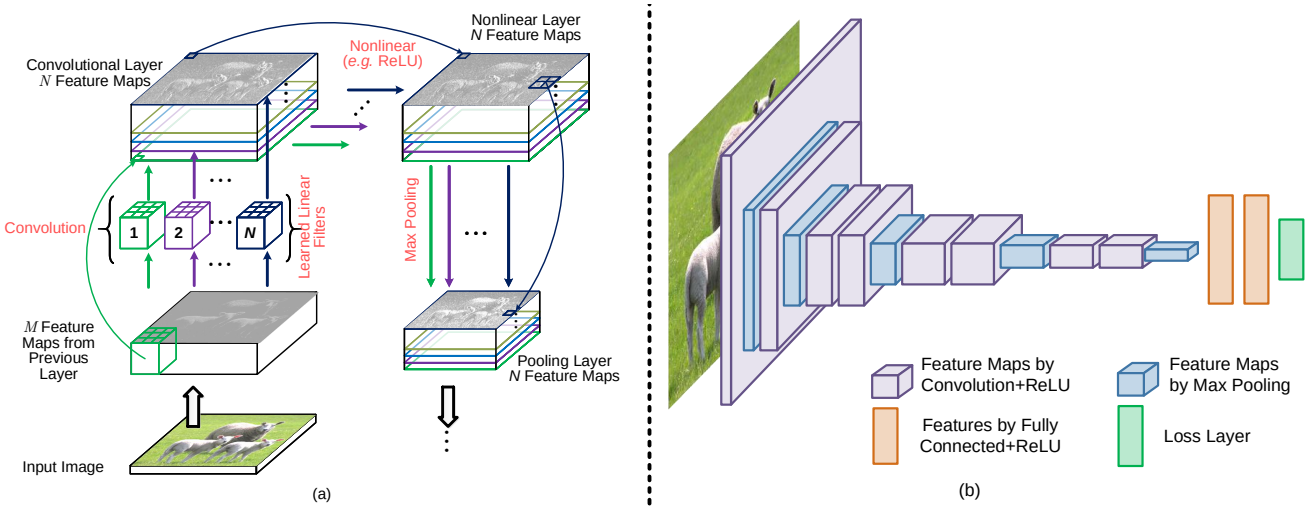



**Fig. 8** (a) Illustration of three operations that are repeatedly applied by a typical CNN: (1) Convolution with a number of linear filters; (2) Nonlinearities (*e.g.* ReLU); (3) Local pooling (*e.g.* Max Pooling). The $M$ feature maps from previous layer are convolved with $N$ different filters (*e.g.* each of size $3 \times 3 \times M$), using a stride of 1. The resulting $N$ feature maps are then passed through a nonlinear function (*e.g.* ReLU), and pooled (*e.g.* max within $2 \times 2$ regions, using stride 2) to give $N$ feature maps with reduced resolution. (b) Illustration of the architecture of VGGNet [244], a typical CNN with 11 weight layers. An image with 3 color channels is presented as the input. The network has 8 convolutional layers, 3 fully connected layers, 5 max pooling layers and a softmax classification layer. With an interleave between convolution and pooling, a feature hierarchy is constructed. The last three fully connected layers take features from the top convolutional layer as input in vector form. The final layer is a $C$ way softmax function, $C$ being the number of classes. The whole network can be optimized on an objective function (*e.g.* mean squared error or cross entropy loss) via the Stochastic Gradient Descent (SGD) method. The filter weights need to be learned by training the whole network with enough labeled training data.

The successes of deep detectors heavily rely on data and energy hungry deep networks with millions or even billions of parameters, and the availability of GPUs with very high computation capability and large scale detection datasets with fully-annotated bounding boxes play a key role in their success. While accurate annotations are labor intensive to obtain. Therefore, detectors must consider methods that can relieve annotation difficulties or can learn with small training datasets.

The successful application of DCNNs to image classification [136] transferred to object detection, resulting in the milestone Region based CNN (RCNN) detector of Girshick *et al.* [83]. 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 [52, 230] and MS COCO [162]. 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 [162, 230].

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 $10^5$ [15]. 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 A Brief Introduction to Deep Learning

Deep learning has revolutionized a wide range of machine learning tasks, from image classification and video processing to speech recognition and natural language understanding. Given this time of tremendously rapid evolution, there exist many recent survey papers on deep learning [13, 86, 89, 145, 166, 212, 280, 290, 305, 312, 317]. These surveys reviewed deep learning techniques from different perspectives [13, 86, 89, 145, 212, 280, 312], or with application to medical image analysis [166], natural language processing [290], speech recognition systems [305], and remote sensing [317].

CNNs are the most representative models of deep learning. CNNs are able to exploit the basic properties underlying natural signals: translation invariance, local connectivity, and compositional hierarchies [145]. A typical CNN has a hierarchical structure and is composed of a number of layers (such as convolution, nonlinearity, pooling *etc*) to learn representations of data with multiple levels of abstraction [145]. Pooling corresponds to downsampling/upsampling of feature maps. Convolution and nonlinearity can be expressed as follows:

$$\boldsymbol{x}_j^l = \sigma\Big(\sum_{i=1}^{N^{l-1}} \boldsymbol{x}_i^{l-1} * \boldsymbol{w}_{i,j}^l + b_j^l\Big), \tag{1}$$

where $\boldsymbol{x}_i^{l-1}$ denotes the $i$th 2D input feature map at layer $l-1$, $\boldsymbol{x}_j^l$ is the $j$th output feature map at layer $l$, $*$ denotes the convolution operation, $\boldsymbol{w}_{i,j}^l$ denotes the 2D convolutional kernel, and $b_j^l$ denotes the bias term. $N^{l-1}$ denotes the number of feature maps for layer $l-1$. The $\sigma(\cdot)$ denotes elementwise nonlinear function, which can be implemented by rectified linear unit ( ReLU) for each element as follows:

$$\sigma(x) = \max\{x, 0\}. \tag{2}$$

The three operations that are repeatedly applied by a typical CNN are illustrated in Fig. 8 (a). DCNNs having a large number of layers, a "deep" network, are referred to as Deep CNNs (DCNNs) and a typical DCNN architecture illustrated in Fig. 8 (b).

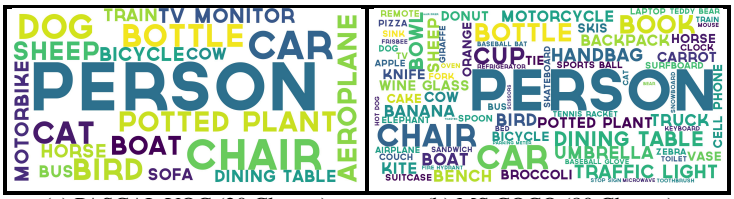
As can be seen from Fig. 8 (b), each layer of a CNN consists of a number of feature maps, within which each pixel acts like a neuron. Each neuron in a convolutional layer is connected to feature maps of the previous layer through a set of weights (essentially a filter). As can be seen in Fig. 8 (b), early layers in a CNN are typically composed of convolutional and pooling layers. The later layers are normally fully connected layers. Some sort of nonlinearity is normally present between each pair of layers.

From earlier to later layers, the input image repeatedly undergoes convolution, and with each layer the receptive field (the region of support) increases. In general, the initial CNN layers extract low-level features (*e.g.,* edges), with later layers extracting features of increasing complexity [296, 13, 145, 195].

DCNNs have a number of outstanding advantages: a hierarchical structure to learn representations of data with multiple levels of abstraction, the capacity to learn very complex functions, and learning feature representations directly and automatically from data with minimal domain knowledge. What has particularly made DCNNs feasible has been the availability of large scale labeled datasets and of GPUs with very high computational capability.

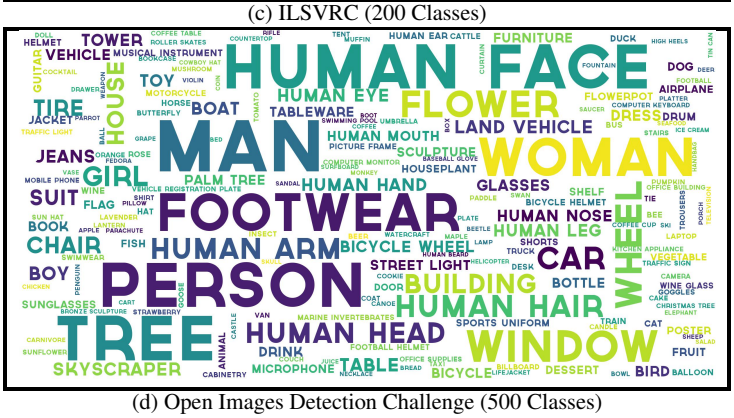
Despite the great successes, known deficiencies remain. In particular, there is an extreme need for labeled training data, there is a requirement of expensive computing resources, and considerable skill and experience are still needed to select appropriate learning parameters and network architecture. Trained networks are poorly interpretable, there is a lack of robustness to image transformations and degradations, and many DCNNs have shown serious vulnerability to attacks, all of which currently limit the use of DCNNs in many real world applications.

**Table 2** Most frequent object classes for each detection challenge. The size of each word is proportional to the frequency of that class in the training dataset.




(a) PASCAL VOC (20 Classes) (b) MS COCO (80 Classes)


(c) ILSVRC (200 Classes)




(d) Open Images Detection Challenge (500 Classes)

## 4 Datasets and Performance Evaluation

### 4.1 Datasets

Datasets have played a key role throughout the history of object recognition research, not only as a common ground for measuring and comparing the performance of competing algorithms, but also pushing the field towards increasingly complex and challenging problems. In particular, with deep learning techniques recently revolutionizing many visual recognition problems, it is large amounts of annotated data which play a key role in their success. 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, enabling unprecedented performance in object recognition.

For generic object detection, there are four famous datasets: PASCAL VOC [66, 67], ImageNet [52], MS COCO [162] and Open Images [139]. Attributes of these datasets are summarized in Table 3, and selected sample images are shown in Fig. 9. There are three steps to creating large-scale annotated datasets: determining the set of target object categories, collecting a diverse set of candidate images to represent the selected categories on the Internet, and annotating the large amount of collected images, typically by designing crowdsourcing strategies (the most challenging step). Recognizing space limitations, we refer interested readers to the original papers [66, 67, 162, 230, 139] for detailed description of these datasets in terms of construction and properties.

The four datasets form the backbone of their respective detection challenges. Each challenge consists of a publicly available dataset of images together with ground truth annotation and standardized evaluation software, and an annual competition and corresponding workshop. Statistics for the number of images and object instances in the training, validation and testing datasets[2] for the detection challenges are given in Table 4. The most frequent object classes in VOC, COCO, ILSVRC and Open Images detection datasets are visualized in Table 2.

**PASCAL VOC** [66, 67] is a multiyear effort devoted to the creation and maintenance of a series of benchmark datasets for classification and object detection, creating the precedent for standardized evaluation of recognition algorithms in the form of annual competitions. Starting from only four categories in 2005, the dataset has increased to 20 categories that are common in everyday life, as shown in Fig. 9.

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

**Table 3** Popular databases for object recognition. Some example images from PASCAL VOC, ImageNet, MS COCO and Open Images are shown in Fig. 9.

| Dataset Name | Total Images | Categories | Images Per Category | Objects Per Image | Image Size | Started Year | Highlights |
|---|---|---|---|---|---|---|---|
| PASCAL VOC (2012) [67] | 11,540 | 20 | 303 ∼ 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 intra-class 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. |
| ImageNet [230] | 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 [162] | 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 [311] | 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 [139] | 9 millions+ | 6000+ | – | – | varied | 2017 | A dataset of about 9 million images that have been annotated with image level labels, object bounding boxes and visual relationships; Support large scale object detection; Support visual relationship detection; |


**Fig. 9** Some example images with object annotations from PASCAL VOC, ILSVRC, MS COCO and Open Images. See Table 3 for summary of these datasets.

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 and, more importantly, meaning that test results can be compared from year to year. Due the availability of larger datasets like ImageNet, MS COCO and Open Images, PASCAL VOC has gradually fallen out of fashion.

**ILSVRC**, the ImageNet Large Scale Visual Recognition Challenge [230] is derived from ImageNet [52]. ILSVRC 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. A subset of ImageNet images (ImageNet1000) with 1000 different object categories and a total of 1.2 million images has been fixed to provide a standardized benchmark for the ILSVRC image classification challenge. The ImageNet1000 is also commonly used for DCNN pretraining.

**MS COCO** is a response to the criticism of ImageNet, that objects in its dataset tend to be large and well centered, making the ImageNet dataset atypical of real world scenarios. To push research to richer image understanding, researchers created the MS COCO database [162] containing complex everyday scenes with common objects in their natural context, closer to real life, where objects are labeled using fully-segmented instances to provide more accurate detector evaluation. The COCO object detection challenge [162] is probably the most challenging detection benchmark, featuring two object detection tasks: using either bounding box output or object instance segmentation output. Compared to ILSVRC it has fewer object categories, more instances per category, and it contains object segmentation annotations not available in ILSVRC. COCO introduced three new challenges:

1. It contains objects at a wide range of scales, including a high percentage of small objects [245];
2. Objects are less iconic and amid clutter or heavy occlusion;
3. The evaluation metric (see Table 5) encourages more accurate object localization.

Just like ImageNet in its time, MS COCO has become the standard for object detection today, with the dataset statistics for training, validation and testing summarized in Table 4.

**OICOD** (the Open Image Challenge Object Detection) is derived from the Open Images V4 [139], currently the largest publicly available object detection dataset, and where the challenge was organized for the first time at ECCV2018. OICOD is different from previous large scale object detection datasets like ILSVRC and MS COCO, not merely in terms of the significantly increased number of classes, images and bounding box annotations, but also regarding the annotation process. In ILSVRC and MS COCO, instances of all classes in the dataset are exhaustively annotated. For Open Images V4, a classifier was applied to each image and the resulting labels with sufficiently high scores were sent for human verification. Therefore in OICOD, for each image, only the object instances of all human confirmed positive labels are annotated,

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

whereas instances of classes not verified by humans are not annotated. OICOD has visual relationship annotations, but currently not object segmentation annotations.

### 4.2 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. To compare performance over all object categories, the *mean AP* (mAP) averaged over all object categories is adopted as the final measure of performance[3]. More details on these metrics can be found in [66, 67, 230, 104].

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 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 category label $c$ is the same as the ground truth label $c_g$.
- The overlap ratio IOU (Intersection Over Union) [66, 230]

$$\text{IOU}(b, b^g) = \frac{area(b \cap b^g)}{area(b \cup b^g)}, \tag{3}$$

between the predicted BB $b$ and the ground truth one $b^g$ is not smaller than a predefined threshold $\varepsilon$. 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 $\varepsilon$ is 0.5.

[3] 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 [230].

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

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[4] in Fig. 10. Based on the TP and FP detections, the precision $P(\beta)$ and recall $R(\beta)$ [66] can be computed as a function of the confidence threshold $\beta$, 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) [66, 230] can be found.

Since the introduction of MS COCO, more attention has been placed on the accuracy of the bounding box location. Instead of using a fixed IoU threshold, MS COCO introduces a few metrics (summarized in Table 5) for characterizing the performance of an object detector. For instance, in contrast to the traditional mAP computed at a single IoU of 0.5, $AP_{coco}$ is averaged across all object categories and multiple IoU values from 0.5 to 0.95 in steps of 0.05. Because 41% of objects in MS COCO are small (area $< 32^2$) and 24% are large (area $> 96^2$), metrics $AP_{coco}^{small}$, $AP_{coco}^{medium}$ and $AP_{coco}^{large}$ are also introduced. Finally, Table 5 summarizes the main metrics used in the PASCAL, ILSVRC and MS COCO object detection challenges, where recent modifications to the PASCAL VOC mAP metric were proposed in [139].

## 5 Detection Frameworks

There has been steady progress in object feature representations and classifiers for recognition, as evidenced by the dramatic change from handcrafted features [269, 50, 70, 95, 268] to learned DCNN features [83, 199, 82, 225, 48].

In contrast, for localization the basic "sliding window" strategy [50, 72, 70] remains mainstream, although with some efforts to avoid exhaustive search [141, 264]. 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 huge search space results in high computational cost. 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. 11 and summarized in Table 11. Nearly all detectors proposed over the last several years are based on one of these milestone detectors, attempting to improve one or more aspects. Broadly these detectors can be organized into two main categories:

[4] It is worth noting that for a given threshold $\beta$, 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.

**Table 4** 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. Starting in 2017, the train and val sets are arranged differently and the test set is divided into two roughly equally sized splits of about 20,000 images each: Test Dev and Test Challenge. Note that 2017 Test Dev/Challenge splits contain the same images as the 2015 Test Dev/Challenge splits so results across years are directly comparable.

| Challenge | Object Classes | Number of Images | | | Number of Annotated Objects | | Summary (Train+Val) | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Train | Val | Test | Train | Val | Images | Boxes | Boxes/Image |
| PASCAL VOC Object Detection Challenge | | | | | | | | | |
| VOC07 | 20 | 2,501 | 2,510 | 4,952 | 6,301(7,844) | 6,307(7,818) | 5,011 | 12,608 | 2.5 |
| VOC08 | 20 | 2,111 | 2,221 | 4,133 | 5,082(6,337) | 5,281(6,347) | 4,332 | 10,364 | 2.4 |
| VOC09 | 20 | 3,473 | 3,581 | 6,650 | 8,505(9,760) | 8,713(9,779) | 7,054 | 17,218 | 2.3 |
| VOC10 | 20 | 4,998 | 5,105 | 9,637 | 11,577(13,339) | 11,797(13,352) | 10,103 | 23,374 | 2.4 |
| VOC11 | 20 | 5,717 | 5,823 | 10,994 | 13,609(15,774) | 13,841(15,787) | 11,540 | 27,450 | 2.4 |
| VOC12 | 20 | 5,717 | 5,823 | 10,991 | 13,609(15,774) | 13,841(15,787) | 11,540 | 27,450 | 2.4 |
| ILSVRC Object Detection Challenge | | | | | | | | | |
| ILSVRC13 | 200 | 395,909 | 20,121 | 40,152 | 345,854 | 55,502 | 416,030 | 401,356 | 1.0 |
| ILSVRC14 | 200 | 456,567 | 20,121 | 40,152 | 478,807 | 55,502 | 476,668 | 534,309 | 1.1 |
| ILSVRC15 | 200 | 456,567 | 20,121 | 51,294 | 478,807 | 55,502 | 476,668 | 534,309 | 1.1 |
| ILSVRC16 | 200 | 456,567 | 20,121 | 60,000 | 478,807 | 55,502 | 476,668 | 534,309 | 1.1 |
| ILSVRC17 | 200 | 456,567 | 20,121 | 65,500 | 478,807 | 55,502 | 476,668 | 534,309 | 1.1 |
| MS COCO Object Detection Challenge | | | | | | | | | |
| MS COCO15 | 80 | 82,783 | 40,504 | 81,434 | 604,907 | 291,875 | 123,287 | 896,782 | 7.3 |
| MS COCO16 | 80 | 82,783 | 40,504 | 81,434 | 604,907 | 291,875 | 123,287 | 896,782 | 7.3 |
| MS COCO17 | 80 | 118,287 | 5,000 | 40,670 | 860,001 | 36,781 | 123,287 | 896,782 | 7.3 |
| MS COCO18 | 80 | 118,287 | 5,000 | 40,670 | 860,001 | 36,781 | 123,287 | 896,782 | 7.3 |
| Open Images Challenge Object Detection (OICOD) (Based on Open Images V4 [139]) | | | | | | | | | |
| OICOD18 | 500 | 1,643,042 | 100,000 | 99,999 | 11,498,734 | 696,410 | 1,743,042 | 12,195,144 | 7.0 |

**Table 5** Summarization of commonly used metrics for evaluating object detectors.

| Metric | Meaning | Definition and Description | |
|---|---|---|---|
| TP | True Positive | A true positive detection, per Fig. 10. | |
| FP | False Positive | A false positive detection, per Fig. 10. | |
| $\beta$ | Confidence Threshold | A confidence threshold for computing $P(\beta)$ and $R(\beta)$. | |
| $\varepsilon$ | 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 $\beta$. | |
| $R(\beta)$ | Recall | The fraction of all $N_c$ objects detected by the detector having a confidence of at least $\beta$. | |
| AP | Average Precision | Computed over the different levels of recall achieved by varying the confidence $\beta$. | |
| 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: $\{0.5 : 0.05 : 0.95\}$;<br>• $AP^{\text{IOU}=0.5}_{coco}$: mAP at IOU=0.50 (PASCAL VOC metric);<br>• $AP^{\text{IOU}=0.75}_{coco}$: mAP at IOU=0.75 (strict metric);<br>• $AP^{\text{small}}_{coco}$: mAP for small objects of area smaller than $32^2$;<br>• $AP^{\text{medium}}_{coco}$: mAP for objects of area between $32^2$ and $96^2$;<br>• $AP^{\text{large}}_{coco}$: 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 | • $AR^{\text{max}=1}_{coco}$: AR given 1 detection per image;<br>• $AR^{\text{max}=10}_{coco}$: AR given 10 detection per image;<br>• $AR^{\text{max}=100}_{coco}$: AR given 100 detection per image;<br>• $AR^{\text{small}}_{coco}$: AR for small objects of area smaller than $32^2$;<br>• $AR^{\text{medium}}_{coco}$: AR for objects of area between $32^2$ and $96^2$;<br>• $AR^{\text{large}}_{coco}$: AR for large objects of area bigger than $96^2$; |

a. Two stage detection framework, which includes a preprocessing step for generating object proposals, 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 6 to Section 9 will discuss fundamental subproblems involved in the detection framework in greater detail, including DCNN features, detection proposals, context modeling, *etc*.

### 5.1 Region Based (Two Stage Framework)

In a region based framework, category-independent region proposals[5] are generated from an image, CNN [136] 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. 11, DetectorNet [254], OverFeat [235], MultiBox [65] and RCNN [83] independently and almost simultaneously proposed using CNNs for generic object detection.

**RCNN** [83]: Inspired by the breakthrough image classification results obtained by CNN and the success of selective search in region proposal for hand-crafted features [264], Girshick *et al.* were among the first to explore CNN for generic object detection and developed RCNN [83, 85], which integrates AlexNet [136] with the region proposal method selective search [264]. As illustrated in great detail in Fig. 12, training in an RCNN framework consists of multistage pipelines:

(i) *Region proposal computation.* Class agnostic region proposals, which are candidate regions that might contain objects, are obtained via selective search [264];
(ii) *CNN model finetuning.* Region proposals, which are cropped from the image and warped into the same size, are used as the

[5] Object proposals, also called region proposals or detection proposals in this paper, are a set of candidate regions or bounding boxes in an image that may potentially contain an object. [27, 106]

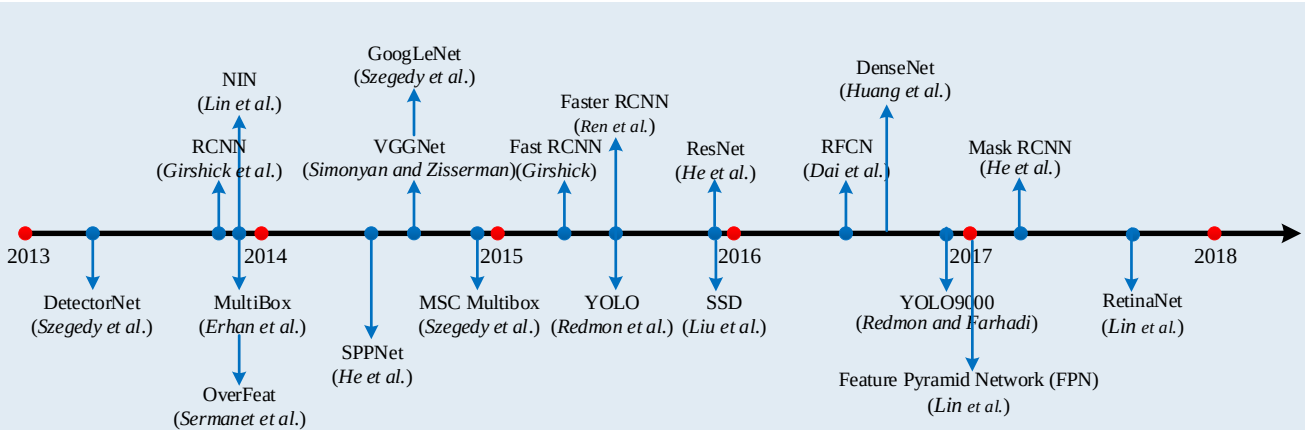



**Fig. 11** Milestones in generic object detection based on the point in time of the first arXiv version.

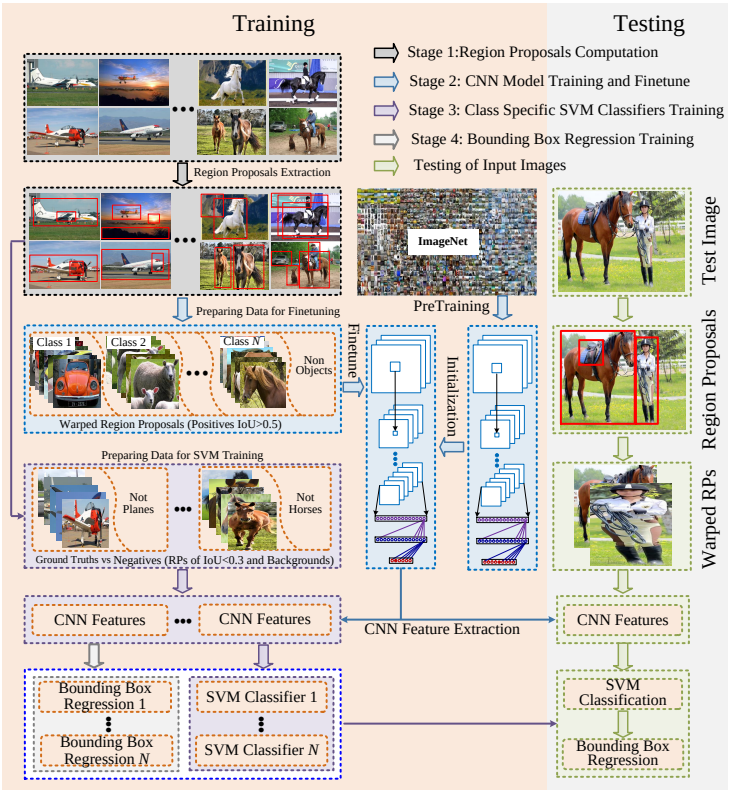



**Fig. 12** Illustration of the milestone detecting framework RCNN [83, 85] in great detail.

input for finetuning a CNN model pretrained using large scale dataset such as ImageNet. At this stage, all region proposals with $\geqslant 0.5$ IOU [6] overlap with a ground truth box are defined as positives for that ground truth box's class and the rest as negatives.

(iii) *Class specific SVM classifiers training.* A set of class specific linear SVM classifiers are trained using fixed length features extracted with CNN, replacing the softmax classifier learned by finetuning. For training SVM classifiers, positive examples are defined simply to be the ground truth boxes for each class. A region proposal with less than 0.3 IOU overlap with all ground truth instances of a class as a negative for that class. Note that the positive and negative examples defined for training the SVM classifiers are different from those for finetuning the CNN.

(iv) *Class specific bounding box regressors training.* Bounding box regression is learned for each object class with CNN features.

In spite of achieving high object detection quality, RCNN has notable drawbacks [82]:

1. Training is a multistage pipeline, which is inelegant, slow and hard to optimize because each individual stage must be trained separately.
2. For SVM classifier and bounding box regressor training, it is expensive in both disk space and time, because CNN features need to be extracted from each object proposal in each image and saved to disk, posing great challenges for large scale detection. With very deep networks, such as VGG16 [244], this process takes long computational time for a large scale dataset. These features require huge amount of storage.
3. Testing is slow, since CNN features are extracted per object proposal in each testing image, without sharing computation.

All of these drawbacks motivate successive innovations in this area, leading to a number of improved detection frameworks such as SPPNet, Fast RCNN, Faster RCNN, *etc*.

**SPPNet** [96]: 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.* [96] introduced the traditional spatial pyramid pooling (SPP) [87, 143] 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 [96] is unable to update the convolutional layers before the SPP layer, which limits the accuracy of very deep networks.

[6] Please refer to Section 4.2 for definition of IOU.

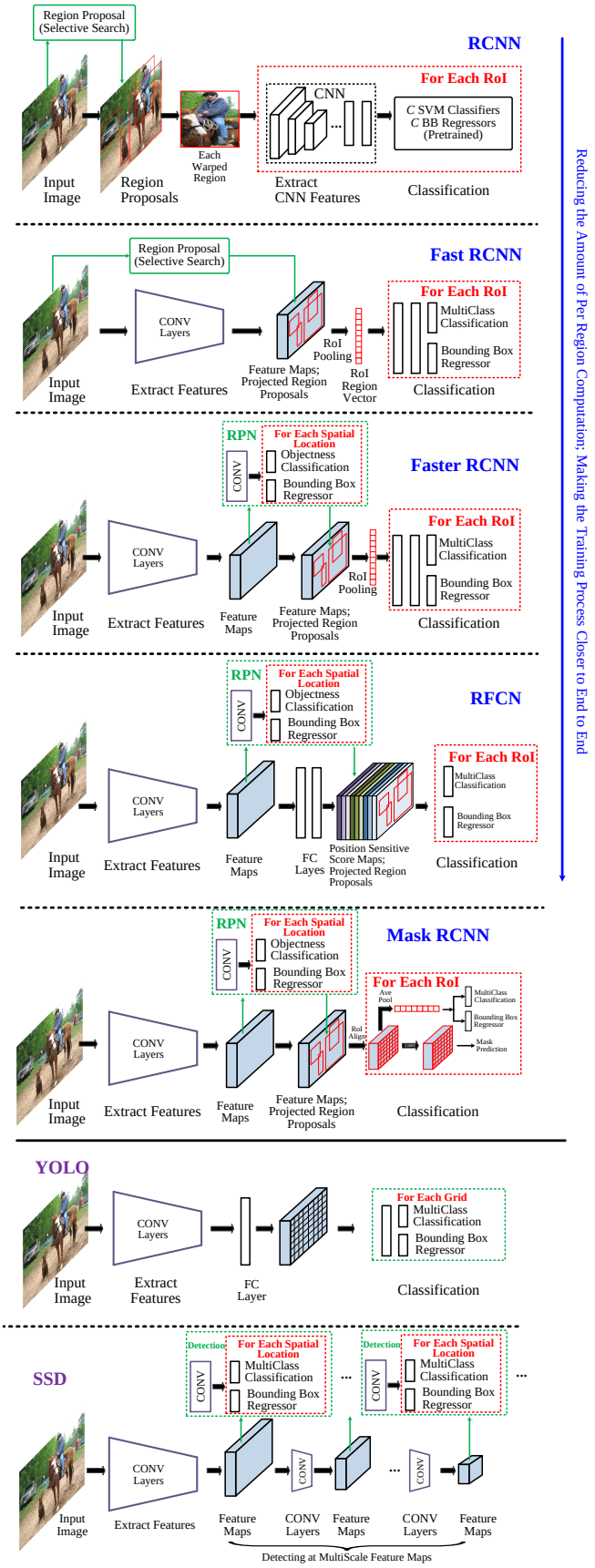



**Fig. 13** High level diagrams of the leading frameworks for generic object detection. The properties of these methods are summarized in Table 11.

**Fast RCNN** [82]: Girshick [82] 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. 13, Fast RCNN enables end-to-end detector training (given fix proposals) 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 Bounding Box Regressors (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** [225, 226]: Although Fast RCNN significantly sped up the detection process, it still relies on external region proposals. Region proposal computation is exposed as the new speed bottleneck in Fast RCNN. Recent work has shown that CNNs have a remarkable ability to localize objects in CONV layers [309, 310, 44, 196, 94], an ability which is weakened in the FC layers. Therefore, the selective search can be replaced by a CNN in producing region proposals. The Faster RCNN framework proposed by Ren *et al.* [225, 226] proposed an efficient and accurate Region Proposal Network (RPN) to generating region proposals. They utilize the same backbone network to accomplish the task of RPN for region proposal and Fast RCNN for region classification. In Faster RCNN, the RPN and the detection network (*i.e.* fast RCNN) share all convolutional layers of the backbone network. The features from the last shared convolutional layer are used for region proposal and region classification from separate branches. As shown in Fig. 13, the detection network is Fast RCNN and RPN builds on top of the last CONV layer. RPN first initializes $k$ reference boxes (*i.e.* the so called *anchors*) of different scales and aspect ratios at each CONV feature map location. Anchor positions are image content independent, but the feature vectors extracted from anchors are image content dependent. Each 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 the detection network Fast RCNN where bounding box regression is performed on features pooled from arbitrarily sized regions and the regression weights are shared by all region sizes, the features used for regression in RPN are of the same-shape anchor box on the feature maps. $k$ anchors lead to $k$ regressors. 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) [173, 237]; Faster RCNN is thus a purely CNN based framework without using handcrafted features. For the VGG16 model [244], Faster RCNN can test at 5 FPS (including all stages) 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 [225] contains several alternating training stages. This was then simplified by one-stage joint training in [226].

Concurrent with the development of Faster RCNN, Lenc and Vedaldi [147] 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 detection in the CONV rather than FC layers. They showed 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.* [48] 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. 13, 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.* [48] 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.* [48] 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.* [48] 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 [82, 225] is added. They showed that the RFCN with ResNet101 [98] 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.* [99] proposed Mask RCNN to tackle pixelwise 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) [173, 237] 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 [284, 163], 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 [99].

**Chained Cascade Network and Cascade RCNN**: The essence of cascade is to learn more discriminative classifiers by using multistage classifiers. Classifiers at early stages discard large number of easy negative samples so that classifiers at latter stage can focus on handling more difficult examples. Cascade has been widely used in object detection [71, 20, 155]. Two-stage object detection can be considered as cascade of two object detectors, the first object detector removes large amount of background regions and the second stage classifying the remaining regions. Recently, end to end learning of more than two cascaded classifiers and DCNNs for generic object detection was first proposed in the Chained Cascade Network [201] and then extended in the Cascade RCNN [23]. Recently, cascade is further applied for simultaneous object detection and instance segmentation [31], led to the winning of COCO 2018 Detection Challenge.

**Light Head RCNN**: In order to further speed up the detection speed of RFCN [48], Li *et al.* [161] 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.* [161] applied a large kernel separable convolution to produce thin feature maps with small channel number (*e.g.* 490 channels for the COCO detection benchmark) and a cheap RCNN subnetwork, leading to an excellent tradeoff of speed and accuracy.

### 5.2 Unified Pipeline (One Stage Pipeline)

The region based pipeline strategies of Section 5.1 have prevailed on detection benchmarks since RCNN [83]. The works introduced in Section 5.1 have led to faster and more accurate detectors, and the current leading results on popular benchmark datasets are all based on Faster RCNN [225]. In spite of that progress, region based approaches are computationally expensive for current 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 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.* [254] 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 [136] 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, thus making it slow.

**OverFeat**, proposed by Sermanet *et al.* [235], can be considered as one of the first modern one stage object detectors based on fully convolutional deep networks. It is one of the most influential object detection frameworks, winning the ILSVRC2013 localization and detection competition. The framework of OverFeat is

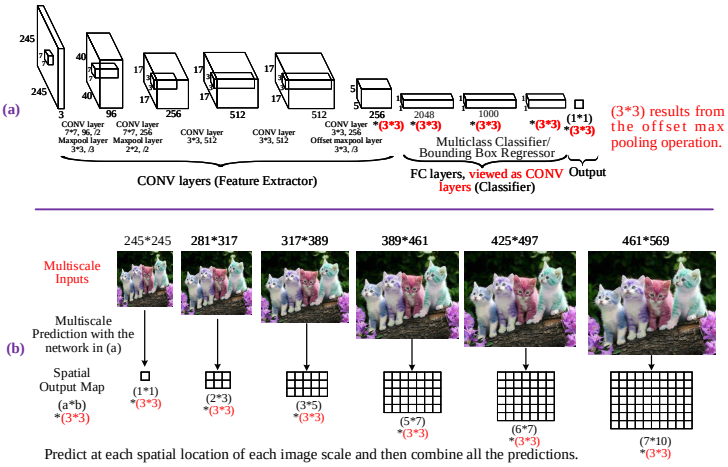



**Fig. 14** Illustration of the influential detecting framework OverFeat [235].

illustrated in Fig. 14. As can be seen from Fig. 14, OverFeat performs object detection in a multiscale sliding window fashion via a single forward pass through the fully convolutional layers in the network (*i.e.* the "Feature Extractor" shown in Fig. 14 (a)). The key steps of object detection at test time can be summarized as follows.

(i) *Generate object candidates by performing object classification via a sliding window fashion on multiscale images.* OverFeat uses a CNN like AlexNet [136] (as shown in Fig. 14 (a)) which seemingly require input images of fixed sizes due to the fully connected layers of them have fixed dimensions. In order to take advantage of the inherent computational efficiency of convolution and make the sliding window approach computationally efficient, OverFeat casts the network (as shown in Fig. 14 (a)) into a fully convolutional network that take input of any size. In specific, OverFeat utilizes the idea of extending a CNN to images of arbitrary sizes by viewing fully connected layers as convolutions with kernels of $1 \times 1$ spatial extent. OverFeat leverages multiscale features to improve the overall performance by passing up to six enlarged scales of the original image through the network (as shown in Fig. 14 (b)), resulting in a significantly increased number of evaluated context views. For each of the multiscale inputs, the classifier (as shown in Fig. 14 (a)) outputs a grid of predictions (a class and a confidence for each location), indicating the the presence of an object, as shown in Fig. 14 (b).

(ii) *Increase the number of predictions by offset max pooling*. In order to increase the resolution of the final prediction, OverFeat applies offset max pooling after the last CONV layer, *i.e.* performing subsampling operation at every offset. This approach yields much more views for voting, which increases robustness while remaining efficient.

(iii) *Bounding box regression.* Once an object is identified, a single bounding box regressor is applied at this location. The classifier and the regressor share the same feature extraction layers (*i.e.* CONV layers), only the FC layers need to be recomputed after computing the classification network. They naturally share computation between overlapping regions.

(iv) *Combine predictions.* OverFeat uses a greedy merge strategy to combine the individual bounding box predictions across all locations and scales.

OverFeat has a significant speed advantage over RCNN [83], which was proposed during the same period, but is significantly less accurate because it was hard to train fully convolutional network at that time. The speed advantage derives from sharing the computation of convolution between overlapping windows using fully convolutional network. OverFeat is similar to latter frameworks such as YOLO [223] and SSD [171]. One main difference is that the classifier and the regressors in OverFeat are trained sequentially.

**YOLO (You Only Look Once)**: Redmon *et al.* [223] 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. 13. Since the region proposal generation stage is completely dropped, YOLO directly predicts detections using a small set of candidate regions [7]. 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 [223], 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 than Fast RCNN, resulting from the coarse division of bounding box location, scale and aspect ratio. As discussed in [223], YOLO may fail to localize some objects, especially small ones. This is 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 per image, such as the ILSVRC detection challenge.

**YOLOv2 and YOLO9000**: Redmon and Farhadi [222] proposed YOLOv2, an improved version of YOLO, in which the custom GoogLeNet [256] network is replaced with a simpler DarkNet19, plus utilizing a number of strategies drawn from existing work, such as batch normalization [97], removing the fully connected layers, using good anchor boxes learned via *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 [222] introduced YOLO9000, which can detect over 9000 object categories in real time by proposing a joint optimization method to train simultaneously on ImageNet classification dataset and COCO detection dataset with WordTree to combine data from multiple sources. Such joint training allows YOLO9000 to perform weakly supervised detection, *i.e.* detecting object classes that don't have bounding box annotations. For instance, YOLO9000 gets 19.7% mAP on the ImageNet detection validation set despite only having detection data for 44 of the 200 classes.

**SSD (Single Shot Detector)**: In order to preserve real-time speed without sacrificing too much detection accuracy, Liu *et al.* [171] proposed SSD, which is faster than YOLO [223] and has ac-

[7] YOLO uses far fewer bounding boxes, only 98 per image compared to about 2000 from Selective Search.

curacy competitive with state of the art region-based detectors, including Faster RCNN [225]. SSD effectively combines ideas from RPN in Faster RCNN [225], YOLO [223] and multiscale CONV features [94] 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 [244] (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 $300 \times 300$ input, SSD achieves $74.3\%$ mAP on the VOC2007 test at 59 FPS on a Nvidia Titan X (vs Faster RCNN 7 FPS with mAP $73.2\%$ or YOLO 45 FPS with mAP $63.4\%$).

**CornerNet:** More recently, Law *et al.* [142] questioned the dominant role that anchor boxes[8] have come to play in SoA object detection frameworks [82, 99, 223, 171]. Law *et al.* [142] argue that the use of anchor boxes, especially in one stage detectors [75, 164, 171, 223], has drawbacks [142, 164] such as causing a huge imbalance between positive and negative examples, slowing down training and introducing extra hyperparameters. Therefore, to avoid the use of anchor boxes and to borrow ideas from the work on Associative Embedding in multiperson pose estimation [191], Law *et al.* [142] proposed CornerNet by formulating bounding box object detection as detecting paired keypoints[9]: the top left corner and the bottom right corner. In CornerNet, the backbone network is by stacking two Hourglass networks [190]. In addition, a simple corner pooling approach is proposed to better localize corners. Law *et al.* [142] showed that CornerNet achieved a $42.1\%$ AP on MS COCO, outperforming all existing one stage detectors. However, the average inference time is about 4FPS on a Titan X GPU, significantly slower than one stage detectors SSD [171] and YOLO [223]. CornerNet [142] often generates some incorrect bounding boxes because it is challenging to decide which pairs of keypoints should be grouped into the same objects. To further improve on CornerNet [142], Duan *et al.* [60] proposed CenterNet to detect each object as a triplet of keypoints, by introducing one extra keypoint, *i.e.* the central region of a proposal. On MS COCO detection task, CenterNet achieved an AP of $47.0\%$, which outperformed most existing one stage detectors. The inference speed of CenterNet is slower than CornerNet.

## 6 Object Representation

As one of the main components in any detector, good feature representations are of primary importance in object detection [54, 83, 80, 316]. In the past, a great deal of effort was devoted to designing local descriptors (*e.g.,* SIFT [174] and HOG [50]) and to explore approaches (*e.g.,* Bag of Words [247] and Fisher Vector [208]) to group and abstract the descriptors into higher level representations in order to allow the discriminative object parts to emerge, however these feature representation methods required careful engineering and considerable domain expertise.

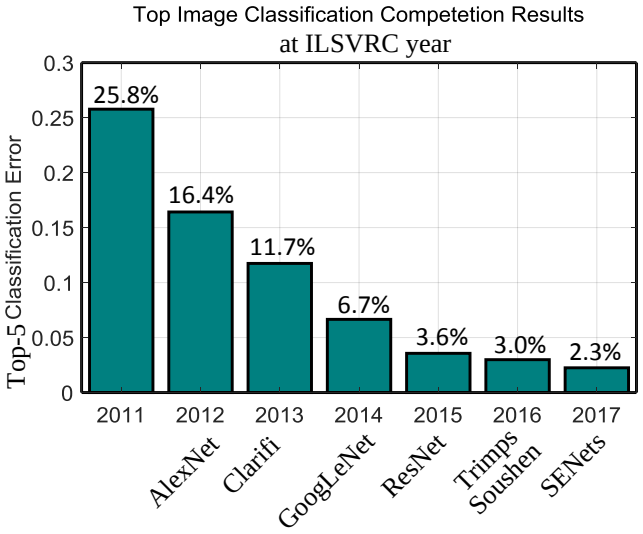



**Fig. 15** Performance of winning entries in the ILSVRC competitions from 2011 to 2017 in the image classification task.

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 [13, 145]. As the learning procedure reduces the dependency of specific domain knowledge and complex procedures needed in traditional feature engineering [13, 145], the burden for feature representation has been transferred to the design of better network architectures and training procedures.

The leading frameworks reviewed in Section 5 (RCNN [83], Fast RCNN [82], Faster RCNN [225], YOLO [223], SSD [171]) have persistently promoted detection accuracy and speed. It is generally accepted that the CNN representation plays a crucial role and the CNN architecture (Section 6.1 and Table 15) 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.

### 6.1 Popular CNN Architectures

CNN architectures (introduced in Section 3) serve as network backbones to be used in the detection frameworks described in Section 5. Representative frameworks include AlexNet [137], ZFNet [296] VGGNet [244], GoogLeNet [256], Inception series [121, 257, 258], ResNet [98], DenseNet [114] and SENet [111], which are summarized in Table 6, and where the network improvement

[8] Boxes of various sizes and aspect ratios that serve as object candidates.

[9] The idea of using keypoints for object detection appeared previously in DeNet [262].

**Table 6** DCNN architectures that were commonly used for generic object detection. Regarding the statistics for "#Paras" and "#Layers", the final FC prediction layer is not taken into consideration. "Test Error" column indicates the Top 5 classification test error on ImageNet1000. When ambiguous, the "#Paras", "#Layers", and "Test Error" refer to: OverFeat (accurate model), VGGNet16, ResNet101 DenseNet201 (Growth Rate 32, DenseNet-BC), ResNeXt50 (32*4d), and SE ResNet50.

| No. | DCNN Architecture | #Paras ($\times 10^6$) | #Layers (CONV+FC) | Test Error (Top 5) | First Used In | Highlights |
|---|---|---|---|---|---|---|
| 1 | AlexNet [137] | 57 | $5+2$ | 15.3% | [83] | The first DCNN found effective for ImageNet classification; the historical turning point from hand-crafted features to CNN; Winning the ILSVRC2012 Image classification competition. |
| 2 | ZFNet (fast) [296] | 58 | $5+2$ | 14.8% | [96] | Highly similar to AlexNet, different in stride for convolution, filter size, and number of filters for some layers. |
| 3 | OverFeat [235] | 140 | $6+2$ | 13.6% | [235] | Similar to AlexNet, different in stride for convolution, filter size, and number of filters for some layers. |
| 4 | VGGNet [244] | 134 | $13+2$ | 6.8% | [82] | Increasing network depth significantly by stacking $3 \times 3$ convolution filters and increasing the network depth step by step. |
| 5 | GoogLeNet [256] | 6 | 22 | 6.7% | [256] | Use Inception module, which uses multiple branches of convolutional layers with different filter sizes and then concatenates feature maps produced by these branches. The first inclusion of bottleneck structure and global average pooling. These designs allow the network to go wider and deeper with fewer parameters and acceptable computation. |
| 6 | Inception v2 [121] | 12 | 31 | 4.8% | [108] | Faster training with the introduce of Batch Normalization. |
| 7 | Inception v3 [257] | 22 | 47 | 3.6% | | Inclusion of separable convolution and spatial resolution reduction. |
| 8 | YOLONet [223] | 64 | $24+1$ | – | [223] | A network inspired by GoogLeNet used in YOLO detector. |
| 9 | ResNet50 [98] | 23.4 | 49 | 3.6% (ResNets) | [98] | With identity mapping added, substantially deeper network can be effectively learned. Requires fewer parameters than VGG by using the global average pooling and bottleneck introduced in GoogLeNet. |
| 10 | ResNet101 [98] | 42 | 100 | | [98] | |
| 11 | InceptionResNet v1 [258] | 21 | 87 | 3.1% (Ensemble) | | Combination of identity mapping and Inception module, with similar computational cost of Inception v3, but faster training process. |
| 12 | InceptionResNet v2 [258] | 30 | 95 | | [116] | A costlier residual version of Inception, with significantly improved recognition performance. |
| 13 | Inception v4 [258] | 41 | 75 | | | An Inception variant without residual connections with roughly the same recognition performance as InceptionResNet v2, but significantly slower. |
| 14 | ResNeXt [284] | 23 | 49 | 3.0% | [284] | Repeating a building block that aggregates a set of transformations with the same topology. |
| 15 | DenseNet201 [114] | 18 | 200 | – | [313] | Concatenate each layer with every other layer in a feed forward fashion within a dense block. Alleviate the vanishing gradient problem, encourage feature reuse, and substantially reduce the number of parameters. |
| 16 | DarkNet [222] | 20 | 19 | – | [222] | Similar to VGGNet, but with significantly less parameters by using fewer filters at each layer. |
| 17 | MobileNet [108] | 3.2 | $27+1$ | – | [108] | Light weight deep CNNs using depth-wise separable convolutions for mobile applications. |
| 18 | SE ResNet [111] | 26 | 50 | 2.3% (SENets) | [111] | Channel-wise attention by a novel block called *Squeeze and Excitation*. Complementary to existing backbone CNNs. |

in object recognition can be seen from Fig. 15. A further review of recent CNN advances can be found in [89].

As can be observed from Table 6, the trend in architecture evolution is that networks are getting deeper: AlexNet has of 8 layers, VGGNet 16 layers, and more recently ResNet and DenseNet both surpassed the 100 layer mark, and it was VGGNet [244] and GoogLeNet [256], in particular, which showed that increasing depth can improve the representational power of deep networks. Interestingly, as can be observed from Table 6, 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. 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 [256] in carefully designed topologies, the parameters of GoogLeNet is dramatically reduced, compared to networks like AlexNet, ZFNet and VGGNet. 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 [98], InceptionResNets [258] combine the Inception networks with shortcut connections, claiming that shortcut connections can significantly accelerate the training of Inception networks. Extending ResNets, Huang *et al.* [114] proposed DenseNets which are built from dense blocks, which connect each layer to every other layer in a feed forward fashion, leading to compelling advantages such as parameter efficiency, implicit deep supervision [10], and feature reuse. Recently, Hu *et al.* [98] 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. The SE block adaptively recalibrates channelwise feature responses by explicitly modeling the interdependencies between convolutional feature channels. The SE

[10] DenseNets perform deep supervision in an implicit way, *i.e.* individual layers receive additional supervision from other layers through the shorter connections. The benefits of deep supervision have previously been demonstrated in Deeply Supervised Nets (DSN) [146].

block can be integrated with state of the art deep networks, leading to winning of the ILSVRC 2017 classification task. Research on CNN architectures remains active, and a number of backbone networks are still emerging such as Hourglass [142], Dilated Residual Networks [292], Xception [43], DetNet [160], Dual Path Networks (DPN) [37], FishNet [250], GLoRe [38], etc.

The training of a CNN requires a large scale labeled dataset with intraclass diversity. Unlike image classification, detection requires localizing (possibly many) objects from an image. It has been shown [202] 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 [219, 8, 58, 289], which can support a wider range of visual recognition tasks. For detection, the pretrained network is typically finetuned[11] on a given detection dataset [58, 83, 85]. Several large scale image classification datasets are used for CNN pretraining. Among them the ImageNet1000 dataset [52, 230] with 1.2 million images of 1000 object categories, or the Places dataset [311] which is much larger than ImageNet1000 but has fewer classes, a recent hybrid dataset [311] combining the Places and ImageNet datasets, or the JFT300M dataset labelled with 18291 hierarchical categories [103, 249].

Pretrained CNNs without finetuning were explored for object classification and detection in [58, 85, 1], where it was shown that detection accuracies are different for features extracted from different layers; for example, for AlexNet pretrained on ImageNet, FC6 / FC7 / Pool5 are in descending order of detection accuracy [58, 85]; finetuning a pretrained network can increase detection performance significantly [83, 85], 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 and target datasets plays a critical role, for example that ImageNet based CNN features show better performance for object detection tasks than for human action related tasks [309, 8].

## 6.2 Methods For Improving Object Representation

Deep CNN based detectors such as RCNN [83], Fast RCNN [82], Faster RCNN [225] and YOLO [223], typically use the deep CNN architectures listed in Table 6 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) [72, 83, 96], 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.

[11] 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.

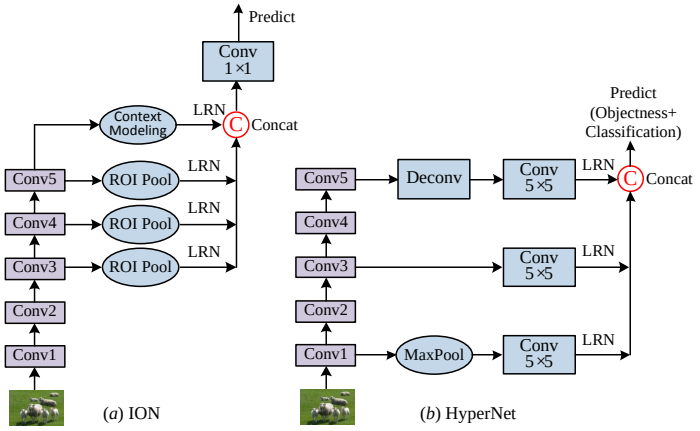



**Fig. 16** Comparison of HyperNet and ION. LRN is Local Response Normalization, which performs a kind of "lateral inhibition" by normalizing over local input regions [123].

### 6.2.1 Handling of Object Scale Variations

This inherent feature hierarchy produces feature maps of different spatial resolutions, but have structural problems [94, 173, 243]: 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 convolution [291] (also named atrous convolution) [48, 33] have been proposed. Dilated convolution increases resolution of features but inevitably increases the computational complexity. 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 [243, 306, 163, 132] have been proposed to improve detection accuracy by exploiting multiple CNN layers, broadly falling into three types of **multiscale object detection**:

(i) Detecting with combined features of multiple CNN layers [94, 131, 11];
(ii) Detecting at multiple CNN layers [171, 24, 169, 238];
(iii) Combinations of the above two methods [75, 163, 243, 132, 313, 301].

**(1) Detecting with combined features of multiple CNN layers.** Many approaches, including Hypercolumns [94], HyperNet [131], and ION [11], combine features from multiple layers before making a prediction. Such feature combination is commonly accomplished via concatenation, a classic neural network idea that concatenates features from different layers, architectures which have recently become popular for semantic segmentation [173, 237, 94]. As shown in Fig. 16 (a), ION [11] uses RoI pooling to extract

**Table 7** Summarization of properties of representative methods in improving DCNN feature representations for generic object detection. Details for Groups (1), (2), and (3) are provided in Section 6.2. Group (1) corresponds to detection with combined features of multiple CNN layers. Group (2) corresponds to detection from multiple CNN layers. Group (3) is a combination of groups (1) and (2). Group (4) handle geometric transformations. Abbreviations: Selective Search (SS), EdgeBoxes (EB), InceptionResNet (IRN). *Conv-Deconv*denotes the use of upsampling and convolutional layers with lateral connections to supplement the standard backbone network. 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 [295] which reported with COCO2015 Test-Standard. Note that 2017 Test Dev/Challenge splits contain the same images as the 2015 Test Dev/Challenge splits so results across years are directly comparable.

| Group | Detector Name | Region Proposal | Backbone DCNN | Pipelined Used | mAP@IoU=0.5 VOC07 | mAP@IoU=0.5 VOC12 | mAP COCO | mAP@IoU=0.5 COCO | Published In | Highlights |
|---|---|---|---|---|---|---|---|---|---|---|
| (1) Single detection with multilayer features | ION [11] | SS+EB MCG+RPN | VGG16 | Fast RCNN | 79.4 (07+12) | 76.4 (07+12) | 55.7 | 33.1 | CVPR16 | Use features from multiple layers; use spatial recurrent neural networks for modeling contextual information; the Best Student Entry and the 3rd overall in the COCO detection challenge 2015. |
| | HyperNet [131] | RPN | VGG16 | Faster RCNN | 76.3 (07+12) | 71.4 (07++12) | – | – | CVPR16 | Use features from multiple layers for both region proposal and region classification. |
| | PVANet [128] | 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 [236], Inception [256], and HyperNet [131]. |
| (2) Detection at multiple layers | SDP+CRC [286] | EB | VGG16 | Fast RCNN | 69.4 (07) | – | – | – | CVPR16 | Use features in multiple layers to reject easy negatives via CRC and then classify remaining proposals using SDP, which extracts convolutional features of an object proposal from a layer corresponding to its scale. |
| | MSCNN [24] | RPN | VGG | Faster RCNN | Only Tested on KITTI | | | | ECCV16 | Region proposal and classification are performed at multiple layers; includes feature upsampling; end to end learning. |
| | MPN [295] | SharpMask [210] | VGG16 | Fast RCNN | – | – | 51.9 | 33.2 | BMVC16 | Concatenate features from different convolutional layers and features of different contextual regions; a new loss that encourages a classifier to perform well at multiple overlap thresholds; ranked 2nd in both the COCO15 detection and segmentation challenges; use segmentation annotations for training. |
| | DSOD [238] | Free | DenseNet | SSD | 77.7 (07+12) | 72.2 (07++12) | 47.3 | 29.3 | ICCV17 | Concatenate feature sequentially, like DenseNet. Train from scratch on the target dataset without pretraining. |
| | RFBNet [169] | Free | VGG16 | SSD | 82.2 (07+12) | 81.2 (07++12) | 55.7 | 34.4 | ECCV18 | Propose a multi-branch convolutional block similar to the Inception block [256], but using dilated convolution. |
| (3) Combination of (1) and (2) | DSSD [75] | Free | ResNet101 | SSD | 81.5 (07+12) | 80.0 (07++12) | 53.3 | 33.2 | 2017 | Use Conv-Deconv, as shown in Fig. 17 (c1, c2). |
| | FPN [163] | RPN | ResNet101 | Faster RCNN | – | – | 59.1 | 36.2 | CVPR17 | Use Conv-Deconv, as shown in Fig. 17 (a1, a2); Widely used in detectors. |
| | TDM [243] | RPN | ResNet101 VGG16 | Faster RCNN | – | – | 57.7 | 36.8 | CVPR17 | Use Conv-Deconv, as shown in Fig. 17 (b2). |
| | RON [132] | RPN | VGG16 | Faster RCNN | 81.3 (07+12+CO) | 80.7 (07++12+CO) | 49.5 | 27.4 | CVPR17 | Use Conv-deconv, as shown in Fig. 17 (d2); Add the objectness prior to significantly reduce the searching space of objects. |
| | ZIP [152] | RPN | Inceptionv2 | Faster RCNN | 79.8 (07+12) | – | – | – | IJCV18 | Use Conv-Deconv, as shown in Fig. 17 (f1). Propose a map attention decision (MAD) unit to assign the weight for features from different layers. |
| | STDN [313] | Free | DenseNet169 | SSD | 80.9 (07+12) | – | 51.0 | 31.8 | CVPR18 | A new scale transfer module, which resizes features of different scales to the same scale in parallel. |
| | RefineDet [301] | RPN | VGG16 ResNet101 | Faster RCNN | 83.8 (07+12) | 83.5 (07++12) | 62.9 | 41.8 | CVPR18 | Use cascade to obtain better and less anchors. Use Conv-deconv as shown in Fig. 17 (e2) to improve features. |
| | PANet [170] | RPN | ResNeXt101 +FPN | Mask RCNN | – | – | **67.2** | **47.4** | CVPR18 | Shown in Fig. 17 (g); Based on FPN, PANet adds another bottom-up path which shortens the information path between lower layers and topmost feature; Presents adaptive feature pooling; Ranks the $1^{st}$ in the COCO 2017 Challenge Instance Segmentation task and the $2^{nd}$ in Object Detection task. |
| | DetNet [160] | RPN | DetNet59+FPN | Faster RCNN | – | – | 61.7 | 40.2 | ECCV18 | Introduces dilated convolution into the ResNet backbone to maintain high resolution in deeper layers; Shown in Fig. 17 (i). |
| | FPR [133] | – | VGG16 ResNet101 | SSD | 82.4 (07+12) | 81.1 (07++12) | 54.3 | 34.6 | ECCV18 | Fuse task oriented features across different spatial locations and scales, globally and locally; Shown in Fig. 17 (h). |
| | M2Det [307] | – | SSD | VGG16 ResNet101 | – | – | 64.6 | 44.2 | AAAI19 | As shown in Fig. 17 (j), the multilayer features extracted by the backbone are fused as the base feature; The base feature was fed into a newly designed top down path to learn a set of multilevel features; The set of multilevel features are recombined to construct a feature pyramid for object detection. |
| (4) Model Geometric Transforms | DeepIDNet [199] | SS+ EB | AlexNet ZFNet OverFeat GoogLeNet | RCNN | 69.0 (07) | – | – | 25.6 | CVPR15 | Introduce a deformation constrained pooling layer to learn deformation constraint for any CNN layer. Deformation constrained pooling layer can be jointly learned with convolutional layers in existing DCNNs. Utilize the following modules that are not trained end to end: cascade, context modeling, model averaging, and bounding box location refinement in the multistage detection pipeline. |
| | DCN [49] | RPN | ResNet101 IRN | RFCN | 82.6 (07+12) | – | 58.0 | 37.5 | CVPR17 | Design deformable convolution and deformable RoI pooling modules that can replace plain convolution in existing DCNNs. |
| | DPFCN [184] | AttractioNet [81] | ResNet | RFCN | 83.3 (07+12) | 81.2 (07++12) | 59.1 | 39.1 | IJCV18 | Design a deformable part based RoI pooling layer to explicitly select discriminative regions around object proposals. |

RoI features from multiple layers, and then the object proposals generated by selective search and edgeboxes are classified by using the concatenated features. HyperNet [131], as shown in Fig. 16 (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 de-

scriptive and is more beneficial for localization and classification, but at increased computational complexity.

**(2) Detecting at multiple CNN layers.** A number of recent approaches improve detection by predicting objects of different resolutions at different layers and then combining these predictions. SSD [171] and MSCNN [24], RBFNet [169], and DSOD [238] 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.* [169] 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 [256], but combining multiple branches with different kernels and convolution layers [33]. MSCNN [24] applies deconvolution on multiple layers of a CNN to increase feature map resolution before using the layers to learn region proposals and pool features. Similar to RFBNet [169], Li *et al.* [159] proposed TridentNet by constructing a parallel multibranch architecture where each branch shares the same transformation parameters but with different receptive fields. In TridentNet, dilated convolution with different dilation rates is used to adapt the receptive fields for objects of different scales.

**(3) Combination of the above two methods.** On one hand, features from different layers are complementary to each other and can improve detection accuracy, as shown by Hypercolumns [94], HyperNet [131] and ION [11]. On the other hand, it is natural to detect objects of different scales using features of approximately the same size of features, which can be achieved by detecting large objects from low resolution while detecting small objects from high resolution. 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. This research line, found to be effective for segmentation [173, 237] and human pose estimation [190], has been widely exploited by both the state of the art (SoA) one-stage detectors and the two-stage detectors to alleviate the problem arising from scale variation across object instances. Representative methods include SharpMask [210], Deconvolutional Single Shot Detector (DSSD) [75], Feature Pyramid Network (FPN) [163], Top Down Modulation (TDM )[243], Reverse connection with Objectness prior Network (RON) [132], ZIP [152], Scale Transfer Detection Network (STDN) [313], RefineDet [301], StairNet [276], Path Aggregation Network (PANet) [170], Feature Pyramid Reconfiguration (FPR) [133], DetNet [160], Scale Aware Network (SAN) [129], Multiscale Location aware Kernel Representation (MLKP) [271] and M2Det [307], as shown in Table 7 and contrasted in Fig. 17.

Early works like FPN [163], DSSD [75], TDM [243], ZIP [152], RON [132] and RefineDet [301] construct the feature pyramid according to the inherent multiscale, pyramidal architecture of the backbone, and achieved encouraging results. As can be observed from Fig. 17 (a1) to (f1), these methods have very 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. 17 (a2) to (e2), the main difference is the design of the simple Feature Fusion Block (FFB) which handles the selection of features from different layers 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 [163] shows significant improvement as a generic feature extractor in several applications including object detection [163, 164] and instance segmentation [99]. Using FPN in a basic Faster RCNN system, SoA single-model results on the COCO detection dataset was achieved. These methods have to add additional layers to obtain multiscale features, introducing cost that can not be neglected. STDN [313] used DenseNet [114] to combine features of different layers and designed a scale transfer module to obtain feature maps with different resolutions. The scale transfer module can be directly embedded into DenseNet with little additional cost.

More recent works, such as PANet [170], FPR [133], DetNet [160], and M2Det [307] (as shown in Fig. 17 (g), (h), (i) and (j) respectively), propose to further improve on the pyramid architectures like FPN [163] from different aspects. The networks shown in Fig. 17 (g1), (h1), (i1) and (j1) are still similar to early works like FPN [163] shown in Fig. 17(a1), but becoming increasing complex.

Based on FPN [163], Liu *et al.* designed PANet [170] (Fig. 17 (g1)) by adding another bottom-up path with clean lateral connections from the low level to top ones, in order to shorten information path and enhance feature pyramid by propagating strong localization signals existing in low layers. Then, an adaptive feature pooling was proposed to aggregate features from all feature levels for each proposal. In addition, in the proposal subnetwork, a complementary branch capturing different views for each proposal is created to further improve mask prediction. These additional steps bring only slightly extra computational overhead, but are effective and make PANet reach the 1st place in the COCO 2017 Challenge Instance Segmentation task and the 2nd place in Object Detection task. Kong *et al.* proposed FPR [133] by explicitly reformulating the feature pyramid construction process (*e.g.* FPN [163]) as feature reconfiguration functions in a highly nonlinear but efficient way. As shown in Fig. 17 (h1), in stead of using a top down path to propagating strong semantic features from the top most layer down like that in FPN [163] (Fig. 17 (a1)), FPR first extracts features from multiple layers in the backbone network by adaptive concatenation and then designs a more complex FFB module (Fig. 17 (h2)) to spread strong semantics to all scales. Li *et al.* proposed DetNet [160] (Fig. 17 (i1)) by introducing dilated convolutions to the later layers of the backbone network in order to maintain high spatial resolution in deeper layers. Zhao *et al.* [307] proposed a MultiLevel Feature Pyramid Network (MLFPN) to build more effective feature pyramids for detecting objects of different scales. As can be seen from Fig. 17 (j1), features from two different layers of the backbone are firstly fused as the base feature. Then a top down path with lateral connections from the base feature is created to build the feature pyramid. As shown in Fig. 17 (j2) and (j5), the FFB module is much more complex than those in methods like FPN [163]. The FFB module involves a Thinned U-shape Module (TUM) to generate another pyramid structure. Finally, the

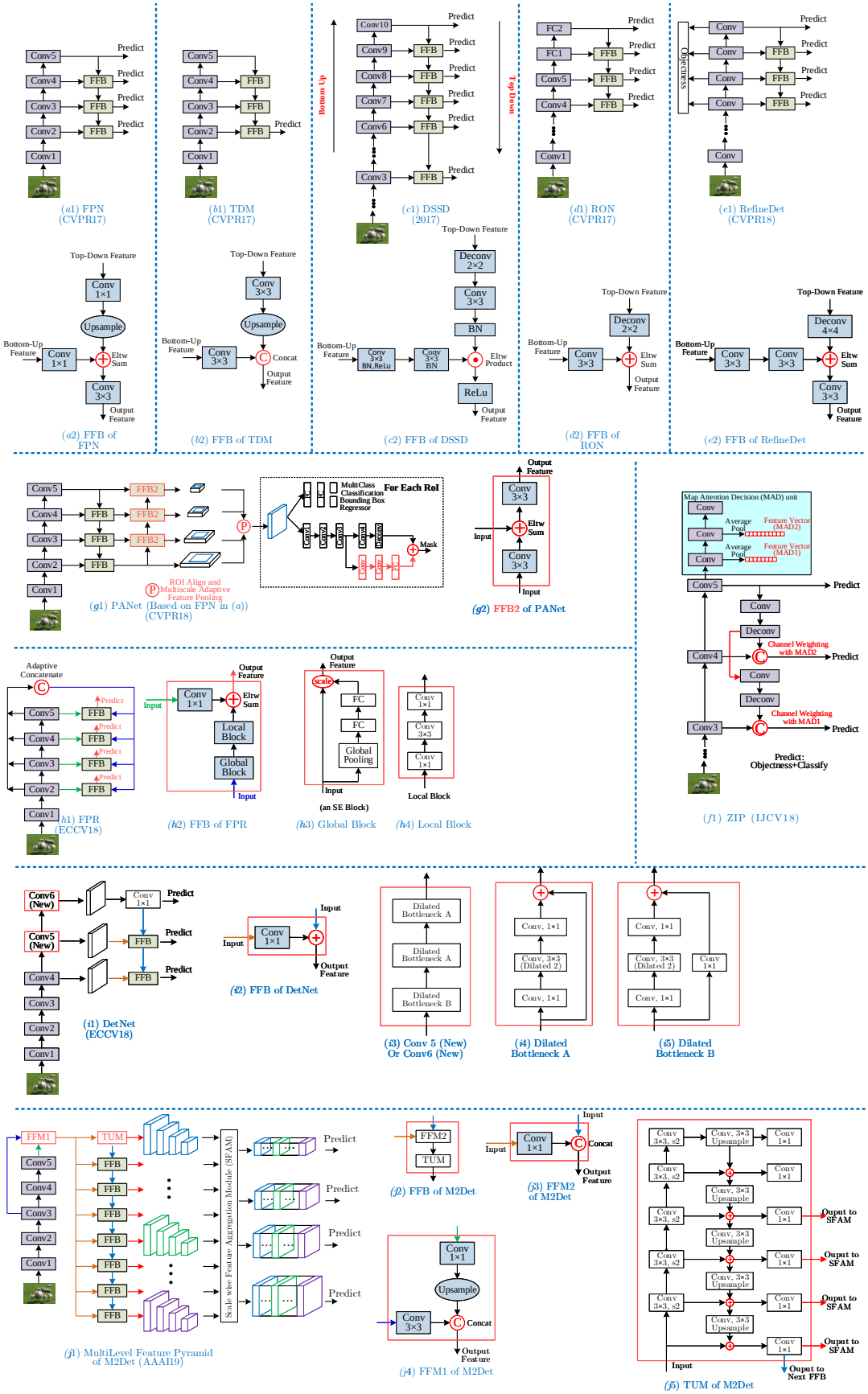



**Fig. 17** Hourglass architectures: Conv1 to Conv5 are the main Conv blocks in backbone networks such as VGG or ResNet. Comparison of a number of Feature Fusion Block (FFB) commonly used in recent approaches: FPN [163], TDM [243], DSSD [75], RON [132], RefineDet [301], ZIP [152], PANet [170], FPR [133], DetNet [160] and M2Det [307]. FFM: Feature Fusion Module, TUM: Thinned U shape Module

feature maps with equivalent sizes from multiple TUMs are gathered up to construct the final feature pyramid for object detection. The authors proposed M2Det by integrating MLFPN into SSD, and achieved better detection performance than SoA one-stage detectors.

#### *6.2.2 Handling of Other Intraclass Variations*

Powerful object representations should combine distinctiveness and robustness. A large amount of recent work has been devoted to handling the great changes in object scale, as we just reviewed in Section 6.2.1. As we discussed in Section 2.2 and summarized in Fig. 6, object detection still requires to be robust to the great intraclass variations in real world images. For ease of summarization, we group these intraclass variations into three categories.

- Geometric transformations in object scale, pose, rotation, viewpoint and part deformations;
- Occlusions;
- Image degradations, such as caused by illumination changes, blur, motion, low resolution, noise and weather conditions.

To handle these intraclass variations, the most straightforward way is to augment the training data datasets with sufficient desired variations. For instance, robustness to rotation changes can be increased by adding rotated objects in any orientation into the training data. Although more robust object representations can be learned by this means, but usually at the cost of expensive training and complex model parameters. Therefore, researchers proposed alternative solutions to alleviate these problems.

**Handling of geometric transformations.** DCNNs are inherently limited by the lack of ability to be spatially invariant to the geometric transformations of the input data [148, 168, 28]. The introduction of local max pooling layers has allowed DCNNs to enjoy small translation invariance. However, the intermediate feature maps in a DCNN are not actually invariant to large geometric transformations of the input data [148]. Therefore, lots of 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 [127, 21], rotation [21, 40, 277, 315], or both [122]. One representative work is Spatial Transformer Networks (STN) [122], which introduces a new learnable module to handle scaling, cropping, rotations, as well as nonrigid deformations via a global parametric transformation. STN has now been used in rotated text detection [122], rotated face detection and generic object detection [273]. However, most of these methods haven't been applied to generic object detection.

Although rotation invariance may be attractive in certain applications, such as scene text detection [100, 180], face detection [239], and detecting rigid objects in the domain of aerial imagery [55, 281], there are few generic object detection works focusing on rotation invariance because popular benchmark detection datasets (PASCAL VOC, ImageNet, COCO) do not present rotated images. In generic object detection, recent work to address geometric transformations and part deformations [49, 84, 184, 199, 270] can be summarized as follows.

Before deep learning, Deformable Part based Models (DPMs) [72] were successful for generic object detection, representing objects by component parts arranged in a deformable configuration. Although DPMs have been significantly outperformed by SoA object detectors, their spirits are still influencing many recent detectors deeply. 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 [49, 84, 184, 199, 270] to explicitly model object composition to improve CNN based detection. The first attempts [84, 270] 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 [199], DCN [49] and DPFCN [184] (shown in Table 7). Although similar in spirit, deformations are computed in a different ways: DeepIDNet [202] 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.* [49] 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 [184], 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.

**Handling of occlusions.** The In real world images, occlusions happen all the time. They may be caused due to the background or other object instances, and result information loss from object instances. The use of synthetic occlusions can alleviate this problem to some extent. Deformable parts idea is useful for occlusion handling, and thus deformable RoI Pooling [49, 184, 198] and deformable convolution [49] were proposed to alleviate the occlusion problem by giving more flexibility to the usually fixed geometric structures. Wang *et al.* [273] propose to learn an adversarial network that generates examples with occlusions and deformations. Context information is also helpful for dealing with occlusions. Despite these efforts, occlusion problem is far from being solved. Applying GANs for this problem might be an interesting research direction.

**Handling of image degradations.** Image noise is a common problem in many real world applications. It is frequently caused by insufficient lighting, low quality cameras, image compression and other factors, such as surveillance applications and the intentional low cost sensor on edge devices and wearable devices. While low image quality is expected to degrade results of visual recognition, most of the current methods are evaluated in a degradation free and clean environment, evidenced by the fact that PASCAL VOC, ImageNet, MS COCO and Open Images focus on relatively high quality images. According to our best knowledge, there are very limited work to address these problems recently.

## 7 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 [14, 10] that context plays an

**Table 8** Summarization of detectors that exploit context information, similar to Table 7.

| Group | Detector Name | Region Proposal | Backbone DCNN | Pipelined Used | mAP@IoU=0.5 VOC07 | mAP@IoU=0.5 VOC12 | mAP COCO | Published In | Highlights |
|---|---|---|---|---|---|---|---|---|---|
| Global Context | SegDeepM [318] | SS+CMPC | VGG16 | RCNN | VOC10 | VOC12 | – | CVPR15 | Use an additional feature extracted from an enlarged object proposal as context information; |
| | DeepIDNet [199] | SS+EB | AlexNet ZFNet | RCNN | 69.0 (07) | – | – | CVPR15 | Use image classification scores as global contextual information to refine the detection scores of each object proposal. |
| | ION [11] | SS+EB | VGG16 | Fast RCNN | 80.1 | 77.9 | 33.1 | CVPR16 | The contextual information outside the region of interest is integrated using spatial recurrent neural networks. |
| | CPF [241] | RPN | VGG16 | Faster RCNN | 76.4 (07+12) | 72.6 (07++12) | – | ECCV16 | Use semantic segmentation to provide top down feedback. |
| Local Context | MRCNN [80] | SS | VGG16 | SPPNet | 78.2 (07+12) | 73.9 (07+12) | – | ICCV15 | Extract features from multiple regions surrounding or inside the object proposals. Integrate the semantic segmentation-aware features. |
| | GBDNet [297, 298] | CRAFT [285] | Inception v2 ResNet269 PolyNet [303] | Fast RCNN | 77.2 (07+12) | – | 27.0 | ECCV16 TPAMI18 | A GBDNet module to learn the relations of multiscale contextualized regions surrounding an object proposal; GBDNet passes messages among features from different context regions through convolution between neighboring support regions in two directions; Gated functions are used to control message passing. |
| | ACCNN[153] | SS | VGG16 | Fast RCNN | 72.0 (07+12) | 70.6 (07++12) | – | TMM17 | Use LSTM to capture global context. Concatenate features from multi-scale contextual regions surrounding an object proposal. The global and local context features are concatenated for recognition. |
| | CoupleNet[319] | RPN | ResNet101 | RFCN | **82.7** (07+12) | **80.4** (07++12) | 34.4 | ICCV17 | Concatenate features from multiscale contextual regions surrounding an object proposal. Features of different contextual regions are then combined by convolution and element-wise sum. |
| | SMN [35] | RPN | VGG16 | Faster RCNN | 70.0 (07) | – | – | ICCV17 | Model object-object relationship efficiently and effectively through spatial memory network. Learn the functionality of NMS automatically. |
| | ORN [110] | RPN | ResNet101 +DCN | Faster RCNN | – | – | **39.0** | CVPR18 | Model the relations of a set of object proposals through interaction between their appearance feature and geometry. Learn the functionality of NMS automatically. |
| | SIN [172] | RPN | VGG16 | Faster RCNN | 76.0 (07+12) | 73.1 (07++12) | 23.2 | CVPR18 | Formulate object detection as graph-structured inference, where the objects are treated as nodes in a graph and relationships between the objects are modeled as edges. |

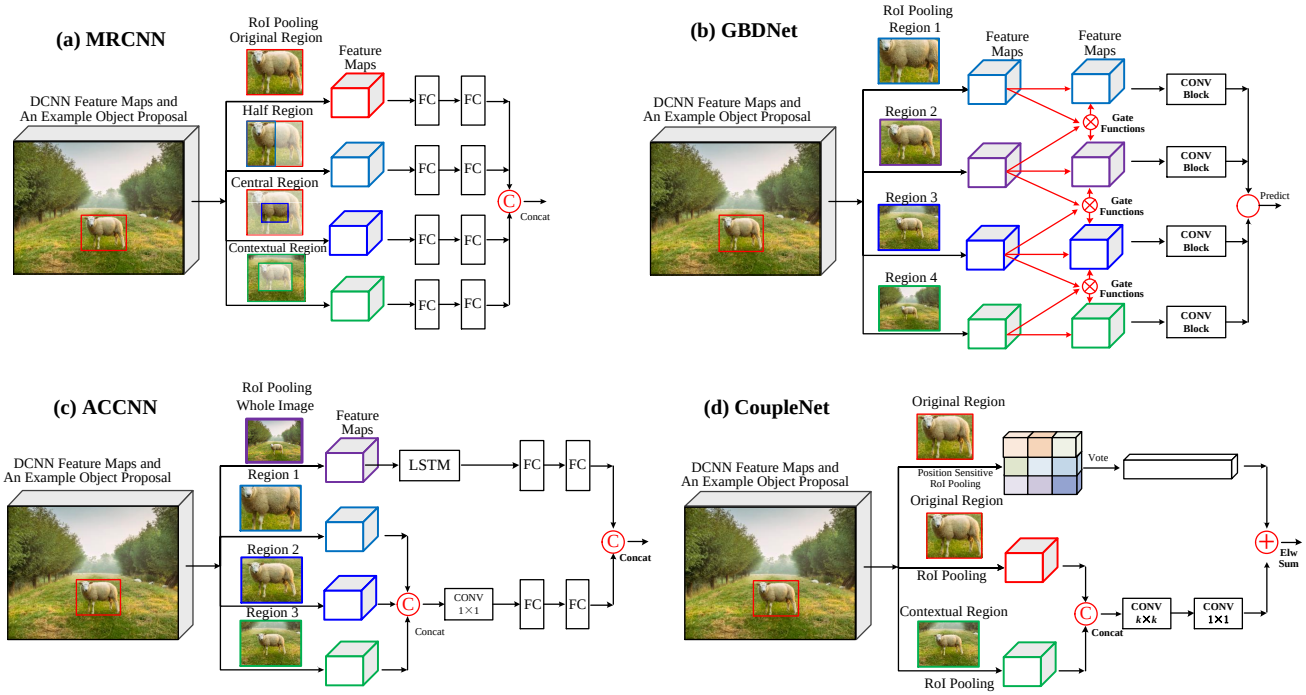



**Fig. 18** Representative approaches that explore local surrounding contextual features: MRCNN [80], GBDNet [297, 298], ACCNN [153] and CoupleNet [319], see also Table 8.

essential role in human object recognition. It is recognized that proper modeling of context helps object detection and recognition [259, 193, 33, 32, 56, 76], 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 [56, 76]. Context can broadly be grouped into one of three categories [14, 76]:

1. Semantic context: The likelihood of an object to be found in some scenes but not in others;
2. Spatial context: The 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 [34, 56, 76, 181, 189, 216, 203] preceded the prevalence of deep learning, however much of this work has not been explored in DCNN based object detectors [35, 110].

The current state of the art in object detection [225, 171, 99] detects objects without explicitly exploiting any contextual information. It is broadly agreed that DCNNs make use of contextual information implicitly [296, 308] since they learn hierarchical representations with multiple levels of abstraction. Nevertheless there is still value in exploring contextual information explicitly in DCNN based detectors [110, 35, 298], 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 [302, 76]. Representative approaches are summarized in Table 8.

### 7.1 Global Context

Global context [302, 76] 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 [199], the image classification scores were used as contextual features, and concatenated with the object detection scores to improve detection results. In ION [11], Bell *et al.* proposed to use spatial Recurrent Neural Networks (RNNs) to explore contextual information across the entire image. In SegDeepM [318], 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 out of a large pool of accurate object segmentation proposals and score the agreement between them. In [241], semantic segmentation was used as a form of contextual priming.

### 7.2 Local Context

Local context [302, 76, 216] considers the relationship among locally nearby objects, as well as 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) [35], Object Relation Network [110], and Structure Inference Network (SIN) [172]. 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 [267], ORN, which processes a set of objects simultaneously through interaction between their appearance feature and geometry, was proposed recently. 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 [172] considered two kinds of context: 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.

**Local context via larger window.** A wider range of methods has approached the problem with a simpler idea, normally by enlarging the detection window size to extract some form of local context. Representative approaches include MRCNN [80], Gated BiDirectional CNN (GBDNet) [297, 298], Attention to Context CNN (ACCNN) [153], CoupleNet [319], and Sermanet *et al.* [234].

In MRCNN [80] (Fig. 18 (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 [295] used only four contextual regions, organized in a foveal structure, where the classifiers along multiple paths are trained jointly end to end. Zeng *et al.* proposed GBDNet [297, 298] (Fig. 18 (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, GBDNet passes 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. Li *et al.* [153] presented ACCNN (Fig. 18 (c)) to utilize both global and local contextual information to facilitate object detection. To capture global context, a Multiscale Local Contextualized (MLC) subnetwork was proposed, 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.* [153] adopted a method similar to that in MRCNN [80]. As shown in Fig. 18 (d), CoupleNet [319] is conceptually similar to ACCNN [153], but built upon RFCN [48]. In addition to the original branch in RFCN [48], which captures object information with position sensitive RoI pooling, CoupleNet [319] added one branch to encode the global context information with RoI pooling.

## 8 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 [175], HOG [50] and LBP [192]), the most successful methods to object detection (*e.g.* the DPM [70]) used *sliding window* techniques [269, 50, 70, 95, 268]. 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 signifi-

cantly increases the search space [12]. 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*[13] [266, 264]. 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. Certainly, if high object recall can be achieved with only a few object proposals (like a hundred), significant speedups over the sliding window approach can be gained, enabling the use of more sophisticated classifiers. 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 method should have the following characteristics:

1. High recall, which can be achieved with only a few proposals;
2. Accurate localization, the proposals match the object bounding boxes as accurately as possible;
3. Low computing cost.

The success of object detection based on detection proposals given by selective search [266, 264] has attracted broad interest [25, 7, 3, 41, 322, 63, 134, 182].

A comprehensive review of object proposal algorithms is outside the scope of this paper, because object proposals have applications beyond object detection [6, 90, 320]. We refer interested readers to the recent surveys [106, 27] which provides an in-depth 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 are related to generic object detection.

In 2014, the integration of object proposals [266, 264] and DCNN features [136] led to the milestone RCNN [83] 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 [66], ILSVRC [230] and MS COCO [162] object detection challenges since 2014 used detection proposals [83, 199, 82, 225, 298, 99].

Among object proposal approaches based on traditional low-level cues (*e.g.,* color, texture, edge and gradients), Selective Search [264], MCG [7] and EdgeBoxes [322] are among the more popular. As the domain rapidly progressed, traditional object proposal approaches [106] (*e.g.* selective search[264] and [322]), which were adopted as external modules independent of the detectors, became the speed bottleneck of the detection pipeline [225]. An emerging class of object proposal algorithms [65, 225, 138, 79, 209, 285] 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 9.

**Bounding Box Proposal Methods** is best exemplified by the RPC method [225] of Ren *et al.*, illustrated in Fig. 19. RPN predicts object proposals by sliding a small network over the feature map of the last shared CONV layer (as shown in Fig. 19). At each sliding window location, it predicts $k$ proposals simultaneously by using $k$ anchor boxes, where each anchor box[14] is centered at some location in the image, and is associated with a particular scale and aspect ratio. Ren *et al.* [225] proposed to integrate RPN and Fast 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 [225]. RPN has been broadly selected as the proposal method by many state of the art object detectors, as can be observed from Tables 7 and 8.

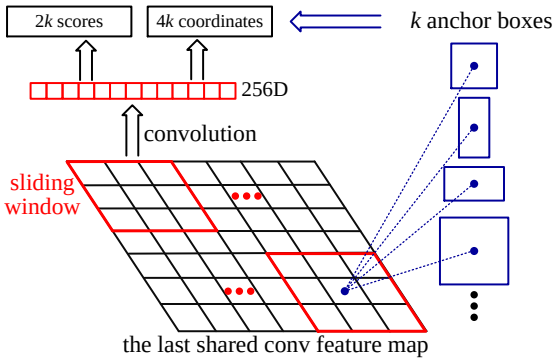



**Fig. 19** Illustration of the Region Proposal Network (RPN) introduced in [225].

Instead of fixing *a priori* a set of anchors as MultiBox [65, 255] and RPN [225], Lu *et al.* [177] 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 [131, 79, 285, 152]. Concurrent with RPN [225], Ghodrati *et al.* [79] 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 [131] 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 [285] 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.* [152] 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 [173].

[12] Sliding window based detection requires classifying around $10^4$-$10^5$ windows per image. The number of windows grows significantly to $10^6$-$10^7$ windows per image when considering multiple scales and aspect ratios.

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

[14] The terminology "an anchor box" or "an anchor" first appeared in [225].

**Table 9** Summarization of object proposal methods using DCNN. The numbers in blue color denote 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 | Highlights |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Bounding Box Object Proposal Methods | MultiBox1[65] | 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. Do not share features for detection. |
| | DeepBox [138] | VGG16 | Fast RCNN | 0.96 (1000) | 0.84 (1000) | 0.15 (1000) | – | – | 37.8 (500) (IoU@0.5) | ICCV15 | Use a light weight CNN to learn to rerank proposals generated by EdgeBox. Can run at 0.26s per image. Do not share features for detection. |
| | RPN[225, 226] | 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 | The first to generate object proposals by sharing full image convolutional features with the detection network. Most widely used object proposal method. Greatly improved the detection speed. |
| | DeepProposal[79] | 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 | Generate proposals inside a DCNN in a multiscale manner. Share features with the detection network. |
| | CRAFT [285] | VGG16 | Faster RCNN | 0.98 (300) | 0.90 (300) | 0.13 (300) | 75.7 (07+12) | 71.3 (12) | – | CVPR16 | Introduced a classification Network (*i.e.* two class Fast RCNN) cascade that comes after the RPN. Not sharing features extracted for detection. |
| | AZNet [177] | VGG16 | Fast RCNN | 0.91 (300) | 0.71 (300) | 0.11 (300) | 70.4 (07) | – | 22.3 | CVPR16 | Use coarse-to-fine search. Start from large region, then recursively search for subregions that might contain objects. This can adaptively guide computational resources to focus on subregions likely to contain objects. |
| | ZIP [152] | Inception v2 | Faster RCNN | 0.85 (300) COCO | 0.74 (300) COCO | 0.35 (300) COCO | 79.8 (07+12) | – | – | IJCV18 | Generate proposals using conv-deconv network with multilayers; Proposed a map attention decision (MAD) unit to assign the weights for features from different layers. |
| | DeNet[262] | 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 [209] | 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 [46] | VGG16 | – | – | | | 0.32 (100), 0.39(1000) | | | ECCV16 | Combine ideas of FCN [173] and DeepMask [209]. Introduce instance sensitive score maps. Need segmentation annotations to train the network. |
| | SharpMask [210] | MPN [295] | Fast RCNN | 0.39 (100), 0.53(1000) | | | 0.30 (100), 0.39(1000) | | | ECCV16 | Leverage features at multiple convolutional layers by introducing a top-down refinement module. Do not share features with detection network. Need segmentation annotations for training. |
| | FastMask[109] | 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 [171]. Use multiscale convolutional features in a deep network. Use segmentation annotations for training. |

Finally, recent work which deserves mention includes Deepbox [138], which proposed a light weight CNN to learn to rerank proposals generated by EdgeBox and is as good as using much larger networks while being much faster, and DeNet [262] which introduces a bounding box corner estimation to predict object proposals efficiently to replace RPN in a Faster RCNN style detector.

**Object Segment Proposal Methods** [209, 210] 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 [93, 47, 158]. In addition, using instance segmentation supervision can improve the performance of bounding box object detection. A pioneering work was DeepMask proposed by Pinheiro *et al.* [209], 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 [235], 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.* [210] proposed SharpMask by augmenting the DeepMask architecture with a refinement module, similar to the architectures shown in Fig. 17 (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 [173] and DeepMask [209], Dai *et al.* proposed InstanceFCN [46] for generating instance segment proposals. Similar to DeepMask, the InstanceFCN network is split into two branches, however 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 [109] to efficiently generate instance segment proposals in a one shot manner similar to SSD [171], 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 $800 \times 600$ resolution image with a slight trade off in average recall.

## 9 Other Special Issues

**Data Augmentation.** Performing data augmentation for learning DCNNs [26, 82, 83] is generally recognized to be important for visual recognition. Trivial data augmentation refers to perturbing an image by transformations that leave the underlying category unchanged, such as cropping, flipping, rotating, scaling, translating, cropping with color perturbations, and adding noise in order to

**Table 10** Representative methods for training strategies and class imbalance handling. Results on COCO are reported with Test Dev. The detection results on COCO is mAP@IoU[0.5, 0.95].

| Detector Name | Region Proposal | Backbone DCNN | Pipelined Used | VOC07 Results | VOC12 Results | COCO Results | Published In | Highlights |
|---|---|---|---|---|---|---|---|---|
| MegDet [205] | 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 [246] | RPN | DPN [37] +DCN [49] | RFCN | – | – | 48.3 | CVPR18 | A new multiscale training scheme. Empirically examined the effect of upsampling for small object detection. During training, only select objects that fit the scale of features as positive samples. |
| SNIPER [246] | RPN | ResNet101 +DCN | Faster RCNN | – | – | 47.6 | 2018 | An efficient multiscale training strategy. Process context regions around ground-truth instances at the appropriate scale. |
| OHEM [242] | SS | VGG16 | Fast RCNN | 78.9 (07+12) | 76.3 (07++12) | 22.4 | CVPR16 | A simple and effective Online Hard Example Mining algorithm to improve training of region based detectors. |
| FactorNet [200] | SS | GooglNet | RCNN | – | – | – | CVPR16 | Identify the imbalance in the number of samples for different object categories. To handle this problem, divide-and-conquer feature learning scheme is proposed. |
| Chained Cascade [23] | SS CRAFT | VGG Inceptionv2 | Fast RCNN, Faster RCNN | 80.4 (07+12) (SS+VGG) | – | – | ICCV17 | Jointly learn DCNN and multiple stages of cascaded classifiers. Save computation for both training and testing. Boost detection accuracy on PASCAL VOC 2007 and ImageNet for both fast RCNN and Faster RCNN using different region proposal methods such as CRAFT and Selective Search. |
| Cascade RCNN [23] | RPN | VGG ResNet101 +FPN | Faster RCNN | – | – | 42.8 | CVPR18 | Jointly learn DCNN and multiple stages of cascaded classifiers. When learning different cascaded classifiers, use different localization accuracy, implemented by overlap threshold, for selecting positive samples. Stack bounding box regression at multiple stages. |
| RetinaNet [164] | – | ResNet101 +FPN | RetinaNet | – | – | 39.1 | ICCV17 | Propose a novel Focal Loss which focuses training on hard examples. Well handles the problem of imbalance of positive and negative samples when training one-stage detector. |

generate additional samples of the class. By artificially enlarging the number of samples, data augmentation helps reducing overfitting and improves generalization. It can be used at training time, at test time, or both. Nevertheless, it has obvious limitations. The time required for training increases significantly, limiting its usage in some applications. In addition, data augmentation by synthesizing new training images [206, 273] is also used. However, it is hard to guarantee that the synthetic images generalize well to real images. Some researchers [62, 91] proposed to augment datasets by pasting real segmented objects into natural images. Dvornik *et al.* [61] went one step further along this direction and showed that modeling appropriately the visual context surrounding objects is crucial to place them in the right environment. Dvornik *et al.* [61] proposed a context model to automatically find appropriate locations on images to place new objects and perform data augmentation.

**Novel Training Strategies.** 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 [116, 171] 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 [116]. Recently, Singh *et al.* proposed advanced and efficient data argumentation methods SNIP [245] and SNIPER [246] to illustrate the scale invariance problem, as summarized in Table 10. 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 [246] 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. Peng *et al.* [205] studied a key factor in the training, *i.e.* minibatch size, and proposed a Large MiniBatch Object Detector called MegDet to enable the training with much larger minibatch size than before (*e.g.* from 16 to 256). To avoid the failure of convergence and significantly speedup the training process, Peng *et al.* [205] proposed a learning rate policy and Cross GPU Batch Normalization and effectively utilized multiple GPUs (*e.g.* 128 GPUs). MegDet is claimed to finish COCO training in 4 hours on 128 GPUs, reaching even higher accuracy. MegDet led to the winning of COCO 2017 Detection Challenge.

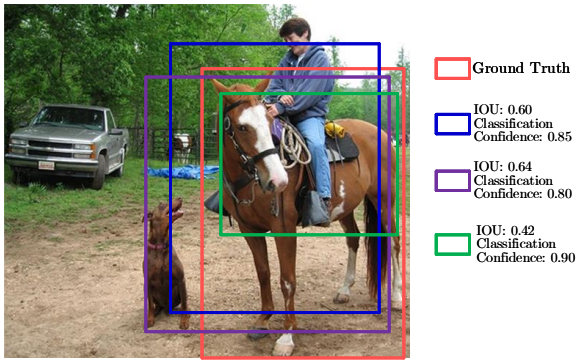



**Fig. 20** Localization error could be insufficient overlap or duplicate detections. Localization error is a frequent cause of false positives.

**Reducing Localization Error.** In object detection, the Intersection Over Union [15] (IOU) between a detected bounding box

[15] Please refer to Section 4.2 for more details on definition of IOU and evaluation criteria.

and its ground truth box is the most popular evaluation metric, and an IOU threshold (*e.g.* a typical value 0.5) is required to define positives and negatives. As can be seen from Fig. 13, in most SoA detectors [82, 171, 99, 225, 223], object detection is formulated as a multitask learning problem, *i.e.* jointly optimizing a softmax classifier which assigns object proposals with proper class labels and bounding box regressors which localize objects by maximizing IOU or other metrics between detection results and the ground truth. Bounding boxes are only a crude approximation for articulated objects, and cannot accommodate flexible objects. Consequently, background pixels are often included in a bounding box, which affects the accuracy of classification and localization. The study in [104] shows that object localization error is one of the most influential forms of error, in addition to the confusion with similar objects. *Localization error* could be insufficient overlap (smaller than the required IOU threshold, such the green box shown in Fig. 20) or duplicate detections (*i.e.* multiple overlapping detections for an object instance). Usually, some post-processing step like NonMaximum Suppression (NMS) is used for eliminating duplicate detections. However, due to misalignment, the bounding box with better localization would possibly be suppressed during NMS, leading to the poorer localization quality of objects (such the purple box shown in Fig. 20). Therefore, there are quite a few methods aiming at improving detection performance by reducing localization error.

MRCNN [80] introduces iterative bounding box regression, where a RCNN is applied several times. CRAFT [285] and AttractioNet [81] use a multistage detection subnetwork to generate accurate proposals, and forward them to a Fast RCNN. According to [23], applying bounding box regression more than twice brings little improvement. Cai and Vasconcelos proposed a multistage extension of the RCNN, namely Cascade RCNN [23]. In Cascade RCNN, the sequence of detectors are trained sequentially with increasing IOU thresholds by leveraging the observation that the output of a detector trained with a certain IOU threshold is a good distribution to train the detector of the next higher IOU threshold, in order to be sequentially more selective against close false positives. It can be built with any RCNN based detectors, and is demonstrated to achieve consistent gains (about 2 to 4 points) independently of the baseline detector strength, at a marginal increase in computation. Recently, there are some works [124, 228, 117] formulating IOU directly as the optimization objective and consider it as an indicator of localization confidence. In addition, there are a few works [18, 101, 107, 263] proposed to improve NMS results, such as Soft NMS [18] and learning NMS [107].

**Class Imbalance Handling.** Different from image classification, object detection has another unique problem, *i.e.* the serious imbalance between the number of labeled object instances and the number of background examples (image regions not belonging to any object class of interest). Most background examples are easy negatives. This imbalance can make the training very inefficient, and the large number of easy negatives tend to overwhelm the training. Traditionally in object detection, this issue is typically addressed via techniques such as bootstrapping [252]. Recently, this problem has also acquired some attention [149, 164, 242]. Because the region proposal stage rapidly filters out most background regions and proposes a small number of object candidates, this class imbalance issue is mitigated to some extent in two stage detectors [83, 82, 225, 99]. Despite this, example mining approaches like Online Hard Example Mining (OHEM) [242] are used to maintain a reasonable balance between foreground and background. In the case of one stage object detectors [223, 171], this imbalance is extremely serious (*e.g.* 100,000 background examples to every one object). To address this, Lin *et al.* [164] proposed Focal Loss to address it by rectifying the Cross Entropy loss such that it downweights the loss assigned to correctly classified examples. Li *et al.* [149] studied this issue from the perspective of gradient norm distribution, and proposed a Gradient Harmonizing Mechanism (GHM) to handle it.

## 10 Discussion and Conclusion

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. At the end of this review, we would like to firstly discuss the SoA performance of object detection in Section 10.1, then provide an overall discussion on a number of key factors in Section 10.2 and finally present our view on a few future research directions in Section 10.3.

### 10.1 Performance

A large variety of detectors has appeared in the last several years, and the introduction of standard benchmarks such as PASCAL VOC [66, 67], ImageNet [230] and COCO [162] has made it easier to compare detectors with respect to accuracy. As can be seen from our earlier discussion from Section 5 to Section 9, it might be unfair to compare detectors in terms of their originally reported performance (*e.g.* accuracy, speed), as they can differ in fundamental / contextual respects, including the following:

- Meta detection frameworks, such as RCNN [83], Fast RCNN [82], Faster RCNN [225], RFCN [48], Mask RCNN [99], YOLO [223] and SSD [171];
- Backbone networks such as VGG [244], Inception [256, 121, 257], ResNet [98], ResNeXt [284], and Xception [43] *etc.* listed in Table 6;
- Innovations such as multilayer feature combination [163, 243, 75], deformable convolutional networks [49], deformable RoI pooling [199, 49], heavier heads [227, 205], and lighter heads [161];
- Pretraining with datasets such as ImageNet [230], COCO [162], Places [311], JFT [103] and Open Images [135]
- Different detection proposal methods and different numbers of object proposals;
- Train/test data augmentation "tricks" such as multicrop, horizontal flipping, multiscale images, novel multiscale training strategies [245, 246] *etc*, and model ensembling.

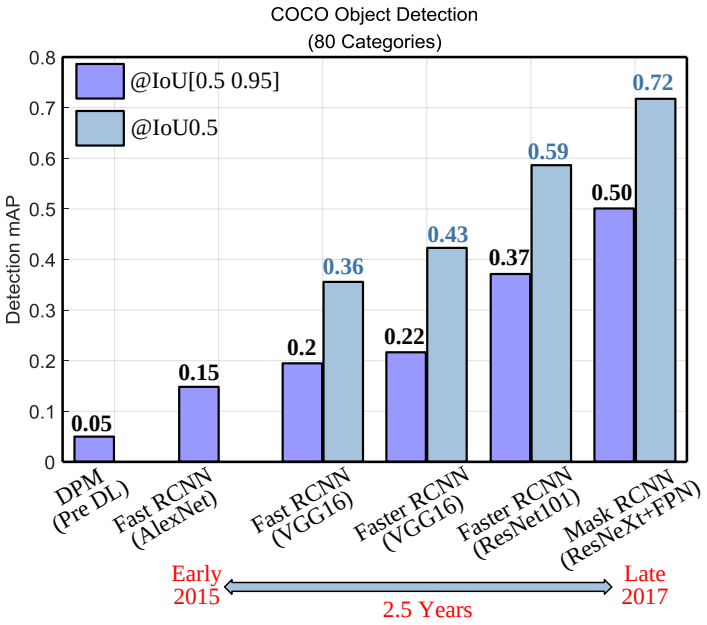



**Fig. 21** Evolution of object detection performance on COCO (Test-Dev results). Results are quoted from [82, 99, 226] accordingly. The backbone network, the design of detection framework and the availability of good and large scale datasets are the three most important factors in detection.

Although it may be impractical to compare every recently proposed detector, it is nevertheless 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 [116] of the trade off between accuracy and speed of three main families of detectors (Faster RCNN [225], RFCN [48] and SSD [171]) by varying the backbone network, image resolution, the number of box proposals; *etc*.

As can be seen from Tables 7, 8, 9, 10 and Table 11, 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. 3 and 21 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 [120, 185, 204]. The competition winner of the open image challenge object detection task achieved 61.71% mAP in the public leader board and 58.66% mAP on the private leader board, obtained by combining the detection results of several two-stage detectors including Fast RCNN [82], Faster RCNN [225], FPN [163], Deformable RCNN [49], and Cascade RCNN [23]. 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 [136] was proposed, the Top5 error on ImageNet classification [230] with 1000 classes has dropped from 16% to 2%, as shown in Fig. 15. However, the mAP of the best performing detector [205] (which is only trained to detect 80 classes) on COCO [162] has reached 73%, even at 0.5 IoU, illustrating how object detection is much harder than image classification. The accuracy and robustness achieved by the state of the art detectors is far from satisfying the requirements of real world applications, so there remains significant room for future improvement.

### 10.2 Summary and Discussion

With hundreds of references and many dozens of methods within the scope and discussed throughout this paper, in this section we would like to focus on the key factors which emerged.

**(1) Detection Frameworks: Two Stage vs. One Stage**

In Section 5 we identified two major categories of detection frameworks: region based (two stage) and unified (one stage). Here an overview of their respective advantages and disadvantages:

- When large computational complexity allows, two-stage detectors generally produce higher detection accuracies than one-stage detectors, evidenced by the fact that most methods (including winning approaches) used in famous detection challenges like MS COCO [162] and Open Images [139] are predominantly based on two stage frameworks because their structure is more flexible and better suited for a refinement stage for region based classification. The most widely used frameworks are Faster RCNN [225], RFCN [48], Mask RCNN [99] and Mask RCNN.
- It has been shown in [116] that the detection accuracy of one-stage SSD [171] is less sensitive to the quality of the backbone network than representative two stage frameworks.
- There is a general impression that one-stage detectors like YOLO [223] and SSD [171] are much simpler and faster than two-stage ones. Potential reasons include not using a preprocessing algorithm to generate object proposals, using lightweight backbone networks, performing prediction with fewer candidate regions (*e.g.,* YOLO), and making the classification subnetwork fully convolutional (*e.g.,* SSD). However, two-stage detectors can run in real time with the introduction of similar techniques. In any event, whether one stage or two, the most time consuming step is the feature extractor (backbone network) [142, 225]; according to [116], two-stage Faster RCNN can be sped up significantly without significant accuracy loss by using fewer proposals, making it competitive with one-stage SSD.
- It has been shown [116, 223, 171] that one stage frameworks like YOLO and SSD typically have much poorer performance on detecting small objects than two-stage architectures like Faster RCNN and RFCN, but are competitive in detecting large objects, even outperforming with lightweight backbone networks.

There have been many attempts to build better (faster, more accurate, or more robust) detectors by attacking each stage of the detection framework. No matter whether one, two or multiple stages, the design of the detection framework has converged towards a number of crucial design choices:

- *Fully convolutional pipeline* — simple, efficient, and elegant
- *Exploring complementary information from other correlated tasks* — *e.g.*, Mask RCNN [99]
- *Sliding window* – making it easier to learn the classifier [225]

- *Fuse information* from different layers of the backbone Network.

The evidence from recent success of cascade for object detection and instance segmentation on COCO [31] and other challenges, *e.g.* WIDER Challenge [176], has shown that multistage object detection could be a future framework for speed-accuracy trade-off. We recommend using speed-accuracy for comparison of object detection approaches in the future. A teaser investigation is being done in the recent WIDER Challenge 2019 [176].

**(2) Backbone Networks**

As discussed in Section 6.1, backbone networks are one of the main driving forces behind the rapid improvement of detection performance, because of the key role played by discriminative object feature representation. Backbone networks are pretrained on image classification data and then transferred to object detection. Generally deeper backbones such as ResNet [98], ResNeXt [284], InceptionResNet [258] perform better, however they are computationally more expensive and require much more data and massive computing for training. Some backbones [108, 119, 304] were proposed to focus on speed instead, such as MobileNet [108] which has been shown to achieve VGGNet16 accuracy on ImageNet with only $\frac{1}{30}$ the computational cost and model size.

Backbone network using pretraining on ImageNet is the blessing, but could be the devil constraining the design of specific deep model suitable for object detection. In the research community, training from scratch becomes possible, as more training data and better training strategy are available [278, 179, 178]. More backbone deep models will be specifically designed for object detection.

**(3) Improving the Robustness of Object Representation**

The variation of real world images is a key factor in making object recognition so challenging. The variation includes lighting changes, object pose, object deformations, background clutter, heavy occlusions, image blur, resolution, noise, and camera distortions.

**(3.1) Object Scales and Small Objects**

Large variations of objects in scale, particularly small objects, pose great challenge. A summary and discussion on the main strategies identified in Section 6.2:

- Using image pyramids: It is simple and effective, pyramids help enlarge small objects and shrink large ones. However, it is also computationally expensive, they are nevertheless commonly used during inference for better accuracy.
- Using features from convolutional layers of different resolutions: In early work like SSD [171], predictions are performed independently and no information from other layers is combined or merged. Now it is quite standard to combine features from different layers, e.g. in FPN [163].
- Using dilated convolutions [160, 159]: A simple and effective method to incorporate broader context and maintain high resolution feature maps. It also increases the computational complexity.
- Using anchor boxes of different scales and aspect ratios: It has drawbacks of including many parameters, scale and aspect ratios usually heuristically determined, a large number of anchors.
- Upscaling: Particularly for the detection of small objects, images may be upscaled, high resolution shallow networks can be developed. Although super-resolution techniques can increase the resolution of image, it is still unclear whether super-resolution improves detection accuracy or not.

Small objects inherently lose visual information, which is hard to be recovered. Despite recent advances, the detection accuracy for small objects is still much lower than that of large objects. Therefore, detection of small objects remains one of the key challenges in object detection. Perhaps different requirements on localization accuracy can be developed for evaluating the detection accuracy of objects of different scales, because many applications, e.g. autonomous car driving, only need to identify the existence of small objects within a large region, while exact localization is not necessary.

**(3.2) Deformation, occlusion, and other factors**

As discussed in Section 2.2, there are lots of designs handling geometric transformation, occlusions, and deformation. To handle geometric transformation and deformation, existing methods are mainly based on two paradigms. The first is spatial transformer network, which uses regression to obtain deformation field and then warp features according to the deformation field [49]. The second is based on deformable part-based model [72], which finds the max response to a part filter with spatial constraint taken into consideration [199, 84, 270].

Rotation invariance may be attractive in certain applications such as detection from satellite image. But there are few generic object detection works focusing on rotation invariance, because objects popular benchmark detection datasets (PASCAL VOC, ImageNet, COCO) do not have large variation in rotation. Occlusion handling is intensively studied in face detection and pedestrian detection. But the study on occlusion handling for generic object detection is few. The annotation mechanism for generic object detection is different from that for face and pedestrian. If the legs of a person are fully occluded, then the legs are included in the annotation for pedestrian detection but excluded in the annotation for generic object detection. This makes occlusion handling for generic object detection different from that for face or pedestrian. In general, despite recent advances, deep networks are still limited by the lack of robustness to many variations which significantly constrains their real-world applications. The other reason might be the limited number of samples with rich variation in deformation, occlusion, and other factors.

**(4) Context Reasoning**

As introduced in Section 7, objects in the wild typically coexist with other objects and environments. It has been recognized that contextual information (object relations, global scene statistics) helps object detection and recognition [193], especially for small objects, occluded objects, and poor image quality. There was extensive work preceding deep learning [181, 189, 216, 56, 76]. New deep learning approaches using context appear each year [80, 297, 298, 35, 110].

How to efficiently and effectively incorporate contextual information remains to be explored, ideally guided by how human use the context in recognizing objects. The recent progress on building scene graph [157] may be another way of building structured understanding of objects in the scene. The full segmentation of objects and scene using panoptic segmentation [130] is expected provide richer contextual information for object detection.

**(5) Detection Proposals**

Detection proposal algorithms, which significantly reduce search spaces, played an important role in object detection. As recommended in [106], future detection proposals will surely have to improve in repeatability, recall, localization accuracy, and speed. Since the success of RPN [225], CNN based detection proposal generation methods have dominated region proposal.

After object proposal generation and detection have been integrated into the same framework in [225], the distinction between detection and proposal generation is becoming blurred. Especially when considering detection proposal as one step of cascade in the detection frameworks with multiple stages of cascade. New detection proposals are recommended to show their improvement for object detection, instead of evaluating detection proposals alone. In the future, new detection proposals will encounter the following question: Are there advantages of proposing a single network for object proposal and another network for classifying the proposals?

**(6) Other Factors**

As discussed in Section 9, there are many other factors affecting object detection, such as data augmentation, novel training strategies, ensembling backbone models, ensembling multiple detection frameworks, incorporating information from other related tasks, methods for reducing localization error, handling the huge imbalance between positive and negative samples, mining of hard negative samples, and improving loss functions for accurate localization.

### 10.3 Research Directions

Despite the recent tremendous progress in the field of object detection, the technology remains significantly more primitive than human vision and cannot satisfactorily address real world challenges like those of Section 2.2. We see a number of long-standing challenges:

- Working in an open world; *e.g.,* being robust to any number of environmental changes, being able to evolve etc.
- Object detection under constrained conditions; *e.g.,* learning from weakly labeled data or few bounding box annotations, wearable devices, unseen object categories etc.
- Object detection in other modalities; *e.g.,* video, RGBD images, 3D point clouds, lidar, remotely sensed imagery *etc*.

Based on these challenges, we see the following directions of future research:

**(1) Open World Learning:** The ultimate goal is to develop object detection capable of accurately and efficiently recognizing and localizing instances in thousands or more object categories in open-world scenes, at a level competitive with the human visual system. Recent object detection algorithms are blind, in principle, to object categories outside of their training dataset, although ideally there should be the ability to recognize novel object categories [140, 92]. Current detection datasets [66, 230, 162] contain only dozens to hundreds of categories, which is significantly smaller than those which can be recognized by humans. New larger-scale datasets with significantly more categories will need to be developed.

**(2) Better and More Efficient Detection Frameworks:** One of the reasons for the success in generic object detection has been the development of superior detection frameworks, both region-based (RCNN [83], Fast RCNN [82], Faster RCNN [225], Mask RCNN [99]) and one-stage detectors (YOLO [223], SSD [171]). Region-based detectors have higher accuracy, but are more computationally intensive. One-stage detectors have the potential to be faster and simpler, but with lower accuracy. One possible limitation is that object detectors depend heavily on the underlying backbone networks, which have been optimized for image classification, causing a learning bias due to the differences between classification and detection. Learning object detectors from scratch could be helpful for new detection frameworks.

**(3) Compact and Efficient CNN Features:** CNNs have increased remarkably in depth, from several layers (*e.g.,* AlexNet [137]) to hundreds of layers (*e.g.,* ResNet [98], DenseNet [114]). These networks have millions to hundreds of millions of parameters, requiring massive data and GPUs for training. In order to remove the redundancy in the model, there has been growing research interest in designing compact and lightweight networks [29, 4, 115, 108, 165, 293], network compression and acceleration [42, 118, 248, 151, 154, 275].

**(4) Automatic Neural Architecture Search:** Deep learning bypasses manual feature engineering which requires human experts with strong domain knowledge. However, DCNNs requires similarly significant expertise. It is natural to consider automated design of detection backbone architectures in order to reduce the demand for human experts. The recent Automated Machine Learning (AutoML) [215] has brought opportunities to achieve such a goal, and AutoML has been successfully applied to image classification and object detection [22, 39, 78, 167, 323, 324].

**(5) Object Instance Segmentation:** Continuing the trend of richer and more detailed understanding of image content, a next challenge would be to tackle pixel-level object instance segmentation [162, 99, 113], as object instance segmentation can play an important role in potential applications that require the precise boundaries of individual instances.

**(6) Weakly Supervised Detection:** Current state-of-the-art detectors employ fully supervised models learned from labeled data with object bounding boxes or segmentation masks [67, 162, 230, 162]. However, fully supervised learning has serious limitations, where the collection of bounding box annotations is labor intensive, especially when the number of object categories is large. Fully supervised learning is not scalable in the absence of fully labeled training data. Therefore, it is valuable to study how the power of CNNs can be leveraged in weakly supervised detection, where only weakly annotated data are provided [17, 53, 240].

**(7) Few Shot Object Detection:** The success of deep detectors relies heavily on gargantuan amounts of annotated training data. When the labeled data are scarce, the performance of deep detectors frequently deteriorates and fails to generalize well. However, the cost of annotating bounding boxes over hundreds of classes is high, particularly so in working with video. In contrast, humans (even children) can learn a visual concept quickly from very few given examples and can often generalize well [16, 140, 69]. Therefore, the ability to learn from few examples, few shot classification, is very appealing [30, 59, 73, 125, 140, 224, 233].

**(8) Zero Shot Object Detection:** Zero shot object detection localizes and recognizes object classes that have never been seen before [9, 51, 218, 217]. Such detection is essential for lifelong learning machines that need to intelligently and incrementally discover new object categories.

**(9) Object Detection in Other Modalities:** Most of the detectors developed have been using still 2D images. Object detection in other modalities, such as RGBD images, videos, 3D point clouds, *etc*., is of great importance and relevance in domains such as autonomous vehicles, unmanned aerial vehicles, and robotics. These modalities raise new challenges in effectively using depth [36, 207, 282, 279], video [68, 126], and point clouds [213, 214].

**(10) Universal Object Detection:** Recently, there has been increasing effort in learning *universal representations*, those which are effective in multiple image domains, such as natural images, videos, aerial images, medical CT images etc. [220, 221]. Most current research focuses on image classification, rarely targeting object detection [274]. Existing detectors are usually domain specific, since high detection accuracy requires a detector specialized on the target dataset. Object detection independent of image domain and cross-domain object detection could be important future directions.

The research field of generic object detection is still far from complete. Given the breakthroughs over the past five years, we are optimistic of future opportunities.

## 11 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 17
2. Alexe B., Deselaers T., Ferrari V. (2010) What is an object? In: CVPR, pp. 73–80 24
3. Alexe B., Deselaers T., Ferrari V. (2012) Measuring the objectness of image windows. IEEE TPAMI 34(11):2189–2202 24
4. Alvarez J., Salzmann M. (2016) Learning the number of neurons in deep networks. In: NIPS, pp. 2270–2278 30
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 24
7. Arbeláez P., Pont-Tuset J., Barron J., Marques F., Malik J. (2014) Multiscale combinatorial grouping. In: CVPR, pp. 328–335 24
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 17
9. Bansal A., Sikka K., Sharma G., Chellappa R., Divakaran A. (2018) Zero shot object detection. In: ECCV 31
10. Bar M. (2004) Visual objects in context. Nature Reviews Neuroscience 5(8):617–629 21
11. 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 17, 18, 19, 22, 23
12. Belongie S., Malik J., Puzicha J. (2002) Shape matching and object recognition using shape contexts. IEEE TPAMI 24(4):509–522 5
13. Bengio Y., Courville A., Vincent P. (2013) Representation learning: A review and new perspectives. IEEE TPAMI 35(8):1798–1828 2, 3, 6, 7, 15
14. Biederman I. (1972) Perceiving real world scenes. IJCV 177(7):77–80 21, 22
15. Biederman I. (1987) Recognition by components: a theory of human image understanding. Psychological review 94(2):115 6
16. Biederman I. (1987) Recognition by components: a theory of human image understanding. Psychological review 94(2):115 30
17. Bilen H., Vedaldi A. (2016) Weakly supervised deep detection networks. In: CVPR, pp. 2846–2854 30
18. Bodla N., Singh B., Chellappa R., Davis L. S. (2017) SoftNMS improving object detection with one line of code. In: ICCV, pp. 5562–5570 27
19. Borji A., Cheng M., Jiang H., Li J. (2014) Salient object detection: A survey. arXiv: 14115878v1 1:1–26 3
20. Bourdev L., Brandt J. (2005) Robust object detection via soft cascade. In: CVPR, vol 2, pp. 236–243 13
21. Bruna J., Mallat S. (2013) Invariant scattering convolution networks. IEEE TPAMI 35(8):1872–1886 21
22. Cai H., Yang J., Zhang W., Han S., Yu Y. (2018) Path level network transformation for efficient architecture search 30
23. Cai Z., Vasconcelos N. (2018) Cascade RCNN: Delving into high quality object detection. In: CVPR 13, 26, 27, 28
24. 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 17, 18, 19
25. Carreira J., Sminchisescu C. (2012) CMPC: Automatic object segmentation using constrained parametric mincuts. IEEE TPAMI 34(7):1312–1328 24
26. Chatfield K., Simonyan K., Vedaldi A., Zisserman A. (2014) Return of the devil in the details: Delving deep into convolutional nets. In: BMVC 25
27. Chavali N., Agrawal H., Mahendru A., Batra D. (2016) Object proposal evaluation protocol is gameable. In: CVPR, pp. 835–844 10, 24
28. Chellappa R. (2016) The changing fortunes of pattern recognition and computer vision. Image and Vision Computing 55:3–5 21
29. Chen G., Choi W., Yu X., Han T., Chandraker M. (2017) Learning efficient object detection models with knowledge distillation. In: NIPS 30
30. Chen H., Wang Y., Wang G., Qiao Y. (2018) LSTD: A low shot transfer detector for object detection. In: AAAI 30
31. Chen K., Pang J., Wang J., Xiong Y., Li X., Sun S., Feng W., Liu Z., Shi J., Ouyang W., et al. (2019) Hybrid task cascade for instance segmentation. In: CVPR 13, 29
32. Chen L., Papandreou G., Kokkinos I., Murphy K., Yuille A. (2015) Semantic image segmentation with deep convolutional nets and fully connected CRFs. In: ICLR 22
33. 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 17, 19, 22
34. Chen Q., Song Z., Dong J., Huang Z., Hua Y., Yan S. (2015) Contextualizing object detection and classification. IEEE TPAMI 37(1):13–27 23
35. Chen X., Gupta A. (2017) Spatial memory for context reasoning in object detection. In: ICCV 22, 23, 29
36. 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 31
37. Chen Y., Li J., Xiao H., Jin X., Yan S., Feng J. (2017) Dual path networks. In: NIPS, pp. 4467–4475 17, 26
38. Chen Y., Rohrbach M., Yan Z., Yan S., Feng J., Kalantidis Y. (2019) Graph based global reasoning networks. In: CVPR 17
39. Chen Y., Yang T., Zhang X., Meng G., Pan C., Sun J. (2019) DetNAS: Neural architecture search on object detection. arXiv:190310979 30

40. Cheng G., Zhou P., Han J. (2016) RIFDCNN: Rotation invariant and fisher discriminative convolutional neural networks for object detection. In: CVPR, pp. 2884–2893 21
41. Cheng M., Zhang Z., Lin W., Torr P. (2014) BING: Binarized normed gradients for objectness estimation at 300fps. In: CVPR, pp. 3286–3293 24
42. 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 30
43. Chollet F. (2017) Xception: Deep learning with depthwise separable convolutions. In: CVPR, pp. 1800–1807 17, 27
44. Cinbis R., Verbeek J., Schmid C. (2017) Weakly supervised object localization with multi-fold multiple instance learning. IEEE TPAMI 39(1):189–203 12
45. 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 3, 5
46. Dai J., He K., Li Y., Ren S., Sun J. (2016) Instance sensitive fully convolutional networks. In: ECCV, pp. 534–549 25
47. Dai J., He K., Sun J. (2016) Instance aware semantic segmentation via multitask network cascades. In: CVPR, pp. 3150–3158 25
48. Dai J., Li Y., He K., Sun J. (2016) RFCN: object detection via region based fully convolutional networks. In: NIPS, pp. 379–387 9, 13, 17, 23, 27, 28, 37
49. Dai J., Qi H., Xiong Y., Li Y., Zhang G., Hu H., Wei Y. (2017) Deformable convolutional networks. In: ICCV 18, 21, 26, 27, 28, 29
50. Dalal N., Triggs B. (2005) Histograms of oriented gradients for human detection. In: CVPR, vol 1, pp. 886–893 3, 5, 9, 15, 23
51. Demirel B., Cinbis R. G., Ikizler-Cinbis N. (2018) Zero shot object detection by hybrid region embedding. In: BMVC 31
52. 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 6, 7, 8, 17
53. Diba A., Sharma V., Pazandeh A. M., Pirsiavash H., Van Gool L. (2017) Weakly supervised cascaded convolutional networks. In: CVPR, vol 3, p. 9 30
54. 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 3, 15
55. Ding J., Xue N., Long Y., Xia G., Lu Q. (2018) Learning RoI transformer for detecting oriented objects in aerial images. In: CVPR 21
56. Divvala S., Hoiem D., Hays J., Efros A., Hebert M. (2009) An empirical study of context in object detection. In: CVPR, pp. 1271–1278 22, 23, 29
57. 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
58. 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 17
59. Dong X., Zheng L., Ma F., Yang Y., Meng D. (2018) Few example object detection with model communication. IEEE TPAMI 30
60. Duan K., Bai S., Xie L., Qi H., Huang Q., Tian Q. (2019) CenterNet: Keypoint triplets for object detection. arXiv preprint arXiv:190408189 15
61. Dvornik N., Mairal J., Schmid C. (2018) Modeling visual context is key to augmenting object detection datasets. In: ECCV, pp. 364–380 26
62. Dwibedi D., Misra I., Hebert M. (2017) Cut, paste and learn: Surprisingly easy synthesis for instance detection. In: ICCV, pp. 1301–1310 26
63. Endres I., Hoiem D. (2010) Category independent object proposals 24
64. Enzweiler M., Gavrila D. M. (2009) Monocular pedestrian detection: Survey and experiments. IEEE TPAMI 31(12):2179–2195 2, 3
65. Erhan D., Szegedy C., Toshev A., Anguelov D. (2014) Scalable object detection using deep neural networks. In: CVPR, pp. 2147–2154 10, 24, 25
66. 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, 3, 4, 5, 7, 9, 24, 27, 30
67. 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 7, 8, 9, 27, 30
68. Feichtenhofer C., Pinz A., Zisserman A. (2017) Detect to track and track to detect. In: ICCV, pp. 918–927 31
69. FeiFei L., Fergus R., Perona P. (2006) One shot learning of object categories. IEEE TPAMI 28(4):594–611 30
70. Felzenszwalb P., McAllester D., Ramanan D. (2008) A discriminatively trained, multiscale, deformable part model. In: CVPR, pp. 1–8 9, 23
71. Felzenszwalb P., Girshick R., McAllester D. (2010) Cascade object detection with deformable part models. In: CVPR, pp. 2241–2248 13
72. Felzenszwalb P., Girshick R., McAllester D., Ramanan D. (2010) Object detection with discriminatively trained part based models. IEEE TPAMI 32(9):1627–1645 3, 9, 17, 21, 29
73. Finn C., Abbeel P., Levine S. (2017) Model agnostic meta learning for fast adaptation of deep networks. In: ICML, pp. 1126–1135 30
74. Fischler M., Elschlager R. (1973) The representation and matching of pictorial structures. IEEE Transactions on computers 100(1):67–92 1, 5
75. Fu C.-Y., Liu W., Ranga A., Tyagi A., Berg A. C. (2017) DSSD: Deconvolutional single shot detector. In: arXiv preprint arXiv:1701.06659 15, 17, 18, 19, 20, 27
76. Galleguillos C., Belongie S. (2010) Context based object categorization: A critical survey. Computer Vision and Image Understanding 114:712–722 3, 22, 23, 29
77. 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
78. Ghiasi G., Lin T., Pang R., Le Q. (2019) NASFPN: learning scalable feature pyramid architecture for object detection. arXiv:190407392 30
79. 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 24, 25
80. Gidaris S., Komodakis N. (2015) Object detection via a multiregion and semantic segmentation aware CNN model. In: ICCV, pp. 1134–1142 15, 22, 23, 27, 29
81. Gidaris S., Komodakis N. (2016) Attend refine repeat: Active box proposal generation via in out localization. In: BMVC 18, 27
82. Girshick R. (2015) Fast R-CNN. In: ICCV, pp. 1440–1448 2, 9, 11, 12, 13, 15, 16, 17, 24, 25, 27, 28, 30, 37
83. 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, 3, 5, 6, 9, 10, 11, 13, 14, 15, 16, 17, 24, 25, 27, 30, 37
84. Girshick R., Iandola F., Darrell T., Malik J. (2015) Deformable part models are convolutional neural networks. In: CVPR, pp. 437–446 21, 29
85. 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 10, 11, 17
86. Goodfellow I., Bengio Y., Courville A. (2016) Deep Learning. MIT press 6
87. Grauman K., Darrell T. (2005) The pyramid match kernel: Discriminative classification with sets of image features. In: ICCV, vol 2, pp. 1458–1465 11
88. Grauman K., Leibe B. (2011) Visual object recognition. Synthesis lectures on artificial intelligence and machine learning 5(2):1–181 1, 2, 3
89. 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, 6, 16
90. Guillaumin M., Küttel D., Ferrari V. (2014) Imagenet autoannotation with segmentation propagation. International Journal of Computer Vision 110(3):328–348 24
91. Gupta A., Vedaldi A., Zisserman A. (2016) Synthetic data for text localisation in natural images. In: CVPR, pp. 2315–2324 26
92. Hariharan B., Girshick R. B. (2017) Low shot visual recognition by shrinking and hallucinating features. In: ICCV, pp. 3037–3046 30
93. Hariharan B., Arbeláez P., Girshick R., Malik J. (2014) Simultaneous detection and segmentation. In: ECCV, pp. 297–312 25
94. Hariharan B., Arbeláez P., Girshick R., Malik J. (2016) Object instance segmentation and fine grained localization using hypercolumns. IEEE TPAMI 12, 15, 17, 19
95. Harzallah H., Jurie F., Schmid C. (2009) Combining efficient object localization and image classification. In: ICCV, pp. 237–244 9, 23
96. 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, 11, 16, 17, 37
97. 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 14
98. He K., Zhang X., Ren S., Sun J. (2016) Deep residual learning for image recognition. In: CVPR, pp. 770–778 3, 13, 15, 16, 27, 29, 30
99. He K., Gkioxari G., Dollár P., Girshick R. (2017) Mask RCNN. In: ICCV 13, 15, 19, 23, 24, 27, 28, 30, 37
100. He T., Tian Z., Huang W., Shen C., Qiao Y., Sun C. (2018) An end to end textspotter with explicit alignment and attention. In: CVPR, pp. 5020–5029 21
101. He Y., Zhu C., Wang J., Savvides M., Zhang X. (2019) Bounding box regression with uncertainty for accurate object detection. In: CVPR 27
102. Hinton G., Salakhutdinov R. (2006) Reducing the dimensionality of data with neural networks. science 313(5786):504–507 1
103. Hinton G., Vinyals O., Dean J. (2015) Distilling the knowledge in a neural network. arXiv:150302531 17, 27
104. Hoiem D., Chodpathumwan Y., Dai Q. (2012) Diagnosing error in object detectors. In: ECCV, pp. 340–353 9, 27
105. 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
106. Hosang J., Benenson R., Dollr P., Schiele B. (2016) What makes for effective detection proposals? IEEE TPAMI 38(4):814–829 10, 24, 30
107. Hosang J., Benenson R., Schiele B. (2017) Learning nonmaximum suppression. In: ICCV 27
108. 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 16, 29, 30
109. Hu H., Lan S., Jiang Y., Cao Z., Sha F. (2017) FastMask: Segment multiscale object candidates in one shot. In: CVPR, pp. 991–999 25
110. Hu H., Gu J., Zhang Z., Dai J., Wei Y. (2018) Relation networks for object detection. In: CVPR 22, 23, 29
111. Hu J., Shen L., Sun G. (2018) Squeeze and excitation networks. In: CVPR 15, 16
112. Hu P., Ramanan D. (2017) Finding tiny faces. In: CVPR, pp. 1522–1530 2
113. Hu R., Dollár P., He K., Darrell T., Girshick R. (2018) Learning to segment every thing. In: CVPR 30
114. Huang G., Liu Z., Weinberger K. Q., van der Maaten L. (2017) Densely connected convolutional networks. In: CVPR 15, 16, 19, 30
115. Huang G., Liu S., van der Maaten L., Weinberger K. (2018) CondenseNet: An efficient densenet using learned group convolutions. In: CVPR 30
116. 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. In: CVPR 16, 26, 28
117. Huang Z., Huang L., Gong Y., Huang C., Wang X. (2019) Mask scoring rcnn. In: CVPR 27
118. Hubara I., Courbariaux M., Soudry D., ElYaniv R., Bengio Y. (2016) Binarized neural networks. In: NIPS, pp. 4107–4115 30
119. Iandola F., Han S., Moskewicz M., Ashraf K., Dally W., Keutzer K. (2016) SqueezeNet: Alexnet level accuracy with 50x fewer parameters and 0.5 mb model size. In: arXiv preprint arXiv:1602.07360 29
120. ILSVRC detection challenge results (2018) http://www.image-net.org/challenges/LSVRC/ 28
121. 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 15, 16, 27
122. Jaderberg M., Simonyan K., Zisserman A., et al. (2015) Spatial transformer networks. In: NIPS, pp. 2017–2025 21
123. Jia Y., Shelhamer E., Donahue J., Karayev S., Long J., Girshick R., Guadarrama S., Darrell T. (2014) Caffe: Convolutional architecture for fast feature embedding. In: ACM MM, pp. 675–678 17
124. Jiang B., Luo R., Mao J., Xiao T., Jiang Y. (2018) Acquisition of localization confidence for accurate object detection. In: ECCV, pp. 784–799 27
125. Kang B., Liu Z., Wang X., Yu F., Feng J., Darrell T. (2018) Few shot object detection via feature reweighting. arXiv preprint arXiv:181201866 30
126. Kang K., Ouyang W., Li H., Wang X. (2016) Object detection from video tubelets with convolutional neural networks. In: CVPR, pp. 817–825 31
127. Kim A., Sharma A., Jacobs D. (2014) Locally scale invariant convolutional neural networks. In: NIPS 21
128. Kim K., Hong S., Roh B., Cheon Y., Park M. (2016) PVANet: Deep but lightweight neural networks for real time object detection. In: NIPSW 18
129. Kim Y., Kang B.-N., Kim D. (2018) SAN: learning relationship between convolutional features for multiscale object detection. In: ECCV, pp. 316–331 19
130. Kirillov A., He K., Girshick R., Rother C., Dollár P. (2018) Panoptic segmentation. arXiv:180100868 29
131. Kong T., Yao A., Chen Y., Sun F. (2016) HyperNet: towards accurate region proposal generation and joint object detection. In: CVPR, pp. 845–853 17, 18, 19, 24
132. 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 17, 18, 19, 20
133. Kong T., Sun F., Tan C., Liu H., Huang W. (2018) Deep feature pyramid reconfiguration for object detection. In: ECCV, pp. 169–185 18, 19, 20
134. Krähenbühl1 P., Koltun V. (2014) Geodesic object proposals. In: ECCV 24
135. 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 27
136. Krizhevsky A., Sutskever I., Hinton G. (2012) ImageNet classification with deep convolutional neural networks. In: NIPS, pp. 1097–1105 2, 3, 5, 6, 10, 13, 14, 24, 28
137. Krizhevsky A., Sutskever I., Hinton G. (2012) ImageNet classification with deep convolutional neural networks. In: NIPS, pp. 1097–1105 15, 16, 30
138. Kuo W., Hariharan B., Malik J. (2015) DeepBox: Learning objectness with convolutional networks. In: ICCV, pp. 2479–2487 24, 25
139. Kuznetsova A., Rom H., Alldrin N., Uijlings J., Krasin I., PontTuset J., Kamali S., Popov S., Malloci M., Duerig T., et al. (2018) The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. arXiv preprint arXiv:181100982 7, 8, 9, 10, 28
140. Lake B., Salakhutdinov R., Tenenbaum J. (2015) Human level concept learning through probabilistic program induction. Science 350(6266):1332–1338 30
141. Lampert C. H., Blaschko M. B., Hofmann T. (2008) Beyond sliding windows: Object localization by efficient subwindow search. In: CVPR, pp. 1–8 9
142. Law H., Deng J. (2018) CornerNet: Detecting objects as paired keypoints. In: ECCV 15, 17, 28
143. 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 3, 5, 11
144. 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
145. LeCun Y., Bengio Y., Hinton G. (2015) Deep learning. Nature 521:436–444 1, 2, 3, 6, 7, 15
146. Lee C., Xie S., Gallagher P., Zhang Z., Tu Z. (2015) Deeply supervised nets. In: Artificial Intelligence and Statistics, pp. 562–570 16
147. Lenc K., Vedaldi A. (2015) R-CNN minus R. In: BMVC15 13, 37
148. Lenc K., Vedaldi A. (2018) Understanding image representations by measuring their equivariance and equivalence. IJCV 21
149. Li B., Liu Y., Wang X. (2019) Gradient harmonized single stage detector. In: AAAI 27
150. 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
151. Li H., Kadav A., Durdanovic I., Samet H., Graf H. P. (2017) Pruning filters for efficient convnets. In: ICLR 30
152. Li H., Liu Y., Ouyang W., XiaogangWang (2018) Zoom out and in network with map attention decision for region proposal and object detection. IJCV 18, 19, 20, 24, 25

153. 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 22, 23
154. Li Q., Jin S., Yan J. (2017) Mimicking very efficient network for object detection. In: CVPR, pp. 7341–7349 30
155. Li S. Z., Zhang Z. (2004) Floatboost learning and statistical face detection. IEEE TPAMI 26(9):1112–1123 13
156. 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
157. Li Y., Ouyang W., Zhou B., Wang K., Wang X. (2017) Scene graph generation from objects, phrases and region captions. In: ICCV, pp. 1261–1270 29
158. Li Y., Qi H., Dai J., Ji X., Wei Y. (2017) Fully convolutional instance aware semantic segmentation. In: CVPR, pp. 4438–4446 25
159. Li Y., Chen Y., Wang N., Zhang Z. (2019) Scale aware trident networks for object detection. arXiv preprint arXiv:190101892 19, 29
160. Li Z., Peng C., Yu G., Zhang X., Deng Y., Sun J. (2018) DetNet: A backbone network for object detection. In: ECCV 17, 18, 19, 20, 29
161. 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 13, 27
162. 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 3, 4, 5, 6, 7, 8, 24, 27, 28, 30
163. Lin T., Dollár P., Girshick R., He K., Hariharan B., Belongie S. (2017) Feature pyramid networks for object detection. In: CVPR 13, 17, 18, 19, 20, 27, 28, 29
164. Lin T., Goyal P., Girshick R., He K., Dollár P. (2017) Focal loss for dense object detection. In: ICCV 15, 19, 26, 27
165. Lin X., Zhao C., Pan W. (2017) Towards accurate binary convolutional neural network. In: NIPS, pp. 344–352 30
166. 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, 6
167. Liu C., Zoph B., Neumann M., Shlens J., Hua W., Li L., FeiFei L., Yuille A., Huang J., Murphy K. (2018) Progressive neural architecture search. In: ECCV, pp. 19–34 30
168. 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 21
169. Liu S., Huang D., Wang Y. (2018) Receptive field block net for accurate and fast object detection. In: ECCV 17, 18, 19
170. Liu S., Qi L., Qin H., Shi J., Jia J. (2018) Path aggregation network for instance segmentation. In: CVPR, pp. 8759–8768 18, 19, 20
171. 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 14, 15, 17, 19, 23, 25, 26, 27, 28, 29, 30, 37
172. 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 22, 23
173. Long J., Shelhamer E., Darrell T. (2015) Fully convolutional networks for semantic segmentation. In: CVPR, pp. 3431–3440 12, 13, 17, 19, 24, 25
174. Lowe D. (1999) Object recognition from local scale invariant features. In: ICCV, vol 2, pp. 1150–1157 3, 5, 15
175. Lowe D. (2004) Distinctive image features from scale-invariant keypoints. IJCV 60(2):91–110 3, 5, 23
176. Loy C., Lin D., Ouyang W., Xiong Y., Yang S., Huang Q., Zhou D., Xia W., Li Q., Luo P., et al. (2019) WIDER face and pedestrian challenge 2018: Methods and results. arXiv:190206854 29
177. Lu Y., Javidi T., Lazebnik S. (2016) Adaptive object detection using adjacency and zoom prediction. In: CVPR, pp. 2351–2359 24, 25
178. Luo P., Wang X., Shao W., Peng Z. (2018) Towards understanding regularization in batch normalization. In: ICLR 29
179. Luo P., Ren J., Peng Z., Zhang R., Li J. (2019) Switchable normalization for learning to normalize deep representation. IEEE TPAMI 29
180. Ma J., Shao W., Ye H., Wang L., Wang H., Zheng Y., Xue X. (2018) Arbitrary oriented scene text detection via rotation proposals. IEEE TMM 20(11):3111–3122 21
181. Malisiewicz T., Efros A. (2009) Beyond categories: The visual memex model for reasoning about object relationships. In: NIPS 23, 29
182. Manen S., Guillaumin M., Van Gool L. (2013) Prime object proposals with randomized prim's algorithm. In: CVPR, pp. 2536–2543 24
183. Mikolajczyk K., Schmid C. (2005) A performance evaluation of local descriptors. IEEE TPAMI 27(10):1615–1630 5
184. 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 18, 21
185. MS COCO detection leaderboard (2018) http://cocodataset.org/#detection-leaderboard 28
186. 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
187. Murase H., Nayar S. (1995) Visual learning and recognition of 3D objects from appearance. IJCV 14(1):5–24 5
188. Murase H., Nayar S. (1995) Visual learning and recognition of 3d objects from appearance. IJCV 14(1):5–24 5
189. Murphy K., Torralba A., Freeman W. (2003) Using the forest to see the trees: a graphical model relating features, objects and scenes. In: NIPS 23, 29
190. Newell A., Yang K., Deng J. (2016) Stacked hourglass networks for human pose estimation. In: ECCV, pp. 483–499 15, 19
191. Newell A., Huang Z., Deng J. (2017) Associative embedding: end to end learning for joint detection and grouping. In: NIPS, pp. 2277–2287 15
192. 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, 23
193. Oliva A., Torralba A. (2007) The role of context in object recognition. Trends in cognitive sciences 11(12):520–527 22, 29
194. Opelt A., Pinz A., Fussenegger M., Auer P. (2006) Generic object recognition with boosting. IEEE TPAMI 28(3):416–431 4
195. Oquab M., Bottou L., Laptev I., Sivic J. (2014) Learning and transferring midlevel image representations using convolutional neural networks. In: CVPR, pp. 1717–1724 7
196. 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 12
197. Osuna E., Freund R., Girosit F. (1997) Training support vector machines: an application to face detection. In: CVPR, pp. 130–136 5
198. Ouyang W., Wang X. (2013) Joint deep learning for pedestrian detection. In: ICCV, pp. 2056–2063 21
199. 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 9, 18, 21, 22, 23, 24, 27, 29
200. Ouyang W., Wang X., Zhang C., Yang X. (2016) Factors in finetuning deep model for object detection with long tail distribution. In: CVPR, pp. 864–873 26
201. Ouyang W., Wang K., Zhu X., Wang X. (2017) Chained cascade network for object detection. ICCV 13
202. 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) DeepID-Net: Object detection with deformable part based convolutional neural networks. IEEE TPAMI 39(7):1320–1334 17, 21
203. 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 23
204. PASCAL VOC detection leaderboard (2018) http://host.robots.ox.ac.uk:8080/leaderboard/main_bootstrap.php 28
205. 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 26, 27, 28
206. Peng X., Sun B., Ali K., Saenko K. (2015) Learning deep object detectors from 3d models. In: ICCV, pp. 1278–1286 26
207. 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 31
208. Perronnin F., Sánchez J., Mensink T. (2010) Improving the fisher kernel for large scale image classification. In: ECCV, pp. 143–156 3, 5, 15
209. Pinheiro P., Collobert R., Dollar P. (2015) Learning to segment object candidates. In: NIPS, pp. 1990–1998 24, 25
210. Pinheiro P., Lin T., Collobert R., Dollár P. (2016) Learning to refine object segments. In: ECCV, pp. 75–91 18, 19, 25

211. Ponce J., Hebert M., Schmid C., Zisserman A. (2007) Toward Category Level Object Recognition. Springer 3, 5
212. Pouyanfar S., Sadiq S., Yan Y., Tian H., Tao Y., Reyes M. P., Shyu M., Chen S., Iyengar S. (2018) A survey on deep learning: Algorithms, techniques, and applications. ACM Computing Surveys 51(5):92:1–92:36 6
213. Qi C. R., Su H., Mo K., Guibas L. J. (2017) PointNet: Deep learning on point sets for 3D classification and segmentation. In: CVPR, pp. 652–660 31
214. Qi C. R., Liu W., Wu C., Su H., Guibas L. J. (2018) Frustum pointnets for 3D object detection from RGBD data. In: CVPR, pp. 918–927 31
215. Quanming Y., Mengshuo W., Hugo J. E., Isabelle G., Yiqi H., Yufeng L., Weiwei T., Qiang Y., Yang Y. (2018) Taking human out of learning applications: A survey on automated machine learning. arXiv:181013306 30
216. Rabinovich A., Vedaldi A., Galleguillos C., Wiewiora E., Belongie S. (2007) Objects in context. In: ICCV 23, 29
217. Rahman S., Khan S., Barnes N. (2018) Polarity loss for zero shot object detection. arXiv preprint arXiv:181108982 31
218. Rahman S., Khan S., Porikli F. (2018) Zero shot object detection: Learning to simultaneously recognize and localize novel concepts. In: ACCV 31
219. 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 17
220. Rebuffi S., Bilen H., Vedaldi A. (2017) Learning multiple visual domains with residual adapters. In: Advances in Neural Information Processing Systems, pp. 506–516 31
221. Rebuffi S., Bilen H., Vedaldi A. (2018) Efficient parametrization of multidomain deep neural networks. In: CVPR, pp. 8119–8127 31
222. Redmon J., Farhadi A. (2017) YOLO9000: Better, faster, stronger. In: CVPR 14, 16, 37
223. Redmon J., Divvala S., Girshick R., Farhadi A. (2016) You only look once: Unified, real time object detection. In: CVPR, pp. 779–788 14, 15, 16, 17, 27, 28, 30, 37
224. Ren M., Triantafillou E., Ravi S., Snell J., Swersky K., Tenenbaum J. B., Larochelle H., Zemel R. S. (2018) Meta learning for semisupervised few shot classification. In: ICLR 30
225. 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 9, 12, 13, 15, 17, 23, 24, 25, 27, 28, 30, 37
226. 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, 12, 13, 25, 28
227. Ren S., He K., Girshick R., Zhang X., Sun J. (2017) Object detection networks on convolutional feature maps. IEEE TPAMI 27
228. Rezatofighi H., Tsoi N., Gwak J., Sadeghian A., Reid I., Savarese S. (2019) Generalized intersection over union: A metric and a loss for bounding box regression. In: CVPR 27
229. Rowley H., Baluja S., Kanade T. (1998) Neural network based face detection. IEEE TPAMI 20(1):23–38 5
230. 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, 3, 4, 5, 6, 7, 8, 9, 17, 24, 27, 28, 30
231. 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
232. Schmid C., Mohr R. (1997) Local grayvalue invariants for image retrieval. IEEE TPAMI 19(5):530–535 5
233. Schwartz E., Karlinsky L., Shtok J., Harary S., Marder M., Pankanti S., Feris R., Kumar A., Giries R., Bronstein A. (2019) RepMet: Representative based metric learning for classification and one shot object detection. In: CVPR 30
234. Sermanet P., Kavukcuoglu K., Chintala S., LeCun Y. (2013) Pedestrian detection with unsupervised multistage feature learning. In: CVPR, pp. 3626–3633 5, 23
235. 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, 3, 10, 13, 14, 16, 25, 37
236. 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 18
237. Shelhamer E., Long J., Darrell T. (2017) Fully convolutional networks for semantic segmentation. IEEE TPAMI 12, 13, 17, 19
238. Shen Z., Liu Z., Li J., Jiang Y., Chen Y., Xue X. (2017) DSOD: Learning deeply supervised object detectors from scratch. In: ICCV 17, 18, 19
239. Shi X., Shan S., Kan M., Wu S., Chen X. (2018) Real time rotation invariant face detection with progressive calibration networks. In: CVPR 21
240. 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 30
241. Shrivastava A., Gupta A. (2016) Contextual priming and feedback for Faster RCNN. In: ECCV, pp. 330–348 22, 23
242. Shrivastava A., Gupta A., Girshick R. (2016) Training region based object detectors with online hard example mining. In: CVPR, pp. 761–769 26, 27
243. Shrivastava A., Sukthankar R., Malik J., Gupta A. (2017) Beyond skip connections: Top down modulation for object detection. In: CVPR 17, 18, 19, 20, 27
244. Simonyan K., Zisserman A. (2015) Very deep convolutional networks for large scale image recognition. In: ICLR 3, 6, 11, 12, 15, 16, 27
245. Singh B., Davis L. (2018) An analysis of scale invariance in object detection-SNIP. In: CVPR 8, 26, 27
246. Singh B., Najibi M., Davis L. S. (2018) SNIPER: Efficient multiscale training. arXiv:180509300 26, 27
247. 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 3, 5, 15
248. Song Han W. J. D. Huizi Mao (2016) Deep Compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In: ICLR 30
249. Sun C., Shrivastava A., Singh S., Gupta A. (2017) Revisiting unreasonable effectiveness of data in deep learning era. In: ICCV, pp. 843–852 17
250. Sun S., Pang J., Shi J., Yi S., Ouyang W. (2018) FishNet: A versatile backbone for image, region, and pixel level prediction. In: NIPS, pp. 754–764 17
251. Sun Z., Bebis G., Miller R. (2006) On road vehicle detection: A review. IEEE TPAMI 28(5):694–711 2, 3
252. Sung K., , Poggio T. (1994) Learning and example selection for object and pattern detection. MIT AI Memo (1521) 27
253. Swain M., Ballard D. (1991) Color indexing. IJCV 7(1):11–32 5
254. Szegedy C., Toshev A., Erhan D. (2013) Deep neural networks for object detection. In: NIPS, pp. 2553–2561 10, 13
255. Szegedy C., Reed S., Erhan D., Anguelov D., Ioffe S. (2014) Scalable, high quality object detection. In: arXiv preprint arXiv:1412.1441 24
256. 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 3, 14, 15, 16, 18, 19, 27
257. Szegedy C., Vanhoucke V., Ioffe S., Shlens J., Wojna Z. (2016) Rethinking the inception architecture for computer vision. In: CVPR, pp. 2818–2826 15, 16, 27
258. 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 15, 16, 29
259. Torralba A. (2003) Contextual priming for object detection. IJCV 53(2):169–191 22
260. Turk M. A., Pentland A. (1991) Face recognition using eigenfaces. In: CVPR, pp. 586–591 5
261. Tuzel O., Porikli F., Meer P. (2006) Region covariance: A fast descriptor for detection and classification. In: ECCV, pp. 589–600 5
262. TychsenSmith L., Petersson L. (2017) DeNet: scalable real time object detection with directed sparse sampling. In: ICCV 15, 25
263. TychsenSmith L., Petersson L. (2018) Improving object localization with fitness nms and bounded iou loss. In: CVPR 27
264. Uijlings J., van de Sande K., Gevers T., Smeulders A. (2013) Selective search for object recognition. IJCV 104(2):154–171 3, 9, 10, 24
265. Vaillant R., Monrocq C., LeCun Y. (1994) Original approach for the localisation of objects in images. IEE Proceedings Vision, Image and Signal Processing 141(4):245–250 5
266. Van de Sande K., Uijlings J., Gevers T., Smeulders A. (2011) Segmentation as selective search for object recognition. In: ICCV, pp. 1879–1886 24

267. 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 23
268. Vedaldi A., Gulshan V., Varma M., Zisserman A. (2009) Multiple kernels for object detection. In: ICCV, pp. 606–613 9, 23
269. Viola P., Jones M. (2001) Rapid object detection using a boosted cascade of simple features. In: CVPR, vol 1, pp. 1–8 3, 5, 9, 23
270. 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 21, 29
271. Wang H., Wang Q., Gao M., Li P., Zuo W. (2018) Multiscale location aware kernel representation for object detection. In: CVPR 19
272. 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 3
273. Wang X., Shrivastava A., Gupta A. (2017) A Fast RCNN: Hard positive generation via adversary for object detection. In: CVPR 21, 26
274. Wang X., Cai Z., Gao D., Vasconcelos N. (2019) Towards universal object detection by domain attention. arXiv:190404402 31
275. Wei Y., Pan X., Qin H., Ouyang W., Yan J. (2018) Quantization mimic: Towards very tiny CNN for object detection. In: ECCV, pp. 267–283 30
276. Woo S., Hwang S., Kweon I. (2018) StairNet: Top down semantic aggregation for accurate one shot detection. In: WACV, pp. 1093–1102 19
277. Worrall D. E., Garbin S. J., Turmukhambetov D., Brostow G. J. (2017) Harmonic networks: Deep translation and rotation equivariance. In: CVPR, vol 2 21
278. Wu Y., He K. (2018) Group normalization. In: ECCV, pp. 3–19 29
279. 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 31
280. Wu Z., Pan S., Chen F., Long G., Zhang C., Yu P. S. (2019) A comprehensive survey on graph neural networks. arXiv preprint arXiv:190100596 6
281. Xia G., Bai X., Ding J., Zhu Z., Belongie S., Luo J., Datcu M., Pelillo M., Zhang L. (2018) DOTA: a large-scale dataset for object detection in aerial images. In: CVPR, pp. 3974–3983 21
282. Xiang Y., Mottaghi R., Savarese S. (2014) Beyond PASCAL: A benchmark for 3D object detection in the wild. In: WACV, pp. 75–82 31
283. Xiao R., Zhu L., Zhang H. (2003) Boosting chain learning for object detection. In: ICCV, pp. 709–715 5
284. Xie S., Girshick R., Dollár P., Tu Z., He K. (2017) Aggregated residual transformations for deep neural networks. In: CVPR 13, 16, 27, 29
285. Yang B., Yan J., Lei Z., Li S. (2016) CRAFT objects from images. In: CVPR, pp. 6043–6051 22, 24, 25, 27
286. 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 18
287. Yang M., Kriegman D., Ahuja N. (2002) Detecting faces in images: A survey. IEEE TPAMI 24(1):34–58 2, 3
288. Ye Q., Doermann D. (2015) Text detection and recognition in imagery: A survey. IEEE TPAMI 37(7):1480–1500 2, 3
289. Yosinski J., Clune J., Bengio Y., Lipson H. (2014) How transferable are features in deep neural networks? In: NIPS, pp. 3320–3328 17
290. Young T., Hazarika D., Poria S., Cambria E. (2018) Recent trends in deep learning based natural language processing. IEEE Computational Intelligence Magazine 13(3):55–75 6
291. Yu F., Koltun V. (2016) Multiscale context aggregation by dilated convolutions 17
292. Yu F., Koltun V., Funkhouser T. (2017) Dilated residual networks. In: CVPR, vol 2, p. 3 17
293. Yu R., Li A., Chen C., Lai J., et al. (2018) NISP: Pruning networks using neuron importance score propagation. CVPR 30
294. 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
295. Zagoruyko S., Lerer A., Lin T., Pinheiro P., Gross S., Chintala S., Dollár P. (2016) A multipath network for object detection. In: BMVC 18, 23, 25
296. Zeiler M., Fergus R. (2014) Visualizing and understanding convolutional networks. In: ECCV, pp. 818–833 7, 15, 16, 23
297. Zeng X., Ouyang W., Yang B., Yan J., Wang X. (2016) Gated bidirectional cnn for object detection. In: ECCV, pp. 354–369 22, 23, 29
298. 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 22, 23, 24, 29
299. 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
300. Zhang L., Lin L., Liang X., He K. (2016) Is faster RCNN doing well for pedestrian detection? In: ECCV, pp. 443–457 2
301. Zhang S., Wen L., Bian X., Lei Z., Li S. (2018) Single shot refinement neural network for object detection. In: CVPR 17, 18, 19, 20
302. 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, 23
303. Zhang X., Li Z., Change Loy C., Lin D. (2017) PolyNet: a pursuit of structural diversity in very deep networks. In: CVPR, pp. 718–726 22
304. Zhang X., Zhou X., Lin M., Sun J. (2018) ShuffleNet: an extremely efficient convolutional neural network for mobile devices. In: CVPR 29
305. Zhang Z., Geiger J., Pohjalainen J., Mousa A. E., Jin W., Schuller B. (2018) Deep learning for environmentally robust speech recognition: An overview of recent developments. ACM Trans Intell Syst Technol 9(5):49:1–49:28 6
306. Zhang Z., Qiao S., Xie C., Shen W., Wang B., Yuille A. (2018) Single shot object detection with enriched semantics. In: CVPR 17
307. Zhao Q., Sheng T., Wang Y., Tang Z., Chen Y., Cai L., Ling H. (2019) M2Det: A single shot object detector based on multilevel feature pyramid network. In: AAAI 18, 19, 20
308. 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 23
309. Zhou B., Khosla A., Lapedriza A., Oliva A., Torralba A. (2015) Object detectors emerge in deep scene CNNs. In: ICLR 12, 17
310. Zhou B., Khosla A., Lapedriza A., Oliva A., Torralba A. (2016) Learning deep features for discriminative localization. In: CVPR, pp. 2921–2929 12
311. 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 8, 17, 27
312. Zhou J., Cui G., Zhang Z., Yang C., Liu Z., Sun M. (2018) Graph neural networks: A review of methods and applications. arXiv preprint arXiv:181208434 6
313. Zhou P., Ni B., Geng C., Hu J., Xu Y. (2018) Scale transferrable object detection. In: CVPR 16, 17, 18, 19
314. 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
315. Zhou Y., Ye Q., Qiu Q., Jiao J. (2017) Oriented response networks. In: CVPR, pp. 4961–4970 21
316. Zhu X., Vondrick C., Fowlkes C., Ramanan D. (2016) Do we need more training data? IJCV 119(1):76–92 15
317. Zhu X., Tuia D., Mou L., Xia G., Zhang L., Xu F., Fraundorfer F. (2017) Deep learning in remote sensing: A comprehensive review and list of resources. IEEE Geoscience and Remote Sensing Magazine 5(4):8–36 6
318. 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 22, 23
319. 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 22, 23
320. Zhu Y., Zhou Y., Ye Q., Qiu Q., Jiao J. (2017) Soft proposal networks for weakly supervised object localization. In: ICCV, pp. 1841–1850 24
321. 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
322. Zitnick C., Dollár P. (2014) Edge boxes: Locating object proposals from edges. In: ECCV, pp. 391–405 24
323. Zoph B., Le Q. (2017) Neural architecture search with reinforcement learning 30
324. Zoph B., Vasudevan V., Shlens J., Le Q. (2018) Learning transferable architectures for scalable image recognition. In: CVPR, pp. 8697–8710 30

**Table 11** Summarization of properties and performance of milestone detection frameworks for generic object detection. See Section 5 for detail discussion. The architectures of some methods listed in this table are illustrated in Fig. 13. The properties of the backbone DCNNs can be found in Table 6.

| | Detector Name | RP | Backbone DCNN | Input ImgSize | VOC07 Results | VOC12 Results | Speed (FPS) | Published In | Source Code | Highlights and Disadvantages |
|---|---|---|---|---|---|---|---|---|---|---|
| **Region based** (Section 5.1) | RCNN [83] | 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 [96] | 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 [82] | 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 [225] | 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 [147] | 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 [48] | 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 [99] | 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 5.2) | OverFeat [235] | – | 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 [223] | – | 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[222] | – | 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 [171] | – | 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.*