International Journal of Computer Vision
https://doi.org/10.1007/s11263-019-01247-4

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: 6 September 2018 / Accepted: 26 September 2019

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 about 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 (illustrated in Fig. 1) has been an active area of research for several decades (Fischler and Elschlager 1973). The goal of object detection is to determine whether there are any instances of objects from given categories (such as humans, cars, bicycles, dogs or cats) in an image and, if present, to return the spatial location and extent of each object instance (e.g., via a bounding box Everingham et al. 2010; Russakovsky et al. 2015). 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 supports 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.

Recently, deep learning techniques (Hinton and Salakhutdinov 2006; LeCun et al. 2015) have emerged as powerful methods for learning feature representations automatically from data. In particular, these techniques have provided major improvements in object detection, as illustrated in Fig. 3.

As illustrated in Fig. 2, object detection can be grouped into one of two types (Grauman and Leibe 2011; Zhang et al. 2013): 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.

Communicated by Bernt Schiele.

✉ 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, Changsha, China

2 University of Oulu, Oulu, Finland

3 University of Sydney, Camperdown, Australia

4 Chinese University of Hong Kong, Sha Tin, China

5 University of Waterloo, Waterloo, Canada

Published online: 31 October 2019

Fig. 1 Most frequent keywords in ICCV and CVPR conference papers from 2016 to 2018. 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

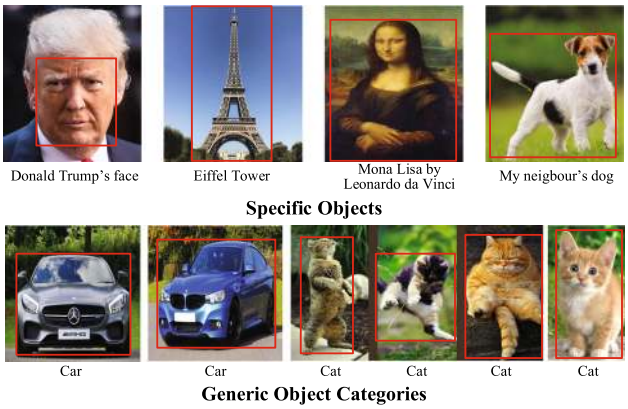

Fig. 2 Object detection includes localizing instances of a *particular* object (top), as well as generalizing to detecting object *categories* in general (bottom). This survey focuses on recent advances for the latter problem of generic object detection

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, over the past several years, the research community has started moving towards the more challenging goal of building general purpose object detection systems where the breadth of object detection ability rivals that of humans.

Krizhevsky et al. (2012a) proposed a Deep Convolutional Neural Network (DCNN) called AlexNet which achieved record breaking image classification accuracy in the Large Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al. 2015). Since that time, the research focus in most aspects of computer vision has been specifically on deep learning methods, indeed including the domain of generic object detection (Girshick et al. 2014; He et al. 2014; Girshick 2015; Sermanet et al. 2014; Ren et al. 2017). Although tremendous progress has been achieved, illustrated in Fig. 3, we are unaware of comprehensive surveys of this subject

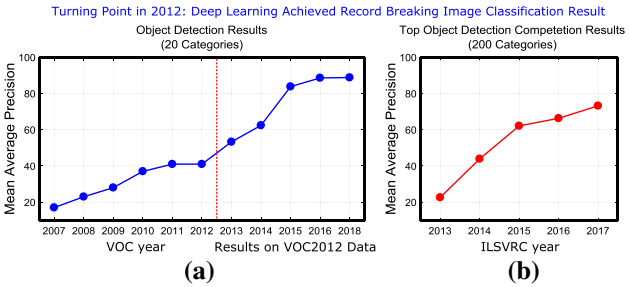

Fig. 3 An overview of recent object detection performance: we can observe a significant improvement in performance (measured as mean average precision) since the arrival of deep learning in 2012. **a** Detection results of winning entries in the VOC2007-2012 competitions, and **b** top object detection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

over the past 5 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 (Enzweiler and Gavrila 2009; Geronimo et al. 2010; Dollar et al. 2012), face detection (Yang et al. 2002; Zafeiriou et al. 2015), vehicle detection (Sun et al. 2006) and text detection (Ye and Doermann 2015). There are comparatively few recent surveys focusing directly on the problem of generic object detection, except for the work by Zhang et al. (2013) who conducted a survey on the topic of object class detection. However, the research reviewed in Grauman and Leibe (2011), Andreopoulos and Tsotsos (2013) and Zhang et al. (2013) is mostly pre-2012, and therefore prior to 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 (LeCun et al. 1998, 2015; Krizhevsky et al. 2012a) 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. (2013), LeCun et al. (2015), Litjens et al. (2017), Gu et al. (2018), and more recently in tutorials at ICCV and CVPR.

In contrast, although many deep learning based methods have been proposed for object detection, we are unaware of

Table 1 Summary of related object detection surveys since 2000

No.	Survey title	References	Year	Venue	Content
1	Monocular pedestrian detection: survey and experiments	Enzweiler and Gavrila (2009)	2009	PAMI	An evaluation of three pedestrian detectors
2	Survey of pedestrian detection for advanced driver assistance systems	Geronimo et al. (2010)	2010	PAMI	A survey of pedestrian detection for advanced driver assistance systems
3	Pedestrian detection: an evaluation of the state of the art	Dollar et al. (2012)	2012	PAMI	A thorough and detailed evaluation of detectors in monocular images
4	Detecting faces in images: a survey	Yang et al. (2002)	2002	PAMI	First survey of face detection from a single image
5	A survey on face detection in the wild: past, present and future	Zafeiriou et al. (2015)	2015	CVIU	A survey of face detection in the wild since 2000
6	On road vehicle detection: a review	Sun et al. (2006)	2006	PAMI	A review of vision based on-road vehicle detection systems
7	Text detection and recognition in imagery: a survey	Ye and Doermann (2015)	2015	PAMI	A survey of text detection and recognition in color imagery
8	Toward category level object recognition	Ponce et al. (2007)	2007	Book	Representative papers on object categorization, detection, and segmentation
9	The evolution of object categorization and the challenge of image abstraction	Dickinson et al. (2009)	2009	Book	A trace of the evolution of object categorization over 4 decades
10	Context based object categorization: a critical survey	Galleguillos and Belongie (2010)	2010	CVIU	A review of contextual information for object categorization
11	50 years of object recognition: directions forward	Andreopoulos and Tsotsos (2013)	2013	CVIU	A review of the evolution of object recognition systems over 5 decades
12	Visual object recognition	Grauman and Leibe (2011)	2011	Tutorial	Instance and category object recognition techniques
13	Object class detection: a survey	Zhang et al. (2013)	2013	ACM CS	Survey of generic object detection methods before 2011
14	Feature representation for statistical learning based object detection: a review	Li et al. (2015b)	2015	PR	Feature representation methods in statistical learning based object detection, including handcrafted and deep learning based features
15	Salient object detection: a survey	Borji et al. (2014)	2014	arXiv	A survey for salient object detection
16	Representation learning: a review and new perspectives	Bengio et al. (2013)	2013	PAMI	Unsupervised feature learning and deep learning, probabilistic models, autoencoders, manifold learning, and deep networks
17	Deep learning	LeCun et al. (2015)	2015	Nature	An introduction to deep learning and applications
18	A survey on deep learning in medical image analysis	Litjens et al. (2017)	2017	MIA	A survey of deep learning for image classification, object detection, segmentation and registration in medical image analysis
19	Recent advances in convolutional neural networks	Gu et al. (2018)	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

Table 1 continued

No.	Survey title	References	Year	Venue	Content
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	A tutorial on methods and principles behind image classification, object detection, instance segmentation, and semantic segmentation
24	Deep learning for generic object detection	Ours	2019	VISI	A comprehensive survey of deep learning for generic object detection

any 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 (Li et al. 2015a; Zhang et al. 2016a; Hu et al. 2017), pedestrian detection (Zhang et al. 2016b; Hosang et al. 2015), vehicle detection (Zhou et al. 2016b) and traffic sign detection (Zhu et al. 2016b) will not be considered.

1.2 Scope

The number of papers on generic object detection based on deep learning is breathtaking. There are 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, in such a way 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 on related topics, readers are referred to the articles in Table 1. This survey focuses on major progress of the last 5 years, and we restrict our attention to still pictures, leaving the important subject of video object detection as a topic for separate consideration in the future.

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 intention 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 and the progress made during the last 2 decades are summarized in Sect. 2. A brief introduction to deep learning is given in Sect. 3. Popular datasets and evaluation criteria are summarized in Sect. 4. We describe the milestone object detection frameworks in Sect. 5. From Sects. 6 to 9, fundamental sub-problems and the relevant issues involved in designing object detectors are discussed. Finally, in Sect. 10, we conclude the paper with an overall discussion of object detection, state-of-the- art performance, and future research directions.

2 Generic Object Detection

2.1 The Problem

Generic object detection, also called generic object category detection, object class detection, or object category detection (Zhang et al. 2013), is defined as follows. Given an image, 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 artic-

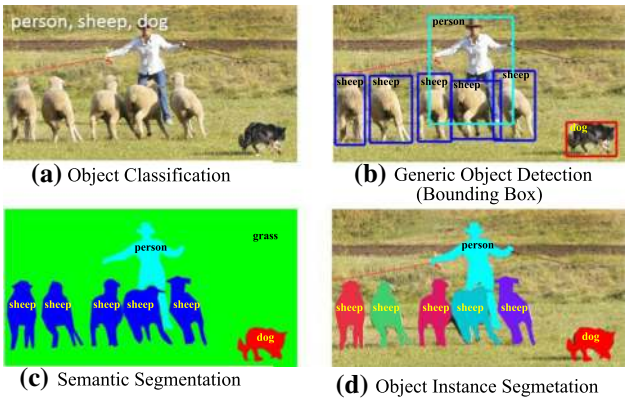

Fig. 4 Recognition problems related to generic object detection: **a** image level object classification, **b** bounding box level generic object detection, **c** pixel-wise semantic segmentation, **d** instance level semantic segmentation

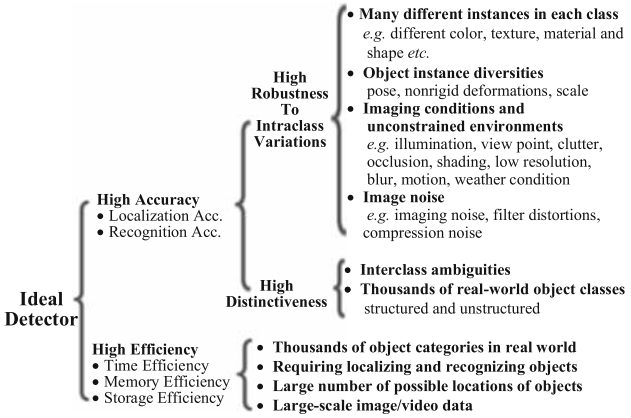

Fig. 5 Taxonomy of challenges in generic object detection

ulated 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) (Everingham et al. 2010; Russakovsky et al. 2015), a precise pixelwise segmentation mask (Zhang et al. 2013), or a closed boundary (Lin et al. 2014; Russell et al. 2008), as illustrated in Fig. 4. To the best of our knowledge, for the evaluation of generic object detection algorithms, it is bounding boxes which are most widely used in the current literature (Everingham et al. 2010; Russakovsky et al. 2015), and therefore this is also 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 pixelwise object segmentation), it is anticipated that future challenges will be at the pixel level (Lin et al. 2014).

There are many problems closely related to that of generic object detection[1]. The goal of *object classification* or *object categorization* (Fig. 4a) is to assess the presence of objects from a given set of object classes in an image; i.e., assigning one or more object class labels to a given image, determining the 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 (Everingham et al. 2010; Russakovsky et al. 2015; Opelt et al. 2006; Andreopoulos and Tsotsos 2013). Generic object detection is closely related to *semantic image segmentation* (Fig. 4c), which aims to assign each pixel in an image to a semantic class label. *Object instance segmentation* (Fig. 4d) aims to distinguish different instances of the same object class, as opposed to semantic segmentation which does not.

2.2 Main Challenges

The ideal of generic object detection is to develop a general-purpose algorithm that achieves two competing goals of *high quality/accuracy* and *high efficiency* (Fig. 5). As illustrated in Fig. 6, high quality detection must 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 intra-class appearance variations, can be localized and recognized (i.e., high robustness). High efficiency requires that the entire detection task runs in real time with acceptable memory and storage demands.

2.2.1 Accuracy Related Challenges

Challenges in detection accuracy stem from (1) the vast range of intra-class variations and (2) the huge number of object categories.

Intra-class variations 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. 6i. Even in a more narrowly defined class, such as human or horse, object instances can appear in different poses, subject to nonrigid deformations or with the addition of clothing.

[1] To the best of our knowledge, there is no universal agreement in the literature on the definitions of various vision subtasks. Terms such as detection, localization, recognition, classification, categorization, verification, identification, annotation, labeling, and understanding are often differently defined (Andreopoulos and Tsotsos 2013).

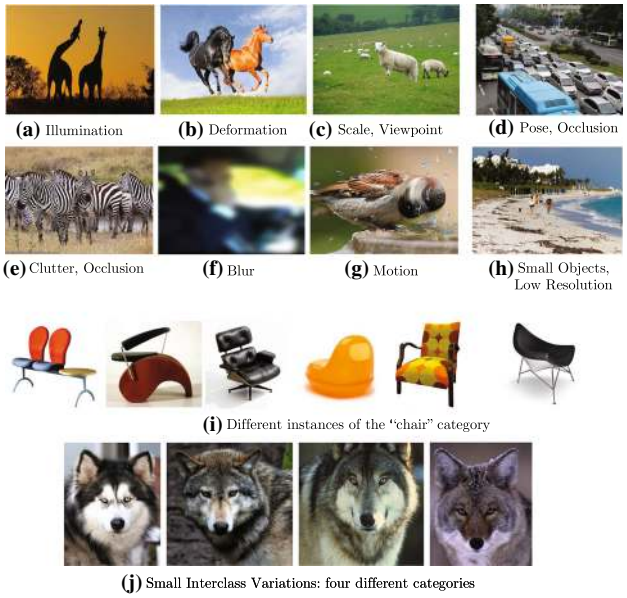

Fig. 6 Changes in appearance of the same class with variations in imaging conditions (**a**–**h**). There is an astonishing variation in what is meant to be a single object class (**i**). In contrast, the four images in **j** appear very similar, but in fact are from four different object classes. Most images are from ImageNet (Russakovsky et al. 2015) and MS COCO (Lin et al. 2014)

Imaging condition variations are caused by the dramatic impacts unconstrained environments can have on object appearance, such as lighting (dawn, day, dusk, indoors), physical location, weather conditions, cameras, backgrounds, illuminations, occlusion, 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. 6a–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 from the detector to distinguish between subtly different *inter*class variations, as illustrated in Fig. 6j. In practice, current detectors focus mainly on structured object categories, such as the 20, 200 and 91 object classes in PASCAL VOC (Everingham et al. 2010), ILSVRC (Russakovsky et al. 2015) and MS COCO (Lin et al. 2014) respectively. Clearly, the number of object categories under consideration in existing benchmark datasets is much smaller than 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, making efficient object detection critical.

The efficiency challenges stem from the need to localize and recognize, computational complexity growing with the (possibly large) number of object categories, and with the (possibly very large) number of locations and scales within a single image, such as the examples in Fig. 6c, d.

A further challenge is that of scalability: A detector should be able to handle previously unseen objects, unknown situ-

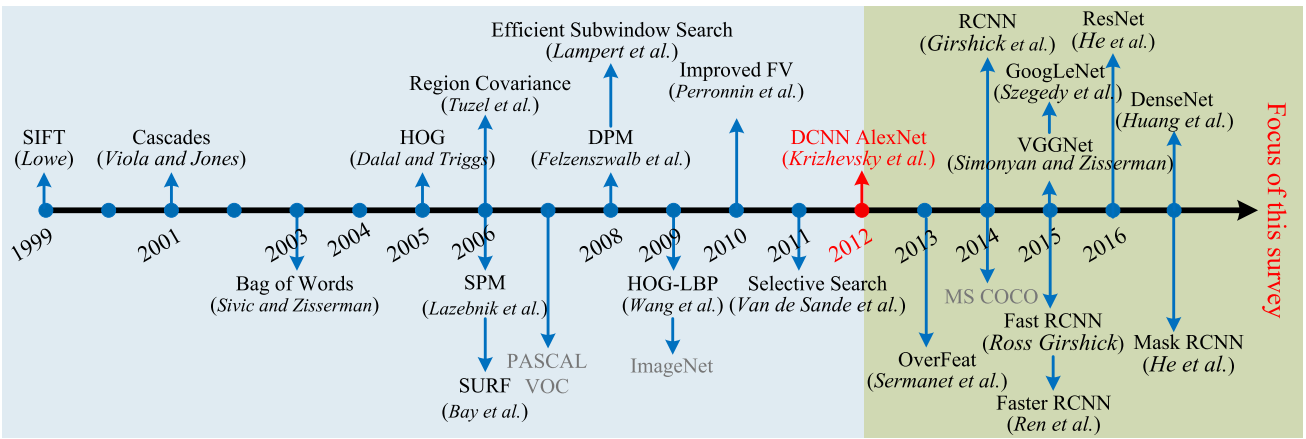

Fig. 7 Milestones of object detection and recognition, including feature representations (Csurka et al. 2004; Dalal and Triggs 2005; He et al. 2016; Krizhevsky et al. 2012a; Lazebnik et al. 2006; Lowe 1999, 2004; Perronnin et al. 2010; Simonyan and Zisserman 2015; Sivic and Zisserman 2003; Szegedy et al. 2015; Viola and Jones 2001; Wang et al. 2009), detection frameworks (Felzenszwalb et al. 2010b; Girshick et al. 2014; Sermanet et al. 2014; Uijlings et al. 2013; Viola and Jones 2001), and datasets (Everingham et al. 2010; Lin et al. 2014; Russakovsky et al. 2015). The time period up to 2012 is dominated by handcrafted features, a transition took place in 2012 with the development of DCNNs for image classification by Krizhevsky et al. (2012a), with methods after 2012 dominated by related deep networks. Most of the listed methods are highly cited and won a major ICCV or CVPR prize. See Sect. 2.3 for details

ations, and high data rates. As the number of images and the number of categories continue to grow, it may become impossible to annotate them manually, forcing a reliance on weakly supervised strategies.

2.3 Progress in the Past 2 Decades

Early research on object recognition was based on template matching techniques and simple part-based models (Fischler and Elschlager 1973), 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 (Mundy 2006; Ponce et al. 2007), with the focus later moving away from geometry and prior models towards the use of statistical classifiers [such as Neural Networks (Rowley et al. 1998), SVM (Osuna et al. 1997) and Adaboost (Viola and Jones 2001; Xiao et al. 2003)] based on appearance features (Murase and Nayar 1995a; Schmid and Mohr 1997). This successful family of object detectors set the stage for most subsequent research in this field.

The milestones of object detection in more recent years are presented in Fig. 7, in which two main eras (SIFT vs. DCNN) are highlighted. The appearance features moved from global representations (Murase and Nayar 1995b; Swain and Ballard 1991; Turk and Pentland 1991) 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 (Lowe 1999), and the progress on various visual recognition tasks was based substantially on the use of local descriptors (Mikolajczyk and Schmid 2005) such as Haar-like features (Viola and Jones 2001), SIFT (Lowe 2004), Shape Contexts (Belongie et al. 2002), Histogram of Gradients (HOG) (Dalal and Triggs 2005) Local Binary Patterns (LBP) (Ojala et al. 2002), and region covariances (Tuzel et al. 2006). These local features are usually aggregated by simple concatenation or feature pooling encoders such as the Bag of Visual Words approach, introduced by Sivic and Zisserman (2003) and Csurka et al. (2004), Spatial Pyramid Matching (SPM) of BoW models (Lazebnik et al. 2006), and Fisher Vectors (Perronnin et al. 2010).

For years, the multistage hand tuned 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 (Krizhevsky et al. 2012a) achieved their record-breaking results in image classification.

The use of CNNs for detection and localization (Rowley et al. 1998) can be traced back to the 1990s, with a modest number of hidden layers used for object detection (Vaillant et al. 1994; Rowley et al. 1998; Sermanet et al. 2013), successful in restricted domains such as face detection. However, more recently, deeper CNNs have led to record-breaking improvements in the detection of more general object categories, a shift which came about when the successful application of DCNNs in image classification (Krizhevsky et al. 2012a) was transferred to object detection, resulting in the milestone Region-based CNN (RCNN) detector of Girshick et al. (2014).

The successes of deep detectors rely heavily on vast training data and large networks with millions or even billions of parameters. The availability of GPUs with very high computational capability and large-scale detection datasets [such as ImageNet (Deng et al. 2009; Russakovsky et al. 2015) and MS COCO (Lin et al. 2014)] play a key role in their success. Large datasets have allowed researchers to target more realistic and complex problems from images with large intra-class variations and inter-class similarities (Lin et al. 2014; Russakovsky et al. 2015). However, accurate annotations are labor intensive to obtain, so detectors must consider methods that can relieve annotation difficulties or can learn with smaller training datasets.

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 3000 entry level categories and 30,000 visual categories overall, and the number of categories distinguishable with domain expertise may be to the order of 10^5 (Biederman 1987a). 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 unresolved 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 tremendously rapid evolution, there exist many recent survey papers on deep learning (Bengio et al. 2013; Goodfellow et al. 2016; Gu et al. 2018; LeCun et al. 2015; Litjens et al. 2017; Pouyanfar et al. 2018; Wu et al. 2019; Young et al. 2018; Zhang et al. 2018d; Zhou et al. 2018a; Zhu et al. 2017). These surveys have reviewed deep learning techniques from different perspectives (Bengio et al. 2013; Goodfellow et al. 2016; Gu et al. 2018; LeCun et al. 2015; Pouyanfar et al. 2018; Wu et al. 2019; Zhou et al. 2018a), or with applications to medical image analysis (Litjens et al. 2017), natural language processing (Young et al. 2018), speech recognition systems (Zhang et al. 2018d), and remote sensing (Zhu et al. 2017).

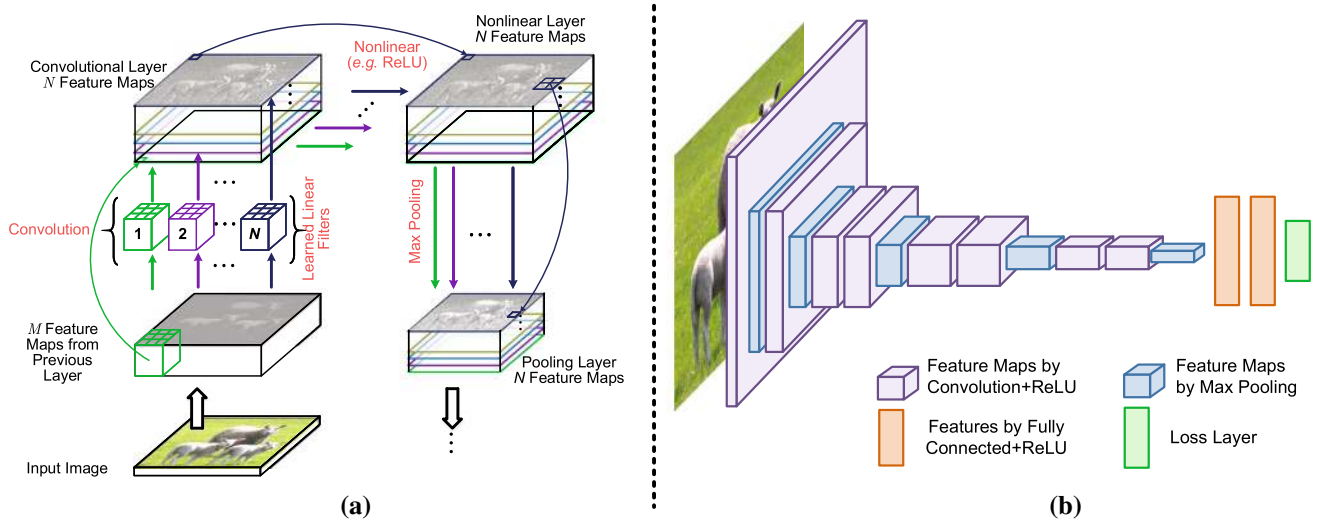

Fig. 8 **a** Illustration of three operations that are repeatedly applied by a typical CNN: convolution with a number of linear filters; Nonlinearities (e.g. ReLU); and local pooling (e.g. max pooling). The M feature maps from a previous layer are convolved with N different filters (here shown as size $3 \times 3 \times M$), using a stride of 1. The resulting N feature maps are then passed through a nonlinear function (e.g. ReLU), and pooled (e.g. taking a maximum over 2×2 regions) to give N feature maps at a reduced resolution. **b** Illustration of the architecture of VGGNet (Simonyan and Zisserman 2015), 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. 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 learned from labeled training data by optimizing an objective function (e.g. mean squared error or cross entropy loss) via stochastic gradient descent (Color figure online)

Convolutional Neural Networks (CNNs), the most representative models of deep learning, are able to exploit the basic properties underlying natural signals: translation invariance, local connectivity, and compositional hierarchies (LeCun et al. 2015). A typical CNN, illustrated in Fig. 8, has a hierarchical structure and is composed of a number of layers to learn representations of data with multiple levels of abstraction (LeCun et al. 2015). We begin with a convolution

$$\boldsymbol{x}^{l-1} * \boldsymbol{w}^{l} \tag{1}$$

between an input feature map $\boldsymbol{x}^{l-1}$ at a feature map from previous layer $l-1$, convolved with a 2D convolutional kernel (or filter or weights) $\boldsymbol{w}^{l}$. This convolution appears over a sequence of layers, subject to a nonlinear operation σ, such that

$$\boldsymbol{x}_j^{l} = \sigma\left(\sum_{i=1}^{N^{l-1}} \boldsymbol{x}_i^{l-1} * \boldsymbol{w}_{i,j}^{l} + b_j^{l}\right), \tag{2}$$

with a convolution now between the N^{l-1} input feature maps $\boldsymbol{x}_i^{l-1}$ and the corresponding kernel $\boldsymbol{w}_{i,j}^{l}$, plus a bias term b_j^{l}. The elementwise nonlinear function $\sigma(\cdot)$ is typically a rectified linear unit (ReLU) for each element,

$$\sigma(x) = \max\{x, 0\}. \tag{3}$$

Finally, pooling corresponds to the downsampling/upsampling of feature maps. These three operations (convolution, nonlinearity, pooling) are illustrated in Fig. 8a; CNNs having a large number of layers, a "deep" network, are referred to as Deep CNNs (DCNNs), with a typical DCNN architecture illustrated in Fig. 8b.

Most layers of a CNN consist 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 $\boldsymbol{w}_{i,j}$ (essentially a set of 2D filters). As can be seen in Fig. 8b, where the early CNN layers are typically composed of convolutional and pooling layers, the later layers are normally fully connected. From earlier to later layers, the input image is repeatedly convolved, and with each layer, the receptive field or region of support increases. In general, the initial CNN layers extract low-level features (e.g., edges), with later layers extracting more general features of increasing complexity (Zeiler and Fergus 2014; Bengio et al. 2013; LeCun et al. 2015; Oquab et al. 2014).

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 successful 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 and a requirement of expensive computing resources, and considerable skill and experience are still needed to select appropriate learning parameters and network architectures. Trained networks are poorly interpretable, there is a lack of robustness to degradations, and many DCNNs have shown serious vulnerability to attacks (Goodfellow et al. 2015), all of which currently limit the use of DCNNs in real-world applications.

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, recently, deep learning techniques have brought tremendous success to many visual recognition problems, and it is the large amounts of annotated data which play a key role in their success. Access to large numbers of images on the Internet makes it possible to build comprehensive datasets in order to capture a vast richness and diversity of objects, enabling unprecedented performance in object recognition.

For generic object detection, there are four famous datasets: PASCAL VOC (Everingham et al. 2010, 2015), ImageNet (Deng et al. 2009), MS COCO (Lin et al. 2014) and Open Images (Kuznetsova et al. 2018). The attributes of these datasets are summarized in Table 2, 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 collected images, typically by designing crowdsourcing strategies. Recognizing space limitations, we refer interested readers to the original papers (Everingham et al. 2010, 2015; Lin et al. 2014; Russakovsky et al. 2015; Kuznetsova et al. 2018) for detailed descriptions 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 3. The most frequent object classes in VOC, COCO, ILSVRC and Open Images detection datasets are visualized in Table 4.

PASCAL VOC Everingham et al. (2010, 2015) is a multi-year 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. Since 2009, the number of images has grown every year, but with all previous images retained to allow test results to 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 (Russakovsky et al. 2015), is derived from ImageNet (Deng et al. 2009), scaling 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. ImageNet1000, a subset of ImageNet images 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.

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 for richer image understanding, researchers created the MS COCO database (Lin et al. 2014) 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 (Lin et al. 2014) features two object detection tasks: using either bounding box output or object instance segmentation output. COCO introduced three new challenges:

1. It contains objects at a wide range of scales, including a high percentage of small objects (Singh and Davis 2018);
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.

OICOD (the Open Image Challenge Object Detection) is derived from Open Images V4 (now V5 in 2019) (Kuznetsova et al. 2018), currently the largest publicly available object

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

Table 2 Popular databases for object recognition

Dataset name	Total images	Categories	Images per category	Objects per image	Image size	Started year	Highlights
PASCAL VOC (2012) (Everingham et al. 2015)	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 intraclass variations; Objects in scene context; Multiple objects in one image; Contains many difficult samples
ImageNet (Russakovsky et al. 2015)	14 millions+	21,841	–	1.5	500 × 400	2009	Large number of object categories; More instances and more categories of objects per image; More challenging than PASCAL VOC; Backbone of the ILSVRC challenge; Images are object-centric
MS COCO (Lin et al. 2014)	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
Places (Zhou et al. 2017a)	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 (Kuznetsova et al. 2018)	9 millions+	6000+	–	8.3	Varied	2017	Annotated with image level labels, object bounding boxes and visual relationships; Open Images V5 supports large scale object detection, object instance segmentation and visual relationship detection

Example images from PASCAL VOC, ImageNet, MS COCO and Open Images are shown in Fig. 9

Fig. 9 Some example images with object annotations from PASCAL VOC, ILSVRC, MS COCO and Open Images. See Table 2 for a summary of these datasets

detection dataset. 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, bounding box annotations and instance segmentation mask annotations, but also regarding the annotation process. In ILSVRC and MS COCO, instances of all

Table 3 Statistics of commonly used object detection datasets

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	2501	2510	4952	6301(7844)	6307(7818)	5011	12,608	2.5
VOC08	20	2111	2221	4133	5082(6337)	5281(6347)	4332	10,364	2.4
VOC09	20	3473	3581	6650	8505(9760)	8713(9779)	7054	17,218	2.3
VOC10	20	4998	5105	9637	11,577(13,339)	11,797(13,352)	10,103	23,374	2.4
VOC11	20	5717	5823	10,994	13,609(15,774)	13,841(15,787)	11,540	27,450	2.4
VOC12	20	5717	5823	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	5000	40,670	860,001	36,781	123,287	896,782	7.3
MS COCO18	80	118,287	5000	40,670	860,001	36,781	123,287	896,782	7.3
Open images challenge object detection (*OICOD*) (*Based on open images V4* Kuznetsova et al. 2018)									
OICOD18	500	1,643,042	100,000	99,999	11,498,734	696,410	1,743,042	12,195,144	7.0

Object statistics for VOC challenges list the non-difficult 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, the 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 the 2017 Test Dev/Challenge splits contain the same images as the 2015 Test Dev/Challenge splits, so results across the years are directly comparable

classes in the dataset are exhaustively annotated, whereas for Open Images V4 a classifier was applied to each image and only those labels with sufficiently high scores were sent for human verification. Therefore in OICOD only the object instances of human-confirmed positive labels are annotated.

4.2 Evaluation Criteria

There are three criteria for evaluating the performance of detection algorithms: detection speed in 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 Everingham et al. (2010), Everingham et al. (2015), Russakovsky et al. (2015), Hoiem et al. (2012).

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 object j, of Bounding Box (BB) b_j, predicted category c_j, and confidence p_j. A predicted detection (b, c, p) is regarded as a True Positive (TP) if

- The predicted category c equals the ground truth label c_g.
- The overlap ratio IOU (Intersection Over Union) (Everingham et al. 2010; Russakovsky et al. 2015)

$$\text{IOU}(b, b^g) = \frac{area\,(b \cap b^g)}{area\,(b \cup b^g)}, \tag{4}$$

between the predicted BB b and the ground truth b^g is not smaller than a predefined threshold ε, where $\cap$ and

[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 (Russakovsky et al. 2015).

Table 4 Most frequent object classes for each detection challenge

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

(c) ILSVRC (200 Classes)

(d) Open Images Detection Challenge (500 Classes)

The size of each word is proportional to the frequency of that class in the training dataset

cup denote intersection and union, respectively. A typical value of ε is 0.5.

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

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 confidence p_{ij} in decreasing order. Each detection (b_{ij}, p_{ij}) is either a TP or an 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 $R(\beta)$ (Everingham et al. 2010) can be computed as a function of the confidence threshold β, so by varying the confidence threshold different pairs (P, R) can be obtained, in principle allowing precision to be regarded as a function of recall, i.e. $P(R)$, from which the Average Precision (AP) (Everingham et al. 2010; Russakovsky et al. 2015) 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 the objects in MS COCO are small and 24% are large, 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, with metric modifications for the Open Images challenges proposed in Kuznetsova et al. (2018).

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 (Viola and Jones 2001; Dalal and Triggs 2005; Felzenszwalb et al. 2008; Harzallah et al. 2009; Vedaldi et al. 2009) to learned DCNN features (Girshick et al. 2014; Ouyang et al. 2015; Girshick 2015; Ren et al. 2015; Dai et al. 2016c). In contrast, in terms of localization, the basic "sliding window" strategy (Dalal and Triggs 2005; Felzenszwalb et al. 2010b, 2008) remains mainstream, although with some efforts to avoid exhaustive search (Lampert et al. 2008; Uijlings et al. 2013). However, the number of windows is large and grows quadratically with the number of image pixels, and the need to search over multiple scales and aspect ratios further increases the search space. Therefore, the design of efficient and effective detection frameworks plays a key role in reducing this computational cost. Commonly adopted strategies include cascading, sharing feature computation, and reducing per-window computation.

This section reviews detection frameworks, listed in Fig. 11 and Table 11, the milestone approaches appearing since deep learning entered the field, organized into two main categories:

(a) Two stage detection frameworks, which include a preprocessing step for generating object proposals;
(b) One stage detection frameworks, or region proposal free frameworks, having a single proposed method which does not separate the process of the detection proposal.

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

Table 5 Summary 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	
β	Confidence threshold	A confidence threshold for computing $P(\beta)$ and $R(\beta)$	
ε	IOU threshold	VOC	Typically around 0.5
		ILSVRC	$\min(0.5, \frac{wh}{(w+10)(h+10)})$; $w \times h$ is the size of a GT box
		MS COCO	Ten IOU thresholds $\varepsilon \in \{0.5 : 0.05 : 0.95\}$
$P(\beta)$	Precision	The fraction of correct detections out of the total detections returned by the detector with confidence of at least β	
$R(\beta)$	Recall	The fraction of all N_c objects detected by the detector having a confidence of at least β	
AP	Average Precision	Computed over the different levels of recall achieved by varying the confidence β	
mAP	mean Average Precision	VOC	AP at a single IOU and averaged over all classes
		ILSVRC	AP at a modified IOU and averaged over all classes
		MS COCO	AP_{coco}: mAP averaged over ten IOUs: $\{0.5 : 0.05 : 0.95\}$;
			$AP_{coco}^{\text{IOU=0.5}}$: mAP at IOU $= 0.50$ (PASCAL VOC metric);
			$AP_{coco}^{\text{IOU=0.75}}$: mAP at IOU $= 0.75$ (strict metric);
			AP_{coco}^{small}: mAP for small objects of area smaller than 32^2;
			$AP_{coco}^{\text{medium}}$: mAP for objects of area between 32^2 and 96^2;
			AP_{coco}^{large}: mAP for large objects of area bigger than 96^2;
AR	Average Recall	The maximum recall given a fixed number of detections per image, averaged over all categories and IOU thresholds	
AR	Average Recall	MS COCO	$AR_{coco}^{\text{max=1}}$: AR given 1 detection per image;
			$AR_{coco}^{\text{max=10}}$: AR given 10 detection per image;
			$AR_{coco}^{\text{max=100}}$: AR given 100 detection per image;
			AR_{coco}^{small}: AR for small objects of area smaller than 32^2;
			$AR_{coco}^{\text{medium}}$: AR for objects of area between 32^2 and 96^2;
			AR_{coco}^{large}: AR for large objects of area bigger than 96^2;

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 $\boldsymbol{a}(j) = 1$ 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

Sections 6–9 will discuss fundamental sub-problems involved in detection frameworks in greater detail, including DCNN features, detection proposals, and context modeling.

5.1 Region Based (Two Stage) Frameworks

In a region-based framework, category-independent region proposals[5] are generated from an image, CNN (Krizhevsky et al. 2012a) 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 (Szegedy et al. 2013), OverFeat (Sermanet et al. 2014), MultiBox (Erhan et al. 2014) and RCNN (Girshick et al. 2014) independently and almost simultaneously proposed using CNNs for generic object detection.

RCNN (Girshick et al. 2014): Inspired by the breakthrough image classification results obtained by CNNs and the success of the selective search in region proposal for handcrafted features (Uijlings et al. 2013), Girshick et al. (2014, 2016) were among the first to explore CNNs for generic object detection and developed RCNN, which integrates

[5] Object proposals, also called region proposals or detection proposals, are a set of candidate regions or bounding boxes in an image that may potentially contain an object (Chavali et al. 2016; Hosang et al. 2016).

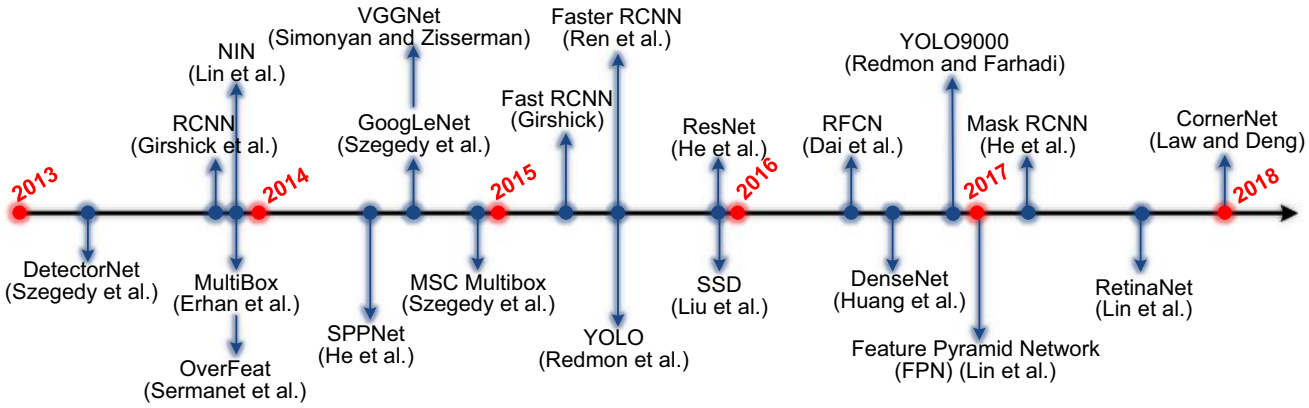

Fig. 11 Milestones in generic object detection

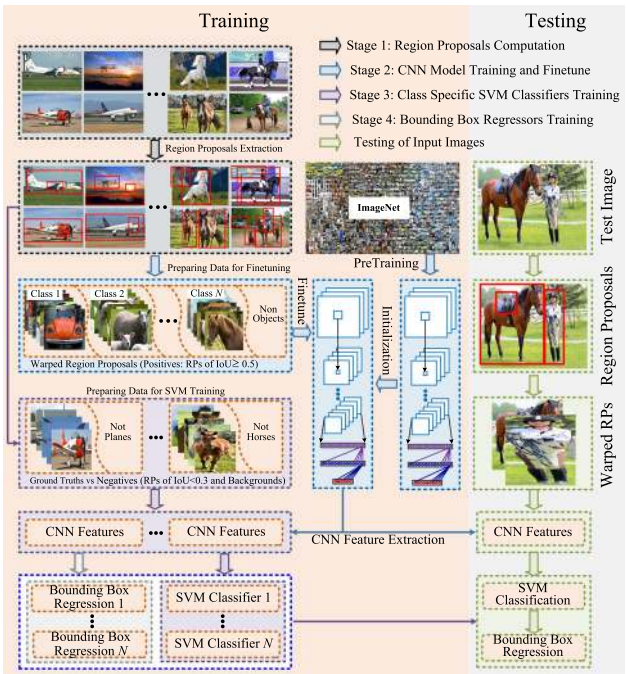

Fig. 12 Illustration of the RCNN detection framework (Girshick et al. 2014, 2016)

AlexNet (Krizhevsky et al. 2012a) with a region proposal selective search (Uijlings et al. 2013). As illustrated in detail in Fig. 12, training an RCNN framework consists of multi-stage pipelines:

1. *Region proposal computation* Class agnostic region proposals, which are candidate regions that might contain objects, are obtained via a selective search (Uijlings et al. 2013).
2. *CNN model finetuning* Region proposals, which are cropped from the image and warped into the same size, are used as the input for fine-tuning a CNN model pre-trained using a large-scale dataset such as ImageNet. At this stage, all region proposals with ≥ 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.
3. *Class specific SVM classifiers training* A set of class-specific linear SVM classifiers are trained using fixed length features extracted with CNN, replacing the soft-max classifier learned by fine-tuning. For training SVM classifiers, positive examples are defined 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 is negative for that class. Note that the positive and negative examples defined for training the SVM classifiers are different from those for fine-tuning the CNN.
4. *Class specific bounding box regressor 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 (Girshick 2015):

1. Training is a multistage pipeline, 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, posing great challenges for large scale detection, particularly with very deep networks, such as VGG16 (Simonyan and Zisserman 2015).
3. Testing is slow, since CNN features are extracted per object proposal in each test image, without shared computation.

All of these drawbacks have motivated successive innovations, leading to a number of improved detection frameworks such as SPPNet, Fast RCNN, Faster RCNN etc., as follows.

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

SPPNet (He et al. 2014) During testing, CNN feature extraction is the main bottleneck of the RCNN detection pipeline, which requires the extraction of CNN features from thousands of warped region proposals per image. As a result, He et al. (2014) introduced traditional spatial pyramid pooling (SPP) (Grauman and Darrell 2005; Lazebnik et al. 2006) into CNN architectures. Since convolutional layers accept inputs of arbitrary sizes, the requirement of fixed-sized images in CNNs is due only to the Fully Connected (FC) layers, therefore He et al. 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, fine-tuning in SPPNet (He et al. 2014) is unable to update the convolutional layers before the SPP layer, which limits the accuracy of very deep networks.

Fast RCNN (Girshick 2015) Girshick proposed Fast RCNN (Girshick 2015) 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 by developing a streamlined training process that simultaneously learns a softmax classifier and class-specific bounding box regression, rather than separately training a softmax classifier, SVMs, and Bounding Box Regressors (BBRs) 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. Essentially, RoI pooling uses warping at the feature level to approximate warping at the 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. Thus there is higher detection quality, a single training process that updates all network layers, and no storage required for feature caching.

Faster RCNN (Ren et al. 2015, 2017) Although Fast RCNN significantly sped up the detection process, it still relies on external region proposals, whose 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 (Zhou et al. 2015, 2016a; Cinbis et al. 2017; Oquab et al. 2015; Hariharan et al. 2016), an

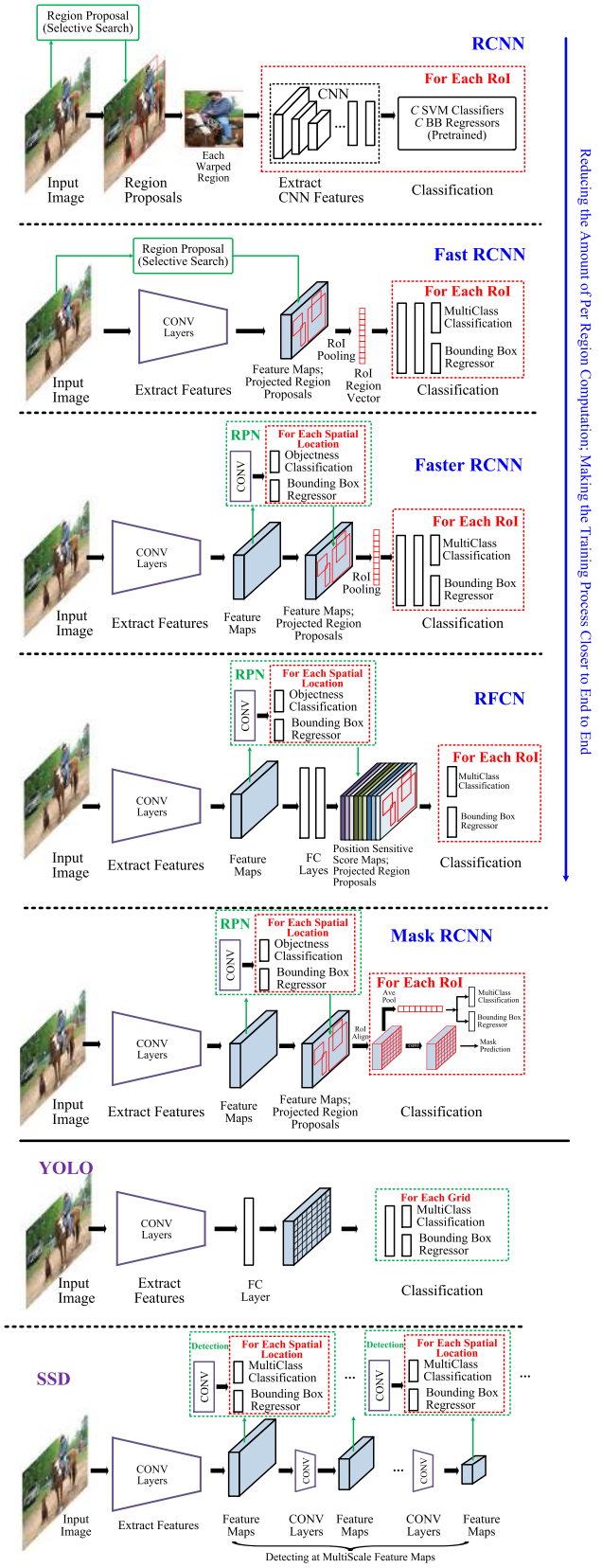

Fig. 13 High level diagrams of the leading frameworks for generic object detection. The properties of these methods are summarized in Table 11

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. (2015, 2017) offered an efficient and accurate Region Proposal Network (RPN) for generating region proposals. They utilize the same backbone network, using features from the last shared convolutional layer to accomplish the task of RPN for region proposal and Fast RCNN for region classification, as shown in Fig. 13.

RPN first initializes k reference boxes (i.e. the so called *anchors*) of different scales and aspect ratios at each CONV feature map location. The anchor *p*ositions are image content independent, but the feature vectors themselves, extracted from anchors, are image content dependent. Each anchor is mapped to a lower dimensional vector, which is fed into two sibling FC layers—an object category classification layer and a box regression layer. In contrast to detection in Fast RCNN, the features used for regression in RPN are of the same shape as the anchor box, thus 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) (Long et al. 2015; Shelhamer et al. 2017); Faster RCNN is thus a purely CNN based framework without using handcrafted features.

For the VGG16 model (Simonyan and Zisserman 2015), 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 Ren et al. (2015) contains several alternating training stages, later simplified in Ren et al. (2017).

Concurrent with the development of Faster RCNN, Lenc and Vedaldi (2015) 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 sub-network still needs to be applied per RoI (several hundred RoIs per image) led Dai et al. (2016c) to propose the RFCN detector which is *fully convolutional* (no hidden FC layers) with almost all computations shared over the entire image. As shown in Fig. 13, RFCN differs from Faster RCNN only in the RoI sub-network. In Faster RCNN, the computation after the RoI pooling layer cannot be shared, so Dai et al. (2016c) proposed using all CONV layers to construct a shared RoI sub-network, and RoI crops are taken from the last layer of CONV features prior to prediction. However, Dai et al. (2016c) 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 semantics, and less sensitive to translation, whereas object detection needs localization representations that respect translation invariance. Based on this observation, Dai et al. (2016c) 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 is added. They showed that RFCN with ResNet101 (He et al. 2016) could achieve comparable accuracy to Faster RCNN, often at faster running times.

Mask RCNN He et al. (2017) 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), but 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) (Long et al. 2015; Shelhamer et al. 2017) 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 (Xie et al. 2017; Lin et al. 2017a), 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 (He et al. 2017).

Chained Cascade Network and Cascade RCNN The essence of cascade (Felzenszwalb et al. 2010a; Bourdev and Brandt 2005; Li and Zhang 2004) is to learn more discriminative classifiers by using multistage classifiers, such that early stages discard a large number of easy negative samples so that later stages can focus on handling more difficult examples. Two-stage object detection can be considered as a cascade, the first detector removing large amounts of background, 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 were proposed in the Chained Cascade Network (Ouyang et al. 2017a), extended in Cascade RCNN (Cai and Vasconcelos 2018), and more recently applied for simultaneous object detection and instance segmentation (Chen et al. 2019a), winning the COCO 2018 Detection Challenge.

Light Head RCNN In order to further increase the detection speed of RFCN (Dai et al. 2016c), Li et al. (2018c) proposed Light Head RCNN, making the head of the detection network as light as possible to reduce the RoI computation. In particular, Li et al. (2018c) applied a convolution to produce thin feature maps with small channel numbers (e.g., 490 channels for COCO) and a cheap RCNN sub-network, leading to an excellent trade-off of speed and accuracy.

5.2 Unified (One Stage) Frameworks

The region-based pipeline strategies of Sect. 5.1 have dominated since RCNN (Girshick et al. 2014), such that the leading results on popular benchmark datasets are all based on Faster RCNN (Ren et al. 2015). Nevertheless, 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 in a monolithic setting that does not involve region proposal generation or post classification / feature resampling, encapsulating all computation in a single network. Since the whole pipeline is a single network, it can be optimized end-to-end directly on detection performance.

DetectorNet (Szegedy et al. 2013) 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 (Krizhevsky et al. 2012a) and replace the final softmax classifier layer with 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. The network needs to be trained per object type and mask type, and does not scale 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. (2014) and illustrated in Fig. 14, can be considered as one of the first single-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. OverFeat performs object detection via a single forward pass through the fully convolutional layers in the network (i.e. the "Feature Extractor", shown in Fig. 14a). The key steps of object detection at test time can be summarized as follows:

1. *Generate object candidates by performing object classification via a sliding window fashion on multiscale images* OverFeat uses a CNN like AlexNet (Krizhevsky et al. 2012a), which would require input images ofa fixed size due to its fully connected layers, in order to make the sliding window approach computationally efficient, OverFeat casts the network (as shown in Fig. 14a) into a fully convolutional network, taking inputs of any size, by viewing fully connected layers as convolutions with kernels of size 1×1. 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. 14b), resulting in a significantly increased number of evaluated context views. For each of the multiscale inputs, the classifier outputs a grid of predictions (class and confidence).
2. *Increase the number of predictions by offset max pooling* In order to increase resolution, OverFeat applies offset max pooling after the last CONV layer, i.e. performing a subsampling operation at every offset, yielding many more views for voting, increasing robustness while remaining efficient.
3. *Bounding box regression* Once an object is identified, a single bounding box regressor is applied. The classifier and the regressor share the same feature extraction (CONV) layers, only the FC layers need to be recomputed after computing the classification network.
4. *Combine predictions* OverFeat uses a greedy merge strategy to combine the individual bounding box predictions across all locations and scales.

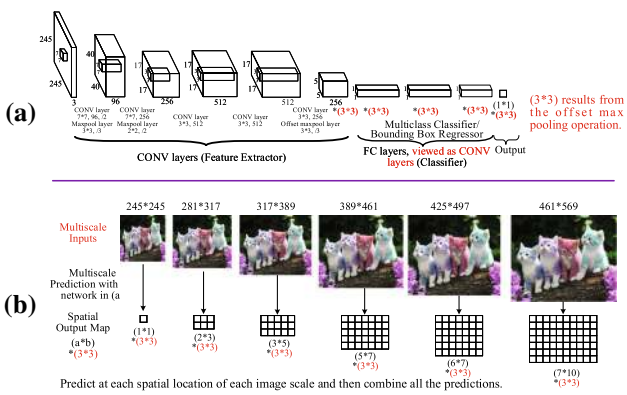

Fig. 14 Illustration of the OverFeat (Sermanet et al. 2014) detection framework

OverFeat has a significant speed advantage, but is less accurate than RCNN (Girshick et al. 2014), because it was difficult to train fully convolutional networks at the time. The speed advantage derives from sharing the computation of convolution between overlapping windows in the fully convolutional network. OverFeat is similar to later frameworks such as YOLO (Redmon et al. 2016) and SSD (Liu et al. 2016), except that the classifier and the regressors in OverFeat are trained sequentially.

YOLO Redmon et al. (2016) proposed YOLO (You Only Look Once), a unified detector casting object detection as a regression problem from image pixels to spatially separated bounding boxes and associated class probabilities, 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 a local region, YOLO uses features from an entire image globally. In particular, YOLO divides an image into an $S \times S$ grid, each predicting C class probabilities, B bounding box locations, and confidence scores. By throwing out the region proposal generation step entirely, YOLO is fast by design, running in real time at 45 FPS and Fast YOLO (Redmon et al. 2016) 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 in the 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 Redmon et al. (2016), YOLO may fail to localize some objects, especially small ones, possibly because of the coarse grid division, and because each grid cell can only contain one object. It is unclear to what extent YOLO can translate to good performance on datasets with many objects per image, such as MS COCO.

YOLOv2 and YOLO9000 Redmon and Farhadi (2017) proposed YOLOv2, an improved version of YOLO, in which the custom GoogLeNet (Szegedy et al. 2015) network is replaced with the simpler DarkNet19, plus batch normalization (He et al. 2015), removing the fully connected layers, and using good anchor boxes[8] learned via *k*means and multiscale training. YOLOv2 achieved state-of-the-art on standard detection tasks. Redmon and Farhadi (2017) also introduced YOLO9000, which can detect over 9000 object categories in real time by proposing a joint optimization method to train simultaneously on an ImageNet classification dataset and a 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 do not have bounding box annotations.

SSD In order to preserve real-time speed without sacrificing too much detection accuracy, Liu et al. (2016) proposed SSD (Single Shot Detector), faster than YOLO (Redmon et al. 2016) and with an accuracy competitive with region-based detectors such as Faster RCNN (Ren et al. 2015). SSD effectively combines ideas from RPN in Faster RCNN (Ren et al. 2015), YOLO (Redmon et al. 2016) and multiscale CONV features (Hariharan et al. 2016) to achieve fast detection speed, while still retaining high detection quality. Like YOLO, SSD predicts a fixed number of bounding boxes and scores, 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 (Simonyan and Zisserman 2015), followed by several auxiliary CONV layers, progressively decreasing in size. The information in the last layer may be too coarse spatially to allow precise localization, so SSD performs detection over multiple scales by operating on multiple CONV feature maps, each of which predicts category scores and box offsets for bounding boxes of appropriate sizes. For a 300×300 input, SSD achieves 74.3% mAP on the VOC2007 test at 59 FPS versus Faster RCNN 7 FPS / mAP 73.2% or YOLO 45 FPS / mAP 63.4%.

CornerNet Recently, Law and Deng (2018) questioned the dominant role that anchor boxes have come to play in SoA object detection frameworks (Girshick 2015; He et al. 2017; Redmon et al. 2016; Liu et al. 2016). Law and Deng (2018) argue that the use of anchor boxes, especially in one stage detectors (Fu et al. 2017; Lin et al. 2017b; Liu et al. 2016; Redmon et al. 2016), has drawbacks (Law and Deng 2018; Lin et al. 2017b) such as causing a huge imbalance between positive and negative examples, slowing down training and introducing extra hyperparameters. Borrowing ideas from the work on Associative Embedding in multiperson pose estimation (Newell et al. 2017), Law and Deng (2018) proposed CornerNet by formulating bounding box object detection as detecting paired top-left and bottom-right keypoints[9]. In CornerNet, the backbone network consists of two stacked Hourglass networks (Newell et al. 2016), with a simple corner pooling approach to better localize corners. CornerNet achieved a 42.1% AP on MS COCO, outperforming all previous one stage detectors; however, the average inference time is about 4FPS on a Titan X GPU, significantly slower than SSD (Liu et al. 2016) and YOLO (Redmon et al. 2016). CornerNet generates 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, Duan et al. (2019) proposed CenterNet to detect each object as a triplet of keypoints, by introducing one extra keypoint at the centre of a proposal, raising the MS COCO AP to 47.0%, but with an inference speed 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 (Dickinson et al. 2009; Girshick et al. 2014; Gidaris and Komodakis 2015; Zhu et al. 2016a). In the past, a great deal of effort was devoted to designing local descriptors [e.g., SIFT (Lowe 1999) and HOG (Dalal and Triggs 2005)] and to explore approaches [e.g., Bag of Words (Sivic and Zisserman 2003) and Fisher Vector (Perronnin et al. 2010)] to group and

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

[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 (TychsenSmith and Petersson 2017).

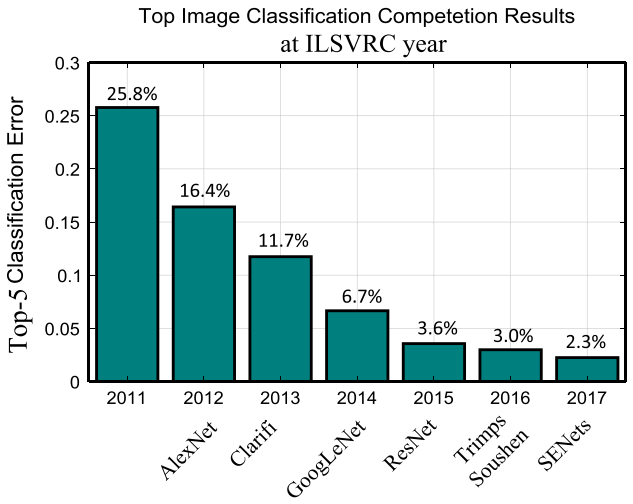

Fig. 15 Performance of winning entries in the ILSVRC competitions from 2011 to 2017 in the image classification task

abstract descriptors into higher level representations in order to allow the discriminative parts to emerge; however, these feature representation methods required careful engineering and considerable domain expertise.

In contrast, deep learning methods (especially *deep* CNNs) can learn powerful feature representations with multiple levels of abstraction directly from raw images (Bengio et al. 2013; LeCun et al. 2015). As the learning procedure reduces the dependency of specific domain knowledge and complex procedures needed in traditional feature engineering (Bengio et al. 2013; LeCun et al. 2015), the burden for feature representation has been transferred to the design of better network architectures and training procedures.

The leading frameworks reviewed in Sect. 5 [RCNN (Girshick et al. 2014), Fast RCNN (Girshick 2015), Faster RCNN (Ren et al. 2015), YOLO (Redmon et al. 2016), SSD (Liu et al. 2016)] have persistently promoted detection accuracy and speed, in which it is generally accepted that the CNN architecture (Sect. 6.1 and Fig. 15) plays a crucial role. As a result, most of the recent improvements in detection accuracy have been 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 (Sect. 3) serve as network backbones used in the detection frameworks of Sect. 5. Representative frameworks include AlexNet (Krizhevsky et al. 2012b), ZFNet (Zeiler and Fergus 2014) VGGNet (Simonyan and Zisserman 2015), GoogLeNet (Szegedy et al. 2015), Inception series (Ioffe and Szegedy 2015; Szegedy et al. 2016, 2017), ResNet (He et al. 2016), DenseNet (Huang et al. 2017a) and SENet (Hu et al. 2018b), summarized in Table 6, and where the improvement over time is seen in Fig. 15. A further review of recent CNN advances can be found in Gu et al. (2018).

The trend in architecture evolution is for greater depth: AlexNet has 8 layers, VGGNet 16 layers, more recently ResNet and DenseNet both surpassed the 100 layer mark, and it was VGGNet (Simonyan and Zisserman 2015) and GoogLeNet (Szegedy et al. 2015) which showed that increasing depth can improve the representational power. As can be observed from Table 6, networks such as AlexNet, OverFeat, ZFNet and VGGNet have an enormous number of parameters, despite being only a 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 great depth, actually have far fewer parameters by avoiding the use of FC layers.

With the use of Inception modules (Szegedy et al. 2015) in carefully designed topologies, the number of parameters of GoogLeNet is dramatically reduced, compared to AlexNet, ZFNet or 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 (He et al. 2016), InceptionResNets (Szegedy et al. 2017) combined the Inception networks with shortcut connections, on the basis that shortcut connections can significantly accelerate network training. Extending ResNets, Huang et al. (2017a) proposed DenseNets, which are built from dense blocksconnecting each layer to every other layer in a feedforward fashion, leading to compelling advantages such as parameter efficiency, implicit deep supervision[10], and feature reuse. Recently, He et al. (2016) proposed Squeeze and Excitation (SE) blocks, which can be combined with existing deep architectures to boost their performance at minimal additional computational cost, adaptively recalibrating channel-wise feature responses by explicitly modeling the interdependencies between convolutional feature channels, and which led to winning the ILSVRC 2017 classification task. Research on CNN architectures remains active, with emerging networks such as Hourglass (Law and Deng 2018), Dilated Residual Networks (Yu et al. 2017), Xception (Chollet 2017), DetNet (Li et al. 2018b), Dual Path Networks (DPN) (Chen et al. 2017b), FishNet (Sun et al. 2018), and GLoRe (Chen et al. 2019b).

[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) (Lee et al. 2015).

Table 6 DCNN architectures that were commonly used for generic object detection

No.	DCNN architecture	#Paras ($\times 10^6$)	#Layers (CONV+FC)	Test error (Top 5)	First used in	Highlights
1	AlexNet (Krizhevsky et al. 2012b)	57	5 + 2	15.3%	Girshick et al. (2014)	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) (Zeiler and Fergus 2014)	58	5 + 2	14.8%	He et al. (2014)	Similar to AlexNet, different in stride for convolution, filter size, and number of filters for some layers
3	OverFeat (Sermanet et al. 2014)	140	6 + 2	13.6%	Sermanet et al. (2014)	Similar to AlexNet, different in stride for convolution, filter size, and number of filters for some layers
4	VGGNet (Simonyan and Zisserman 2015)	134	13 + 2	6.8%	Girshick (2015)	Increasing network depth significantly by stacking 3×3 convolution filters and increasing the network depth step by step
5	GoogLeNet (Szegedy et al. 2015)	6	22	6.7%	Szegedy et al. (2015)	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
6	Inception v2 (Ioffe and Szegedy 2015)	12	31	4.8%	Howard et al. (2017)	Faster training with the introduce of batch normalization
7	Inception v3 (Szegedy et al. 2016)	22	47	3.6%		Inclusion of separable convolution and spatial resolution reduction
8	YOLONet (Redmon et al. 2016)	64	24 + 1	–	Redmon et al. (2016)	A network inspired by GoogLeNet used in YOLO detector
9	ResNet50 (He et al. 2016)	23.4	49	3.6% (ResNets)	He et al. (2016)	With identity mapping, substantially deeper networks can be learned

Table 6 continued

No.	DCNN architecture	#Paras ($\times 10^6$)	#Layers (CONV+FC)	Test error (Top 5)	First used in	Highlights
10	ResNet101 (He et al. 2016)	42	100		He et al. (2016)	Requires fewer parameters than VGG by using the global average pooling and bottleneck introduced in GoogLeNet
11	InceptionResNet v1 (Szegedy et al. 2017)	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 Szegedy et al. (2017)	30	95		(Huang et al. 2017b)	A costlier residual version of Inception, with significantly improved recognition performance
13	Inception v4 Szegedy et al. (2017)	41	75			An Inception variant without residual connections, with roughly the same recognition performance as InceptionResNet v2, but significantly slower
14	ResNeXt (Xie et al. 2017)	23	49	3.0%	Xie et al. (2017)	Repeating a building block that aggregates a set of transformations with the same topology
15	DenseNet201 (Huang et al. 2017a)	18	200	–	Zhou et al. (2018b)	Concatenate each layer with every other layer in a feed forward fashion. Alleviate the vanishing gradient problem, encourage feature reuse, reduction in number of parameters
16	DarkNet (Redmon and Farhadi 2017)	20	19	–	Redmon and Farhadi (2017)	Similar to VGGNet, but with significantly fewer parameters
17	MobileNet (Howard et al. 2017)	3.2	$27 + 1$	–	Howard et al. (2017)	Light weight deep CNNs using depth-wise separable convolutions
18	SE ResNet (Hu et al. 2018b)	26	50	2.3% (SENets)	Hu et al. (2018b)	Channel-wise attention by a novel block called *Squeeze and Excitation*. Complementary to existing backbone CNNs

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

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 (Ouyang et al. 2017b) that pretraining a deep model with a large scale dataset having object level annotations (such as ImageNet), instead of only the 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 (Razavian et al. 2014; Azizpour et al. 2016; Donahue et al. 2014; Yosinski et al. 2014), which can support a wider range of visual recognition tasks. For detection, the pre-trained network is typically fine-tuned[11] on a given detection dataset (Donahue et al. 2014; Girshick et al. 2014, 2016). Several large scale image classification datasets are used for CNN pre-training, among them ImageNet1000 (Deng et al. 2009; Russakovsky et al. 2015) with 1.2 million images of 1000 object categories, Places (Zhou et al. 2017a), which is much larger than ImageNet1000 but with fewer classes, a recent Places-Imagenet hybrid (Zhou et al. 2017a), or JFT300M (Hinton et al. 2015; Sun et al. 2017).

Pretrained CNNs without fine-tuning were explored for object classification and detection in Donahue et al. (2014), Girshick et al. (2016), Agrawal et al. (2014), 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 (Donahue et al. 2014; Girshick et al. 2016). Fine-tuning a pre-trained network can increase detection performance significantly (Girshick et al. 2014, 2016), although in the case of AlexNet, the fine-tuning performance boost was shown to be much larger for FC6 / FC7 than for Pool5, suggesting that Pool5 features are more general. Furthermore, the relationship between the source and target datasets plays a critical role, for example that ImageNet based CNN features show better performance for object detection than for human action (Zhou et al. 2015; Azizpour et al. 2016).

6.2 Methods For Improving Object Representation

Deep CNN based detectors such as RCNN (Girshick et al. 2014), Fast RCNN (Girshick 2015), Faster RCNN (Ren et al. 2015) and YOLO (Redmon et al. 2016), 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 representations; 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) (Felzenszwalb et al. 2010b; Girshick et al. 2014; He et al. 2014), which typically produces more accurate detection, with, however, obvious limitations of inference time and memory.

6.2.1 Handling of Object Scale Variations

Since a CNN computes its feature hierarchy layer by layer, the sub-sampling layers in the feature hierarchy already lead to an inherent multiscale pyramid, producing feature maps at different spatial resolutions, but subject to challenges (Hariharan et al. 2016; Long et al. 2015; Shrivastava et al. 2017). In particular, the 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. In contrast, lower layers have a small receptive field and rich geometric details, but the resolution is high and 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 or "atrous" convolution (Yu and Koltun 2015; Dai et al. 2016c; Chen et al. 2018b) have been proposed, increasing feature resolution, but increasing computational complexity. On the other hand, if the target object is large, then the semantic concept will emerge in much later layers. A number of methods (Shrivastava et al. 2017; Zhang et al. 2018e; Lin et al. 2017a; Kong et al. 2017) have been proposed to improve detection accuracy by exploiting multiple CNN layers, broadly falling into three types of **multiscale object detection**:

1. Detecting with combined features of multiple layers;
2. Detecting at multiple layers;
3. Combinations of the above two methods.

(1) *Detecting with combined features of multiple CNN layers* Many approaches, including Hypercolumns (Hariharan et al. 2016), HyperNet (Kong et al. 2016), and ION (Bell et al. 2016), 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 (Long et al. 2015; Shelhamer et al. 2017; Hariharan et al. 2016). As shown in Fig. 16a, ION (Bell et al. 2016) uses RoI pooling to extract RoI features from multiple

[11] Fine-tuning 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.

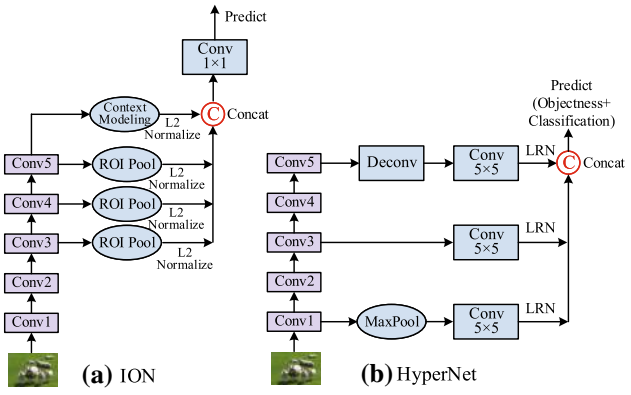

Fig. 16 Comparison of HyperNet and ION. LRN is local response normalization, which performs a kind of "lateral inhibition" by normalizing over local input regions (Jia et al. 2014)

layers, and then the object proposals generated by selective search and edgeboxes are classified by using the concatenated features. HyperNet (Kong et al. 2016), shown in Fig. 16b, follows a similar idea, and integrates deep, intermediate and shallow features to generate object proposals and to predict objects via an end to end joint training strategy. The combined feature is more descriptive, 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 (Liu et al. 2016) and MSCNN (Cai et al. 2016), RBFNet (Liu et al. 2018b), and DSOD (Shen et al. 2017). SSD (Liu et al. 2016) spreads out default boxes of different scales to multiple layers within a CNN, and forces each layer to focus on predicting objects of a certain scale. RFBNet (Liu et al. 2018b) 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 (Szegedy et al. 2015), but combining multiple branches with different kernels and convolution layers (Chen et al. 2018b). MSCNN (Cai et al. 2016) 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 (Liu et al. 2018b), TridentNet (Li et al. 2019b) constructs a parallel multibranch architecture where each branch shares the same transformation parameters but with different receptive fields; dilated convolution with different dilation rates are used to adapt the receptive fields for objects of different scales.

(3) *Combinations of the above two methods* Features from different layers are complementary to each other and can improve detection accuracy, as shown by Hypercolumns (Hariharan et al. 2016), HyperNet (Kong et al. 2016) and ION (Bell et al. 2016). On the other hand, however, it is natural to detect objects of different scales using features of approximately the same size, which can be achieved by detecting large objects from downscaled feature maps while detecting small objects from upscaled feature maps. Therefore, in order to combine the best of both worlds, some recent works propose to detect objects at multiple layers, and the resulting features obtained by combining features from different layers. This approach has been found to be effective for segmentation (Long et al. 2015; Shelhamer et al. 2017) and human pose estimation (Newell et al. 2016), has been widely exploited by both one-stage and two-stage detectors to alleviate problems of scale variation across object instances. Representative methods include SharpMask (Pinheiro et al. 2016), Deconvolutional Single Shot Detector (DSSD) (Fu et al. 2017), Feature Pyramid Network (FPN) (Lin et al. 2017a), Top Down Modulation (TDM)(Shrivastava et al. 2017), Reverse connection with Objectness prior Network (RON) (Kong et al. 2017), ZIP (Li et al. 2018a), Scale Transfer Detection Network (STDN) (Zhou et al. 2018b), RefineDet (Zhang et al. 2018a), StairNet (Woo et al. 2018), Path Aggregation Network (PANet) (Liu et al. 2018c), Feature Pyramid Reconfiguration (FPR) (Kong et al. 2018), DetNet (Li et al. 2018b), Scale Aware Network (SAN) (Kim et al. 2018), Multiscale Location aware Kernel Representation (MLKP) (Wang et al. 2018) and M2Det (Zhao et al. 2019), as shown in Table 7 and contrasted in Fig. 17.

Early works like FPN (Lin et al. 2017a), DSSD (Fu et al. 2017), TDM (Shrivastava et al. 2017), ZIP (Li et al. 2018a), RON (Kong et al. 2017) and RefineDet (Zhang et al. 2018a) construct the feature pyramid according to the inherent multiscale, pyramidal architecture of the backbone, and achieved encouraging results. As can be observed from Fig. 17a1–f1, these methods have very similar detection architectures which incorporate a top-down network with lateral connections to supplement the standard bottom-up, feed-forward 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, and the combined features are then used for detection. As can be seen from Fig. 17a2–e2, the main differences lie in the design of the simple Feature Fusion Block (FFB), which handles the selection of features from different layers and the combination of multilayer features.

FPN (Lin et al. 2017a) shows significant improvement as a generic feature extractor in several applications including object detection (Lin et al. 2017a, b) and instance segmentation (He et al. 2017). Using FPN in a basic Faster RCNN system achieved state-of-the-art results on the COCO detection dataset. STDN (Zhou et al. 2018b) used DenseNet (Huang et al. 2017a) to combine features of different layers and designed a scale transfer module to obtain feature maps

Table 7 Summary of properties of representative methods in improving DCNN feature representations for generic object detection

Group	Detector name	Region proposal	Backbone DCNN	Pipelined used	mAP@IoU = 0.5			mAP	Published in	Highlights
					VOC07	VOC12	COCO	COCO		
(1) Single detection with multilayer features	ION (Bell et al. 2016)	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 (Kong et al. 2016)	RPN	VGG16	Faster RCNN	76.3 (07+12)	71.4 (07T+12)	–	–	CVPR16	Use features from multiple layers for both region proposal and region classification
	PVANet (Kim et al. 2016)	RPN	PVANet	Faster RCNN	**84.9** (07+12+CO)	**84.2** (07T+12+CO)	–	–	NIPSW16	Deep but lightweight; Combine ideas from concatenated ReLU (Shang et al. 2016), Inception (Szegedy et al. 2015), and HyperNet (Kong et al. 2016)

Table 7 continued

Group	Detector name	Region proposal	Backbone DCNN	Pipelined used	mAP@IoU=0.5			mAP	Published in	Highlights
					VOC07	VOC12	COCO	COCO		
(2) Detection at multiple layers	SDP+CRC (Yang et al. 2016b)	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
	MSCNN (Cai et al. 2016)	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 (Zagoruyko et al. 2016)	SharpMask (Pinheiro et al. 2016)	VGG16	Fast RCNN	–	–	51.9	33.2	BMVC16	Concatenate features from different convolutional layers and features of different contextual regions; loss function for multiple overlap thresholds; ranked 2nd in both the COCO15 detection and segmentation challenges
	DSOD (Shen et al. 2017)	Free	DenseNet	SSD	77.7 (07+12)	72.2 (07T+12)	47.3	29.3	ICCV17	Concatenate feature sequentially, like DenseNet. Train from scratch on the target dataset without pre-training
	RFBNet (Liu et al. 2018b)	Free	VGG16	SSD	82.2 (07+12)	81.2 (07T+12)	55.7	34.4	ECCV18	Propose a multi-branch convolutional block similar to Inception (Szegedy et al. 2015), but using dilated convolution

Table 7 continued

Group	Detector name	Region proposal	Backbone DCNN	Pipelined used	mAP@IoU = 0.5			mAP	Published in	Highlights
					VOC07	VOC12	COCO	COCO		
(3) Combination of (1) and (2)	DSSD (Fu et al. 2017)	Free	ResNet101	SSD	81.5 (07+12)	80.0 (07T+12)	53.3	33.2	2017	Use Conv-Deconv, as shown in Fig. 17c1, c2
	FPN (Lin et al. 2017a)	RPN	ResNet101	Faster RCNN	–	–	59.1	36.2	CVPR17	Use Conv-Deconv, as shown in Fig. 17a1, a2; Widely used in detectors
	TDM (Shrivastava et al. 2017)	RPN	ResNet101 VGG16	Faster RCNN	–	–	57.7	36.8	CVPR17	Use Conv-Deconv, as shown in Fig. 17b2
	RON (Kong et al. 2017)	RPN	VGG16	Faster RCNN	81.3 (07+12+CO)	80.7 (07T+12+CO)	49.5	27.4	CVPR17	Use Conv-deconv, as shown in Fig. 17d2; Add the objectness prior to significantly reduce object search space
	ZIP (Li et al. 2018a)	RPN	Inceptionv2	Faster RCNN	79.8 (07+12)	–	–	–	IJCV18	Use Conv-Deconv, as shown in Fig. 17f1. Propose a map attention decision (MAD) unit for features from different layers

Table 7 continued

Group	Detector name	Region proposal	Backbone DCNN	Pipelined used	mAP@IoU = 0.5			mAP	Published in	Highlights
					VOC07	VOC12	COCO	COCO		
	STDN (Zhou et al. 2018b)	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 (Zhang et al. 2018a)	RPN	VGG16 ResNet101	Faster RCNN	83.8 (07+12)	83.5 (07T+12)	62.9	41.8	CVPR18	Use cascade to obtain better and less anchors. Use Conv-deconv, as shown in Fig. 17e2 to improve features
	PANet (Liu et al. 2018c)	RPN	ResNeXt101 +FPN	Mask RCNN	–	–	**67.2**	**47.4**	CVPR18	Shown in Fig. 17g. Based on FPN, add another bottom-up path to pass information between lower and topmost layers; adaptive feature pooling. Ranked 1st and 2nd in COCO 2017 tasks
	DetNet (Li et al. 2018b)	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. 17i
	FPR (Kong et al. 2018)	–	VGG16 ResNet101	SSD	82.4 (07+12)	81.1 (07T+12)	54.3	34.6	ECCV18	Fuse task oriented features across different spatial locations and scales, globally and locally; Shown in Fig. 17h
	M2Det (Zhao et al. 2019)	–	SSD	VGG16 ResNet101	–	–	64.6	44.2	AAAI19	Shown in Fig. 17j, newly designed top down path to learn a set of multilevel features, recombined to construct a feature pyramid for object detection

Table 7 continued

Group	Detector name	Region proposal	Backbone DCNN	Pipelined used	mAP@IoU = 0.5			mAP	Published in	Highlights
					VOC07	VOC12	COCO	COCO		
(4) Model geometric transforms	DeepIDNet (Ouyang et al. 2015)	SS+ EB	AlexNet ZFNet OverFeat GoogLeNet	RCNN	69.0 (07)	–	–	25.6	CVPR15	Introduce a deformation constrained pooling layer, 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 (Dai et al. 2017)	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 (Mordan et al. 2018)	AttractioNet (Gidaris and Komodakis 2016)	ResNet	RFCN	83.3 (07+12)	81.2 (07T+12)	59.1	39.1	IJCV18	Design a deformable part based RoI pooling layer to explicitly select discriminative regions around object proposals

Details for Groups (1), (2), and (3) are provided in Sect. 6.2. 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 additional COCO results are computed as the average of mAP for IoU thresholds from 0.5 to 0.95. Training data: "07"←VOC2007 trainval; "07T"←VOC2007 trainval and test; "12"←VOC2012 trainval; CO← COCO trainval. The COCO detection results were reported with COCO2015 Test-Dev, except for MPN (Zagoruyko et al. 2016) which reported with COCO2015 Test-Standard

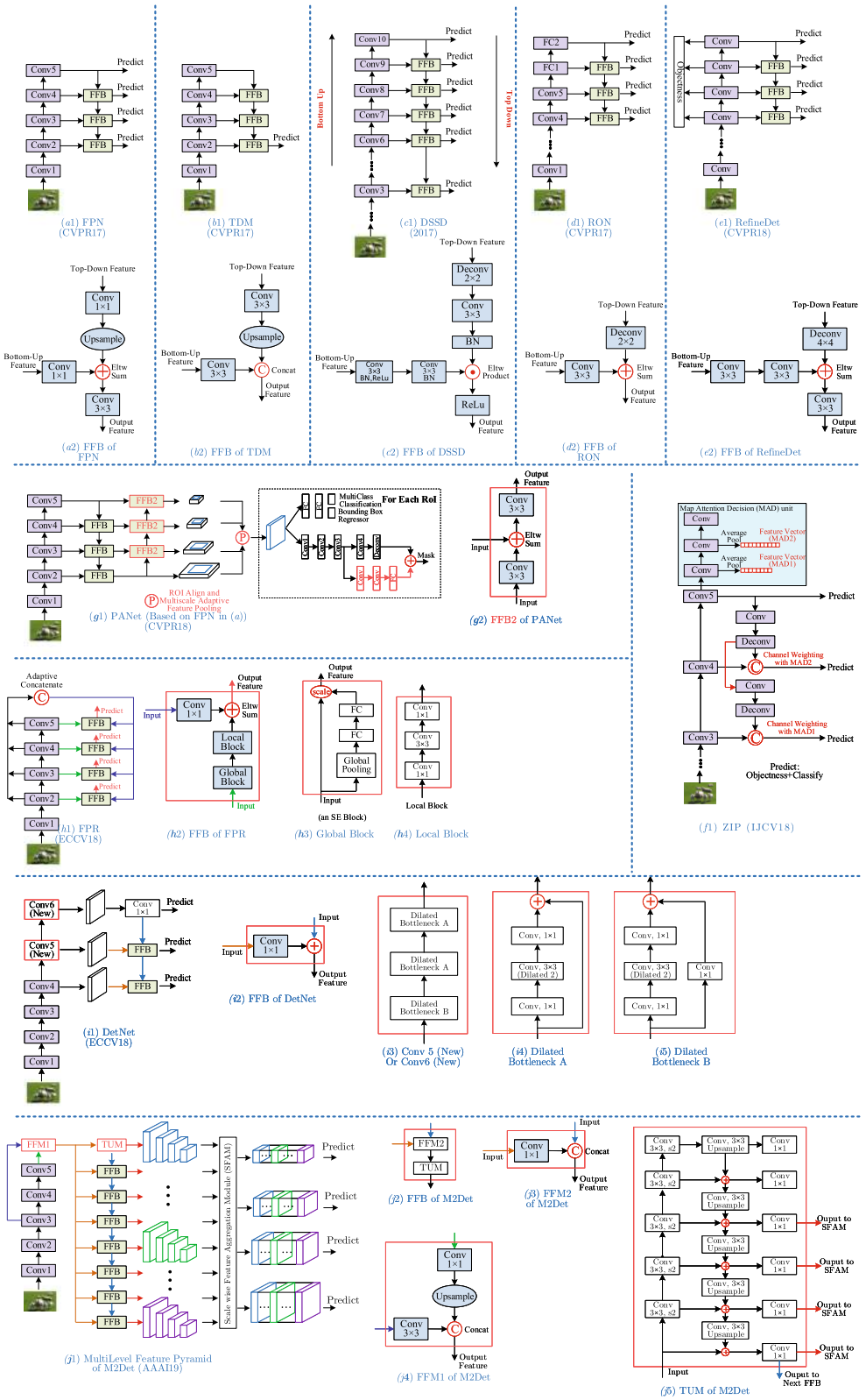

Fig. 17 Hourglass architectures: Conv1 to Conv5 are the main Conv blocks in backbone networks such as VGG or ResNet. The figure compares a number of feature fusion blocks (FFB) commonly used in recent approaches: FPN (Lin et al. 2017a), TDM (Shrivastava et al. 2017), DSSD (Fu et al. 2017), RON (Kong et al. 2017), RefineDet (Zhang et al. 2018a), ZIP (Li et al. 2018a), PANet (Liu et al. 2018c), FPR (Kong et al. 2018), DetNet (Li et al. 2018b) and M2Det (Zhao et al. 2019). *FFM* feature fusion module, *TUM* thinned U-shaped module

with different resolutions. The scale transfer module can be directly embedded into DenseNet with little additional cost.

More recent work, such as PANet (Liu et al. 2018c), FPR (Kong et al. 2018), DetNet (Li et al. 2018b), and M2Det (Zhao et al. 2019), as shown in Fig. 17g–j, propose to further improve on the pyramid architectures like FPN in different ways. Based on FPN, Liu et al. designed PANet (Liu et al. 2018c) (Fig. 17g1) by adding another bottom-up path with clean lateral connections from low to top levels, in order to shorten the information path and to enhance the feature pyramid. Then, an adaptive feature pooling was proposed to aggregate features from all feature levels for each proposal. In addition, in the proposal sub-network, 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 allowed PANet to reach 1st place in the COCO 2017 Challenge Instance Segmentation task and 2nd place in the Object Detection task. Kong et al. proposed FPR (Kong et al. 2018) by explicitly reformulating the feature pyramid construction process [e.g. FPN (Lin et al. 2017a)] as feature reconfiguration functions in a highly nonlinear but efficient way. As shown in Fig. 17h1, instead of using a top-down path to propagate strong semantic features from the topmost layer down as in FPN, FPR first extracts features from multiple layers in the backbone network by adaptive concatenation, and then designs a more complex FFB module (Fig. 17h2) to spread strong semantics to all scales. Li et al. (2018b) proposed DetNet (Fig. 17i1) 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. (2019) 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. 17j1, features from two different layers of the backbone are first fused as the base feature, after which a top-down path with lateral connections from the base feature is created to build the feature pyramid. As shown in Fig. 17j2, j5, the FFB module is much more complex than those like FPN, in that FFB involves a Thinned U-shaped Module (TUM) to generate a second pyramid structure, after which the feature maps with equivalent sizes from multiple TUMs are combined for object detection. The authors proposed M2Det by integrating MLFPN into SSD, and achieved better detection performance than other one-stage detectors.

6.3 Handling of Other Intraclass Variations

Powerful object representations should combine distinctiveness and robustness. A large amount of recent work has been devoted to handling changes in object scale, as reviewed in Sect. 6.2.1. As discussed in Sect. 2.2 and summarized in Fig. 5, object detection still requires robustness to real-world variations other than just scale, which we group into three categories:

- Geometric transformations,
- Occlusions, and
- Image degradations.

To handle these intra-class variations, the most straightforward approach is to augment the training datasets with a sufficient amount of variations; for example, robustness to rotation could be achieved by adding rotated objects at many orientations to the training data. Robustness can frequently be learned this way, but usually at the cost of expensive training and complex model parameters. Therefore, researchers have proposed alternative solutions to these problems.

Handling of geometric transformations DCNNs are inherently limited by the lack of ability to be spatially invariant to geometric transformations of the input data (Lenc and Vedaldi 2018; Liu et al. 2017; Chellappa 2016). The introduction of local max pooling layers has allowed DCNNs to enjoy some translation invariance, however the intermediate feature maps are not actually invariant to large geometric transformations of the input data (Lenc and Vedaldi 2018). Therefore, many approaches have been presented to enhance robustness, aiming at learning invariant CNN representations with respect to different types of transformations such as scale (Kim et al. 2014; Bruna and Mallat 2013), rotation (Bruna and Mallat 2013; Cheng et al. 2016; Worrall et al. 2017; Zhou et al. 2017b), or both (Jaderberg et al. 2015). One representative work is Spatial Transformer Network (STN) (Jaderberg et al. 2015), which introduces a new learnable module to handle scaling, cropping, rotations, as well as non-rigid deformations via a global parametric transformation. STN has now been used in rotated text detection (Jaderberg et al. 2015), rotated face detection and generic object detection (Wang et al. 2017).

Although rotation invariance may be attractive in certain applications, such as scene text detection (He et al. 2018; Ma et al. 2018), face detection (Shi et al. 2018), and aerial imagery (Ding et al. 2018; Xia et al. 2018), there is limited generic object detection work focusing on rotation invariance because popular benchmark detection datasets (e.g. PASCAL VOC, ImageNet, COCO) do not actually present rotated images.

Before deep learning, Deformable Part based Models (DPMs) (Felzenszwalb et al. 2010b) were successful for generic object detection, representing objects by component parts arranged in a deformable configuration. Although DPMs have been significantly outperformed by more recent object detectors, their spirit still deeply influences many recent detectors. DPM modeling is less sensitive to transformations in object pose, viewpoint and nonrigid deformations, motivating researchers (Dai et al. 2017; Girshick et al. 2015;

Mordan et al. 2018; Ouyang et al. 2015; Wan et al. 2015) to explicitly model object composition to improve CNN based detection. The first attempts (Girshick et al. 2015; Wan et al. 2015) combined DPMs with CNNs by using deep features learned by AlexNet in DPM based detection, but without region proposals. To enable a CNN to benefit from the built-in capability of modeling the deformations of object parts, a number of approaches were proposed, including DeepIDNet (Ouyang et al. 2015), DCN (Dai et al. 2017) and DPFCN (Mordan et al. 2018) (shown in Table 7). Although similar in spirit, deformations are computed in different ways: DeepIDNet (Ouyang et al. 2017b) designed a deformation constrained pooling layer to replace regular max pooling, to learn the shared visual patterns and their deformation properties across different object classes; DCN (Dai et al. 2017) designed a deformable convolution layer and a deformable RoI pooling layer, both of which are based on the idea of augmenting regular grid sampling locations in feature maps; and DPFCN (Mordan et al. 2018) proposed a 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 In real-world images, occlusions are common, resulting in information loss from object instances. A deformable parts idea can be useful for occlusion handling, so deformable RoI Pooling (Dai et al. 2017; Mordan et al. 2018; Ouyang and Wang 2013) and deformable convolution (Dai et al. 2017) have been proposed to alleviate occlusion by giving more flexibility to the typically fixed geometric structures. Wang et al. (2017) propose to learn an adversarial network that generates examples with occlusions and deformations, and context may be helpful in dealing with occlusions (Zhang et al. 2018b). Despite these efforts, the occlusion problem is far from being solved; applying GANs to this problem may be a promising 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, or the intentional low-cost sensors on edge devices and wearable devices. While low image quality may be expected to degrade the performance of visual recognition, most current methods are evaluated in a degradation free and clean environment, evidenced by the fact that PASCAL VOC, ImageNet, MS COCO and Open Images all focus on relatively high quality images. To the best of our knowledge, there is so far very limited work to address this problem.

7 Context Modeling

In the physical world, visual objects occur in particular environments and usually coexist with other related objects. There is strong psychological evidence (Biederman 1972; Bar 2004) that context plays an essential role in human object recognition, and it is recognized that a proper modeling of context helps object detection and recognition (Torralba 2003; Oliva and Torralba 2007; Chen et al. 2018b, 2015a; Divvala et al. 2009; Galleguillos and Belongie 2010), especially when object appearance features are insufficient because of small object size, object occlusion, or poor image quality. Many different types of context have been discussed (Divvala et al. 2009; Galleguillos and Belongie 2010), and can broadly be grouped into one of three categories:

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 (Chen et al. 2015b; Divvala et al. 2009; Galleguillos and Belongie 2010; Malisiewicz and Efros 2009; Murphy et al. 2003; Rabinovich et al. 2007; Parikh et al. 2012) preceded the prevalence of deep learning, and much of this work has yet to be explored in DCNN-based object detectors (Chen and Gupta 2017; Hu et al. 2018a).

The current state of the art in object detection (Ren et al. 2015; Liu et al. 2016; He et al. 2017) detects objects without explicitly exploiting any contextual information. It is broadly agreed that DCNNs make use of contextual information implicitly (Zeiler and Fergus 2014; Zheng et al. 2015) since they learn hierarchical representations with multiple levels of abstraction. Nevertheless, there is value in exploring contextual information explicitly in DCNN based detectors (Hu et al. 2018a; Chen and Gupta 2017; Zeng et al. 2017), 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 Zhang et al. (2013), Galleguillos and Belongie (2010). Representative approaches are summarized in Table 8.

7.1 Global Context

Global context (Zhang et al. 2013; Galleguillos and Belongie 2010) refers to image or scene level contexts, which can serve as cues for object detection (e.g., a bedroom will predict the presence of a bed). In DeepIDNet (Ouyang et al. 2015), the image classification scores were used as contextual features, and concatenated with the object detection scores to improve detection results. In ION (Bell et al. 2016), Bell et al. proposed to use spatial Recurrent Neural Networks (RNNs) to explore contextual information across the entire image. In SegDeepM (Zhu et al. 2015), Zhu et al. proposed a Markov random field model that scores appearance as well as context

Table 8 Summary of detectors that exploit context information, with labelling details as in Table 7

Group	Detector name	Region proposal	Backbone DCNN	Pipelined Used	mAP@IoU = 0.5		mAP	Published in	Highlights
					VOC07	VOC12	COCO		
Global context	SegDeepM (Zhu et al. 2015)	SS+CMPC	VGG16	RCNN	VOC10	VOC12	–	CVPR15	Additional features extracted from an enlarged object proposal as context information
	DeepIDNet (Ouyang et al. 2015)	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 (Bell et al. 2016)	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 (Shrivastava and Gupta 2016)	RPN	VGG16	Faster RCNN	76.4 (07+12)	72.6 (07T+12)	–	ECCV16	Use semantic segmentation to provide top-down feedback
Local context	MRCNN (Gidaris and Komodakis 2015)	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 (Zeng et al. 2016, 2017)	CRAFT (Yang et al. 2016a)	Inception v2 ResNet269 PolyNet (Zhang et al. 2017)	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

Table 8 continued

Group	Detector name	Region proposal	Backbone DCNN	Pipelined Used	mAP@IoU = 0.5		mAP	Published in	Highlights
					VOC07	VOC12	COCO		
	ACCNN (Li et al. 2017b)	SS	VGG16	Fast RCNN	72.0 (07+12)	70.6 (07T+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 (Zhu et al. 2017a)	RPN	ResNet101	RFCN	**82.7** (07+12)	**80.4** (07T+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 (Chen and Gupta 2017)	RPN	VGG16	Faster RCNN	70.0 (07)	–	–	ICCV17	Model object-object relationships efficiently through a spatial memory network. Learn the functionality of NMS automatically
	ORN (Hu et al. 2018a)	RPN	ResNet101 +DCN	Faster RCNN	–	–	**39.0**	CVPR18	Model the relations of a set of object proposals through the interactions between their appearance features and geometry. Learn the functionality of NMS automatically
	SIN (Liu et al. 2018d)	RPN	VGG16	Faster RCNN	76.0 (07+12)	73.1 (07T+12)	23.2	CVPR18	Formulate object detection as graph-structured inference, where objects are graph nodes and relationships the edges

for each detection, and allows each candidate box to select a segment out of a large pool of object segmentation proposals and score the agreement between them. In Shrivastava and Gupta (2016), semantic segmentation was used as a form of contextual priming.

7.2 Local Context

Local context (Zhang et al. 2013; Galleguillos and Belongie 2010; Rabinovich et al. 2007) 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. Deep learning research that explicitly models object relations is quite limited, with representative ones being Spatial Memory Network (SMN) (Chen and Gupta 2017), Object Relation Network (Hu et al. 2018a), and Structure Inference Network (SIN) (Liu et al. 2018d). 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 (Vaswani et al. 2017), ORN processes a set of objects simultaneously through the interaction between their appearance feature and geometry. It does not require additional supervision, and it is easy to embed into existing networks, 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 (Liu et al. 2018d) considered two kinds of context: scene contextual information and object relationships within a single image. It formulates object detection as a problem of graph inference, where the objects are treated as nodes in a graph and relationships between objects are modeled as edges.

A wider range of methods has approached the context challenge with a simpler idea: enlarging the detection window size to extract some form of local context. Representative approaches include MRCNN (Gidaris and Komodakis 2015), Gated BiDirectional CNN (GBDNet) Zeng et al. (2016), Zeng et al. (2017), Attention to Context CNN (ACCNN) (Li et al. 2017b), CoupleNet (Zhu et al. 2017a), and Sermanet et al. (2013). In MRCNN (Gidaris and Komodakis 2015) (Fig. 18a), 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 by concatenation.

Quite a number of methods, all closely related to MRCNN, have been proposed since then. The method in Zagoruyko et al. (2016) 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. (2016), Zeng et al. (2017) proposed GBDNet (Fig. 18b) to extract features from multiscale contextualized regions surrounding an object proposal to improve detection performance. In contrast to the somewhat naive approach of learning CNN features for each region separately and then concatenating them, GBDNet passes messages among features from different contextual regions. Noting that message passing is not always helpful, but dependent on individual samples, Zeng et al. (2016) used gated functions to control message transmission. Li et al. (2017b) presented ACCNN (Fig. 18c) to utilize both global and local contextual information: the global context was captured using a Multiscale Local Contextualized (MLC) subnetwork, which recurrently generates an attention map for an input image to highlight promising contextual locations; local context adopted a method similar to that of MRCNN (Gidaris and Komodakis 2015). As shown in Fig. 18d, CoupleNet (Zhu et al. 2017a) is conceptually similar to ACCNN (Li et al. 2017b), but built upon RFCN (Dai et al. 2016c), which captures object information with position sensitive RoI pooling, CoupleNet added a branch to encode the global context 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 [SIFT (Lowe 2004), HOG (Dalal and Triggs 2005) and LBP (Ojala et al. 2002)], the most successful methods for object detection [e.g. DPM (Felzenszwalb et al. 2008)] used *sliding window* techniques (Viola and Jones 2001; Dalal and Triggs 2005; Felzenszwalb et al. 2008; Harzallah et al. 2009; Vedaldi et al. 2009). However, the number of windows is huge, growing with the number of pixels in an image, and the need to search at multiple scales and aspect ratios further increases the search space[12]. Therefore, it is computationally too expensive to apply sophisticated classifiers.

Around 2011, researchers proposed to relieve the tension between computational tractability and high detection qual-

[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.

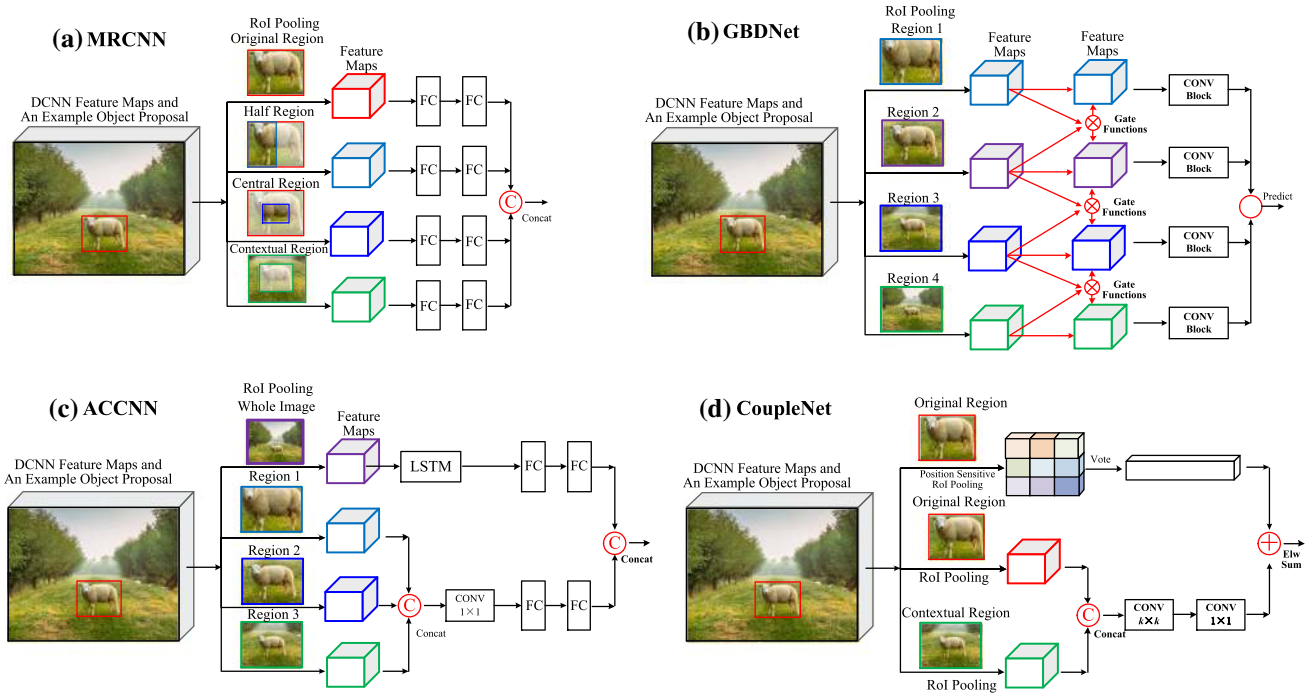

Fig. 18 Representative approaches that explore local surrounding contextual features: MRCNN (Gidaris and Komodakis 2015), GBDNet (Zeng et al. 2016, 2017), ACCNN (Li et al. 2017b) and CoupleNet (Zhu et al. 2017a); also see Table 8

ity by using *detection proposals*[13] (Van de Sande et al. 2011; Uijlings et al. 2013). Originating in the idea of *objectness* proposed by Alexe et al. (2010), object proposals are a set of candidate regions in an image that are likely to contain objects, and if high object recall can be achieved with a modest number of object proposals (like one hundred), significant speed-ups over the sliding window approach can be gained, allowing the use of more sophisticated classifiers. Detection proposals are usually used as a pre-processing step, limiting the number of regions that need to be evaluated by the detector, and should have the following characteristics:

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

The success of object detection based on detection proposals (Van de Sande et al. 2011; Uijlings et al. 2013) has attracted broad interest (Carreira and Sminchisescu 2012; Arbeláez et al. 2014; Alexe et al. 2012; Cheng et al. 2014; Zitnick and Dollár 2014; Endres and Hoiem 2010; Krähenbühl and Koltun 2014; Manen et al. 2013). A comprehensive review of object proposal algorithms is beyond the scope of this paper, because object proposals have applications beyond object detection (Arbeláez et al. 2012; Guillaumin et al. 2014; Zhu et al. 2017b). We refer interested readers to the recent surveys (Hosang et al. 2016; Chavali et al. 2016) which provide 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 (Van de Sande et al. 2011; Uijlings et al. 2013) and DCNN features (Krizhevsky et al. 2012a) led to the milestone RCNN (Girshick et al. 2014) in generic object detection. Since then, detection proposal has quickly become a standard preprocessing step, based on the fact that all winning entries in the PASCAL VOC (Everingham et al. 2010), ILSVRC (Russakovsky et al. 2015) and MS COCO (Lin et al. 2014) object detection challenges since 2014 used detection proposals (Girshick et al. 2014; Ouyang et al. 2015; Girshick 2015; Ren et al. 2015; Zeng et al. 2017; He et al. 2017).

Among object proposal approaches based on traditional low-level cues (e.g., color, texture, edge and gradients), Selective Search (Uijlings et al. 2013), MCG (Arbeláez et al. 2014) and EdgeBoxes (Zitnick and Dollár 2014) are among the more popular. As the domain rapidly progressed, traditional object proposal approaches (Uijlings et al. 2013; Hosang et al. 2016; Zitnick and Dollár 2014), which were adopted as external modules independent of the detectors,

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

became the speed bottleneck of the detection pipeline (Ren et al. 2015). An emerging class of object proposal algorithms (Erhan et al. 2014; Ren et al. 2015; Kuo et al. 2015; Ghodrati et al. 2015; Pinheiro et al. 2015; Yang et al. 2016a) 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 are best exemplified by the RPC method of Ren et al. (2015), illustrated in Fig. 19. RPN predicts object proposals by sliding a small network over the feature map of the last shared CONV layer. At each sliding window location, k proposals are predicted 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. (2015) proposed integrating RPN and Fast RCNN into a single network by sharing their convolutional layers, leading to Faster RCNN, the first end-to-end detection pipeline. 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.

Instead of fixing *a priori* a set of anchors as MultiBox (Erhan et al. 2014; Szegedy et al. 2014) and RPN (Ren et al. 2015), Lu et al. (2016) proposed generating anchor locations by using a recursive search strategy which can adaptively guide computational resources to focus on sub-regions 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 by an Adjacency and Zoom Network (AZNet), which extends RPN by adding a branch to compute the scalar zoom indicator in parallel with the existing branch.

Further work attempts to generate object proposals by exploiting multilayer convolutional features. Concurrent with RPN (Ren et al. 2015), Ghodrati et al. (2015) 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 (Kong et al. 2016) 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. (2016a) proposed CRAFT 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. (2018a) proposed ZIP to improve RPN by predicting object proposals with multiple convolutional feature maps at different network depths to integrate both low level details and high level semantics. The backbone used in ZIP is a "zoom out and in" network inspired by the conv and deconv structure (Long et al. 2015).

Finally, recent work which deserves mention includes Deepbox (Kuo et al. 2015), which proposed a lightweight CNN to learn to rerank proposals generated by EdgeBox, and DeNet (TychsenSmith and Petersson 2017) which introduces bounding box corner estimation to predict object proposals efficiently to replace RPN in a Faster RCNN style detector.

Object Segment Proposal Methods Pinheiro et al. (2015), Pinheiro et al. (2016) 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 (Hariharan et al. 2014; Dai et al. 2016b; Li et al. 2017e). In addition, using instance segmentation supervision can improve the performance of bounding box object detection. The pioneering work of DeepMask, proposed by Pinheiro et al. (2015), segments proposals learnt directly from raw image data with a deep network. Similarly to RPN, after a number of shared convolutional layers DeepMask splits the network into two branches in order to predict a class agnostic mask and an associated objectness score. Also similar to the efficient sliding window strategy in OverFeat (Sermanet et al. 2014), 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. (2016) proposed SharpMask by augmenting the DeepMask architecture with a refinement module, similar to the architectures shown in Fig. 17 (b1) and (b2), augmenting the feed-forward network with a top-down refinement process. SharpMask can efficiently integrate spatially rich information from early features with strong semantic information encoded in later layers to generate high fidelity object masks.

Motivated by Fully Convolutional Networks (FCN) for semantic segmentation (Long et al. 2015) and DeepMask (Pinheiro et al. 2015; Dai et al. 2016a) proposed InstanceFCN to generate instance segment proposals. Similar to DeepMask, the InstanceFCN network is split into two fully convolutional branches, one to generate instance sensitive score maps, the other to predict the objectness score. Hu et al. (2017) proposed FastMask to efficiently generate instance segment proposals in a one-shot manner, similar to SSD (Liu et al. 2016), in order to make use of multiscale convolutional features. Sliding windows extracted densely from multiscale convolutional feature maps were input to a scale-tolerant attentional head module in order to predict segmentation masks and objectness scores. FastMask is claimed to run at 13 FPS on 800 $\times$ 600 images.

[14] The concept of "anchor" first appeared in Ren et al. (2015).

Table 9 Summary of object proposal methods using DCNN. Bold values indicates the number of object proposals

Proposer name	Backbone network	Detector tested	Recall@IoU (VOC07)			Detection results (mAP)			Published in	Highlights
			0.5	0.7	0.9	VOC07	VOC12	COCO		
Bounding box object proposal methods										
MultiBox1 (Erhan et al. 2014)	AlexNet	RCNN	–	–	–	29.0 (**10**) (12)	–	–	CVPR14	Learns a class agnostic regressor on a small set of 800 predefined anchor boxes. Do not share features for detection
DeepBox (Kuo et al. 2015)	VGG16	Fast RCNN	0.96 (**1000**)	0.84 (**1000**)	0.15 (**1000**)	–	–	37.8 (**500**) (IoU@0.5)	ICCV15	Use a lightweight CNN to learn to rerank proposals generated by EdgeBox. Can run at 0.26s per image. Do not share features for detection
RPN (Ren et al. 2015, 2017)	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 detection. Most widely used object proposal method. Significant improvements in detection speed
DeepProposal (Ghodrati et al. 2015)	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 (Yang et al. 2016a)	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

Table 9 continued

Proposer name	Backbone network	Detector tested	Recall@IoU (VOC07)			Detection results (mAP)			Published in	Highlights
			0.5	0.7	0.9	VOC07	VOC12	COCO		
AZNet (Lu et al. 2016)	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 regions, then recursively search for subregions that may contain objects. Adaptively guide computational resources to focus on likely subregions
ZIP (Li et al. 2018a)	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 (TychsenSmith and Petersson 2017)	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 predicting object proposals efficiently to replace RPN; Does not require predefined anchors

Table 9 continued

Proposer name	Backbone network	Detector tested	Box proposals (AR, COCO)	Segment proposals (AR, COCO)	Published in	Highlights
Segment proposal methods						
DeepMask (Pinheiro et al. 2015)	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 (Dai et al. 2016a)	VGG16	–	–	0.32 (**100**), 0.39 (**1000**)	ECCV16	Combines ideas of FCN (Long et al. 2015) and DeepMask (Pinheiro et al. 2015). Introduces instance sensitive score maps. Needs segmentation annotations to train the network
SharpMask (Pinheiro et al. 2016)	MPN (Zagoruyko et al. 2016)	Fast RCNN	0.39 (**100**), 0.53 (**1000**)	0.30 (**100**), 0.39 (**1000**)	ECCV16	Leverages features at multiple convolutional layers by introducing a top-down refinement module. Does not share features with detection network. Needs segmentation annotations for training
FastMask (Hu et al. 2017)	ResNet39	–	0.43 (**100**), 0.57 (**1000**)	0.32 (**100**), 0.41 (**1000**)	CVPR17	Generates instance segment proposals efficiently in one-shot manner similar to SSD (Liu et al. 2016). Uses multiscale convolutional features. Uses segmentation annotations for training

The detection results on COCO are based on mAP@IoU[0.5, 0.95], unless stated otherwise

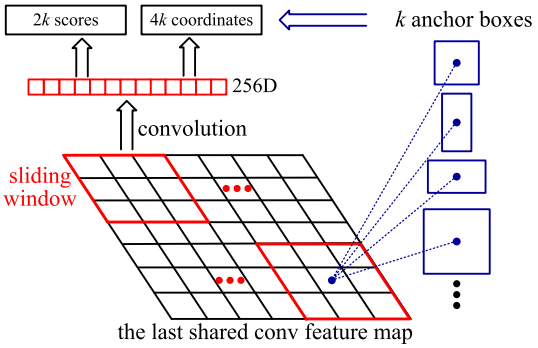

Fig. 19 Illustration of the region proposal network (RPN) introduced in Ren et al. (2015)

9 Other Issues

Data Augmentation Performing data augmentation for learning DCNNs (Chatfield et al. 2014; Girshick 2015; Girshick et al. 2014) 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, color perturbations, and adding noise. By artificially enlarging the number of samples, data augmentation helps in reducing overfitting and improving generalization. It can be used at training time, at test time, or both. Nevertheless, it has the obvious limitation that the time required for training increases significantly. Data augmentation may synthesize completely new training images (Peng et al. 2015; Wang et al. 2017), however it is hard to guarantee that the synthetic images generalize well to real ones. Some researchers (Dwibedi et al. 2017; Gupta et al. 2016) proposed augmenting datasets by pasting real segmented objects into natural images; indeed, Dvornik et al. (2018) showed that appropriately modeling the visual context surrounding objects is crucial to place them in the right environment, and proposed a context model to automatically find appropriate locations on images to place new objects for data augmentation.

Novel Training Strategies Detecting objects under a wide range of scale variations, especially the detection of very small objects, stands out as a key challenge. It has been shown (Huang et al. 2017b; Liu et al. 2016) that image resolution has a considerable impact on detection accuracy, therefore scaling is particularly commonly used in data augmentation, since higher resolutions increase the possibility of detecting small objects (Huang et al. 2017b). Recently, Singh et al. proposed advanced and efficient data argumentation methods SNIP (Singh and Davis 2018) and SNIPER (Singh et al. 2018b) to 1 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, SNIP introduces a novel training scheme that can reduce scale variations during training, but without reducing training samples; SNIPER allows for efficient multiscale training, only processing context regions around ground truth objects at the appropriate scale, instead of processing a whole image pyramid. Peng et al. (2018) studied a key factor in training, the minibatch size, and proposed MegDet, a Large MiniBatch Object Detector, to enable the training with a much larger minibatch size than before (from 16 to 256). To avoid the failure of convergence and significantly speed up the training process, Peng et al. (2018) proposed a learning rate policy and Cross GPU Batch Normalization, and effectively utilized 128 GPUs, allowing MegDet to finish COCO training in 4 hours on 128 GPUs, and winning the COCO 2017 Detection Challenge.

Reducing Localization Error In object detection, the Intersection Over Union[15] (IOU) between a detected bounding box and its ground truth box is the most popular evaluation metric, and an IOU threshold (e.g. typical value of 0.5) is required to define positives and negatives. From Fig. 13, in most state of the art detectors (Girshick 2015; Liu et al. 2016; He et al. 2017; Ren et al. 2015; Redmon et al. 2016) object detection is formulated as a multitask learning problem, i.e., jointly optimizing a softmax classifier which assigns object proposals with class labels and bounding box regressors, localizing objects by maximizing IOU or other metrics between detection results and ground truth. Bounding boxes are only a crude approximation for articulated objects, consequently background pixels are almost invariably included in a bounding box, which affects the accuracy of classification and localization. The study in Hoiem et al. (2012) shows that object localization error is one of the most influential forms of error, in addition to confusion between similar objects. Localization error could stem from insufficient overlap (smaller than the required IOU threshold, such as the green box in Fig. 20) or duplicate detections (i.e., multiple overlapping detections for an object instance). Usually, some post-processing step like NonMaximum Suppression (NMS) (Bodla et al. 2017; Hosang et al. 2017) is used for eliminating duplicate detections. However, due to misalignments the bounding box with better localization could be suppressed during NMS, leading to poorer localization quality (such as the purple box shown in Fig. 20). Therefore, there are quite a few methods aiming at improving detection performance by reducing localization error.

MRCNN (Gidaris and Komodakis 2015) introduces iterative bounding box regression, where an RCNN is applied several times. CRAFT (Yang et al. 2016a) and AttractioNet (Gidaris and Komodakis 2016) use a multi-stage detection sub-network to generate accurate proposals, to forward to Fast RCNN. Cai and Vasconcelos (2018) proposed Cascade RCNN, a multistage extension of RCNN, in which a sequence of detectors is trained sequentially with increasing

[15] Please refer to Sect. 4.2 for more details on the definition of IOU.

Table 10 Representative methods for training strategies and class imbalance handling

Detector name	Region proposal	Backbone DCNN	Pipelined used	VOC07 results	VOC12 results	COCO results	Published in	Highlights
MegDet (Peng et al. 2018)	RPN	ResNet50+FPN	FasterRCNN	–	–	52.5	CVPR18	Allow training with much larger minibatch size 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 (Singh et al. 2018b)	RPN	DPN (Chen et al. 2017b) +DCN (Dai et al. 2017)	RFCN	–	–	48.3	CVPR18	A new multiscale training scheme. Empirically examined the effect of up-sampling for small object detection. During training, only select objects that fit the scale of features as positive samples
SNIPER (Singh et al. 2018b)	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 (Shrivastava et al. 2016)	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

Table 10 continued

Detector name	Region proposal	Backbone DCNN	Pipelined used	VOC07 results	VOC12 results	COCO results	Published in	Highlights
FactorNet (Ouyang et al. 2016)	SS	GooglNet	RCNN	–	–	–	CVPR16	Identify the imbalance in the number of samples for different object categories; propose a divide-and-conquer feature learning scheme
Chained Cascade (Cai and Vasconcelos 2018)	SS CRAFT	VGG Inceptionv2	Fast RCNN, Faster RCNN	80.4 (07+12) (SS+VGG)	–	–	ICCV17	Jointly learn DCNN and multiple stages of cascaded classifiers. Boost detection accuracy on PASCAL VOC 2007 and ImageNet for both fast RCNN and Faster RCNN using different region proposal methods
Cascade RCNN (Cai and Vasconcelos 2018)	RPN	VGG ResNet101 +FPN	Faster RCNN	–	–	42.8	CVPR18	Jointly learn DCNN and multiple stages of cascaded classifiers, which are learned using different localization accuracy for selecting positive samples. Stack bounding box regression at multiple stages
RetinaNet (Lin et al. 2017b)	–	ResNet101 +FPN	RetinaNet	–	–	39.1	ICCV17	Propose a novel Focal Loss which focuses training on hard examples. Handles well the problem of imbalance of positive and negative samples when training a one-stage detector

Results on COCO are reported with Test Dev. The detection results on COCO are based on mAP@IoU[0.5, 0.95]

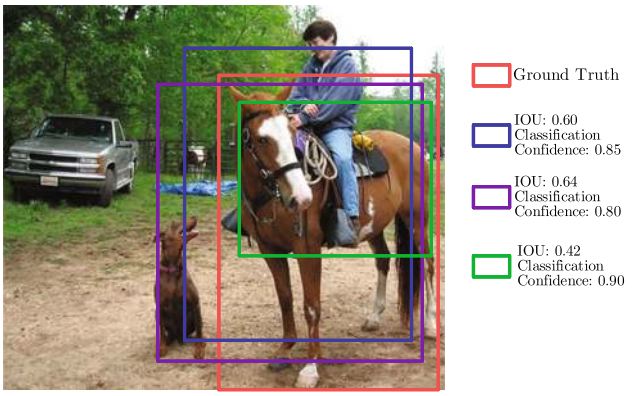

Fig. 20 Localization error could stem from insufficient overlap or duplicate detections. Localization error is a frequent cause of false positives (Color figure online)

IOU thresholds, based on the observation that the output of a detector trained with a certain IOU 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. This approach can be built with any RCNN-based detector, and is demonstrated to achieve consistent gains (about 2 to 4 points) independent of the baseline detector strength, at a marginal increase in computation. There is also recent work (Jiang et al. 2018; Rezatofighi et al. 2019; Huang et al. 2019) formulating IOU directly as the optimization objective, and in proposing improved NMS results (Bodla et al. 2017; He et al. 2019; Hosang et al. 2017; TychsenSmith and Petersson 2018), such as Soft NMS (Bodla et al. 2017) and learning NMS (Hosang et al. 2017).

Class Imbalance Handling Unlike image classification, object detection has another unique problem: 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, however this imbalance can make the training very inefficient, and the large number of easy negatives tends to overwhelm the training. In the past, this issue has typically been addressed via techniques such as bootstrapping (Sung and Poggio 1994). More recently, this problem has also seen some attention (Li et al. 2019a; Lin et al. 2017b; Shrivastava et al. 2016). 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 (Girshick et al. 2014; Girshick 2015; Ren et al. 2015; He et al. 2017), although example mining approaches, such as Online Hard Example Mining (OHEM) (Shrivastava et al. 2016), may be used to maintain a reasonable balance between foreground and background. In the case of one-stage object detectors (Redmon et al. 2016; Liu et al. 2016), this imbalance is extremely serious (e.g. 100,000 background examples to every object). Lin et al. (2017b) proposed Focal Loss to address this by rectifying the Cross Entropy loss, such that it down-weights the loss assigned to correctly classified examples. Li et al. (2019a) 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 developments in 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. We conclude this review with a discussion of the state of the art in Sect. 10.1, an overall discussion of key issues in Sect. 10.2, and finally suggested future research directions in Sect. 10.3.

10.1 State of the Art Performance

A large variety of detectors has appeared in the last few years, and the introduction of standard benchmarks, such as PASCAL VOC (Everingham et al. 2010, 2015), ImageNet (Russakovsky et al. 2015) and COCO (Lin et al. 2014), has made it easier to compare detectors. As can be seen from our earlier discussion in Sects. 5–9, it may be misleading 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 choices:

- Meta detection frameworks, such as RCNN (Girshick et al. 2014), Fast RCNN (Girshick 2015), Faster RCNN (Ren et al. 2015), RFCN (Dai et al. 2016c), Mask RCNN (He et al. 2017), YOLO (Redmon et al. 2016) and SSD (Liu et al. 2016);
- Backbone networks such as VGG (Simonyan and Zisserman 2015), Inception (Szegedy et al. 2015; Ioffe and Szegedy 2015; Szegedy et al. 2016), ResNet (He et al. 2016), ResNeXt (Xie et al. 2017), and Xception (Chollet 2017) *etc.* listed in Table 6;
- Innovations such as multilayer feature combination (Lin et al. 2017a; Shrivastava et al. 2017; Fu et al. 2017), deformable convolutional networks (Dai et al. 2017), deformable RoI pooling (Ouyang et al. 2015; Dai et al. 2017), heavier heads (Ren et al. 2016; Peng et al. 2018), and lighter heads (Li et al. 2018c);

- Pretraining with datasets such as ImageNet (Russakovsky et al. 2015), COCO (Lin et al. 2014), Places (Zhou et al. 2017a), JFT (Hinton et al. 2015) and Open Images (Krasin et al. 2017);
- Different detection proposal methods and different numbers of object proposals;
- Train/test data augmentation, novel multiscale training strategies (Singh and Davis 2018; Singh et al. 2018b) etc, and model ensembling.

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 (Huang et al. 2017b) of the three main families of detectors [Faster RCNN (Ren et al. 2015), RFCN (Dai et al. 2016c) and SSD (Liu et al. 2016)] by varying the backbone network, image resolution, and the number of box proposals.

As can be seen from Tables 7, 8, 9, 10, 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.

Figures 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 (ILSVRC 2018; MS COCO 2018; PASCAL VOC 2018). 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 (Girshick 2015), Faster RCNN (Ren et al. 2015), FPN (Lin et al. 2017a), Deformable RCNN (Dai et al. 2017), and Cascade RCNN (Cai and Vasconcelos 2018). In summary, the backbone network, the detection framework, and the availability of large scale datasets are the three most important factors in detection accuracy. Ensembles of multiple models, the incorporation of context features, and data augmentation all help to achieve better accuracy.

In less than 5 years, since AlexNet (Krizhevsky et al. 2012a) was proposed, the Top5 error on ImageNet classification (Russakovsky et al. 2015) with 1000 classes has dropped from 16% to 2%, as shown in Fig. 15. However, the mAP of the best performing detector (Peng et al. 2018) on COCO (Lin et al. 2014), trained to detect only 80 classes, is only at 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 far from satisfies 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 discussed throughout this paper, we would now like to focus on the key factors which have emerged in generic object detection based on deep learning.

(1) Detection frameworks: two stage versus one stage

In Sect. 5 we identified two major categories of detection frameworks: region based (two stage) and unified (one stage):

- When large computational cost is allowed, two-stage detectors generally produce higher detection accuracies than one-stage, evidenced by the fact that most winning approaches used in famous detection challenges like are predominantly based on two-stage frameworks, because their structure is more flexible and better suited for region based classification. The most widely used frameworks are Faster RCNN (Ren et al. 2015), RFCN (Dai et al. 2016c) and Mask RCNN (He et al. 2017).
- It has been shown in Huang et al. (2017b) that the detection accuracy of one-stage SSD (Liu et al. 2016) is less sensitive to the quality of the backbone network than representative two-stage frameworks.
- One-stage detectors like YOLO (Redmon et al. 2016) and SSD (Liu et al. 2016) are generally faster than two-stage ones, because of avoiding preprocessing algorithms, using lightweight backbone networks, performing prediction with fewer candidate regions, and making the classification subnetwork fully convolutional. 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) (Law and Deng 2018; Ren et al. 2015).
- It has been shown (Huang et al. 2017b; Redmon et al. 2016; Liu et al. 2016) that one-stage frameworks like YOLO and SSD typically have much poorer performance when detecting small objects than two-stage architectures like Faster RCNN and RFCN, but are competitive in detecting large objects.

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:

- A fully convolutional pipeline
- Exploring complementary information from other correlated tasks, e.g., Mask RCNN (He et al. 2017)
- Sliding windows (Ren et al. 2015)

Table 11 Summary of properties and performance of milestone detection frameworks for generic object detection

Detector name	RP	Backbone DCNN	Input ImgSize	VOC07 results	VOC12 results	Speed (FPS)	Published in	Source code	Highlights and Disadvantages
Region based (*Sect.* 5.1)									
RCNN (Girshick et al. 2014)	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 artP **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 (He et al. 2014)	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 **Disadvantages:** Inherit disadvantages of RCNN; Does not result in much training speedup; Fine-tuning not able to update the CONV layers before SPP layer
Fast RCNN (Girshick 2015)	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 (ignoring RP generation); Design a RoI pooling 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

Table 11 continued

Detector name	RP	Backbone DCNN	Input ImgSize	VOC07 results	VOC12 results	Speed (FPS)	Published in	Source code	Highlights and Disadvantages
Faster RCNN (Ren et al. 2015)	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 falls short of real time
RCNN⊖R (Lenc and Vedaldi 2015)	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:** Falls short of real time; Decreased accuracy from poor RPs
RFCN (Dai et al. 2016c)	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; 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 falls short of real time

Table 11 continued

Detector name	RP	Backbone DCNN	Input ImgSize	VOC07 results	VOC12 results	Speed (FPS)	Published in	Source code	Highlights and Disadvantages
Mask RCNN (He et al. 2017)	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; Outstanding performance **Disadvantages:** Falls short of real time applications
Unified (*Sect.* 5.2)									
OverFeat (Sermanet et al. 2014)	–	AlexNet like	Arbitrary	–	–	< 0.1	ICLR14	c++	**Highlights:** Convolutional feature sharing; Multiscale image pyramid CNN feature extraction; Won the ISLVRC2013 localization competition; Significantly faster than RCNN **Disadvantages:** Multi-stage pipeline sequentially trained; Single bounding box regressor; Cannot handle multiple object instances of the same class; Too slow for real time applications
YOLO (Redmon et al. 2016)	–	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, Fast YOLO at 155 FPS; **Disadvantages:** Accuracy falls far behind state of the art detectors; Struggle to localize small objects

Table 11 continued

Detector name	RP	Backbone DCNN	Input ImgSize	VOC07 results	VOC12 results	Speed (FPS)	Published in	Source code	Highlights and Disadvantages
YOLOv2 (Redmon and Farhadi 2017)	–	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 (Liu et al. 2016)	–	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 multi-scale CONV layers; Faster and significantly more accurate than YOLO; Can run at 59 FPS; **Disadvantages:** Not good at detecting small objects

See Sect. 5 for a detailed discussion. Some architectures are illustrated in Fig. 13. The properties of the backbone DCNNs can be found in Table 6
Training data: "07"←VOC2007 trainval; "07T"←VOC2007 trainval and test; "12"←VOC2012 trainval; "CO"←COCO trainval. The "Speed" column roughly estimates the detection speed with a single Nvidia Titan X GPU
RP region proposal; *SS* selective search; *RPN* region proposal network; $RCNN \ominus R$ RCNN minus R and used a trivial RP method

- Fusing information from different layers of the backbone.

The evidence from recent success of cascade for object detection (Cai and Vasconcelos 2018; Cheng et al. 2018a, b) and instance segmentation on COCO (Chen et al. 2019a) and other challenges has shown that multistage object detection could be a future framework for a speed-accuracy trade-off. A teaser investigation is being done in the 2019 WIDER Challenge (Loy et al. 2019).

(2) Backbone networks

As discussed in Sect. 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. Generally, deeper backbones such as ResNet (He et al. 2016), ResNeXt (Xie et al. 2017), InceptionResNet (Szegedy et al. 2017) perform better; however, they are computationally more expensive and require much more data and massive computing for training. Some backbones (Howard et al. 2017; Iandola et al. 2016; Zhang et al. 2018c) were proposed for focusing on speed instead, such as MobileNet (Howard et al. 2017) which has been shown to achieve VGGNet16 accuracy on ImageNet with only $\frac{1}{30}$ the computational cost and model size. Backbone training from scratch may become possible as more training data and better training strategies are available (Wu and He 2018; Luo et al. 2018, 2019).

(3) Improving the robustness of object representation

The variation of real world images is a key challenge in object recognition. The variations include lighting, pose, deformations, background clutter, occlusions, blur, resolution, noise, and camera distortions.

(3.1) Object scale and small object size

Large variations of object scale, particularly those of small objects, pose a great challenge. Here a summary and discussion on the main strategies identified in Sect. 6.2:

- Using image pyramids: They are simple and effective, helping to enlarge small objects and to shrink large ones. They are computationally expensive, but are nevertheless commonly used during inference for better accuracy.
- Using features from convolutional layers of different resolutions: In early work like SSD (Liu et al. 2016), 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 (Lin et al. 2017a).

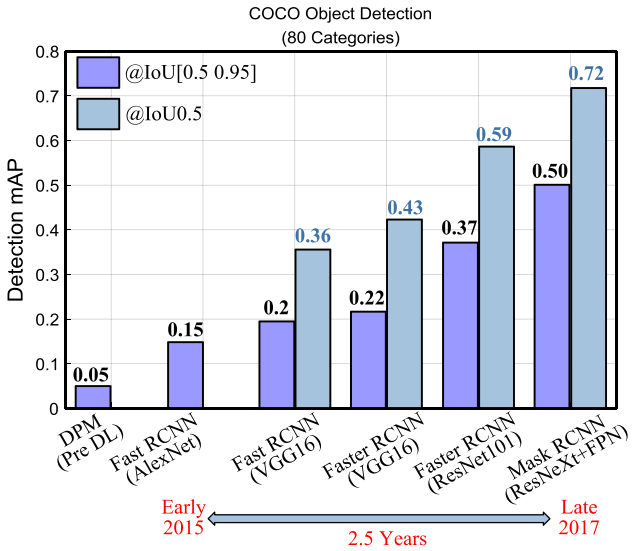

Fig. 21 Evolution of object detection performance on COCO (Test-Dev results). Results are quoted from (Girshick 2015; He et al. 2017; Ren et al. 2017). The backbone network, the design of detection framework and the availability of good and large scale datasets are the three most important factors in detection accuracy

- Using dilated convolutions (Li et al. 2018b, 2019b): A simple and effective method to incorporate broader context and maintain high resolution feature maps.
- Using anchor boxes of different scales and aspect ratios: Drawbacks of having many parameters, and scales and aspect ratios of anchor boxes are usually heuristically determined.
- Up-scaling: Particularly for the detection of small objects, high-resolution networks (Sun et al. 2019a, b) can be developed. It remains unclear whether super-resolution techniques improve detection accuracy or not.

Despite recent advances, the detection accuracy for small objects is still much lower than that of larger ones. Therefore, the detection of small objects remains one of the key challenges in object detection. Perhaps localization requirements need to be generalized as a function of scale, since certain applications, e.g. autonomous driving, only require the identification of the existence of small objects within a larger region, and exact localization is not necessary.

(3.2) Deformation, occlusion, and other factors

As discussed in Sect. 2.2, there are approaches to handling geometric transformation, occlusions, and deformation mainly based on two paradigms. The first is a spatial transformer network, which uses regression to obtain a deformation field and then warp features according to the deformation field (Dai et al. 2017). The second is based on a deformable part-based model (Felzenszwalb et al. 2010b),

which finds the maximum response to a part filter with spatial constraints taken into consideration (Ouyang et al. 2015; Girshick et al. 2015; Wan et al. 2015).

Rotation invariance may be attractive in certain applications, but there are limited generic object detection work focusing on rotation invariance, because popular benchmark detection datasets (PASCAL VOC, ImageNet, COCO) do not have large variations in rotation. Occlusion handling is intensively studied in face detection and pedestrian detection, but very little work has been devoted to occlusion handling for generic object detection. In general, despite recent advances, deep networks are still limited by the lack of robustness to a number of variations, which significantly constrains their real-world applications.

(4) Context reasoning

As introduced in Sect. 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 (Oliva and Torralba 2007), especially for small objects, occluded objects, and with poor image quality. There was extensive work preceding deep learning (Malisiewicz and Efros 2009; Murphy et al. 2003; Rabinovich et al. 2007; Divvala et al. 2009; Galleguillos and Belongie 2010), and also quite a few works in the era of deep learning (Gidaris and Komodakis 2015; Zeng et al. 2016, 2017; Chen and Gupta 2017; Hu et al. 2018a). How to efficiently and effectively incorporate contextual information remains to be explored, possibly guided by how human vision uses context, based on scene graphs (Li et al. 2017d), or via the full segmentation of objects and scenes using panoptic segmentation (Kirillov et al. 2018).

(5) Detection proposals

Detection proposals significantly reduce search spaces. As recommended in Hosang et al. (2016), future detection proposals will surely have to improve in repeatability, recall, localization accuracy, and speed. Since the success of RPN (Ren et al. 2015), which integrated proposal generation and detection into a common framework, CNN based detection proposal generation methods have dominated region proposal. It is recommended that new detection proposals should be assessed for object detection, instead of evaluating detection proposals alone.

(6) Other factors

As discussed in Sect. 9, there are many other factors affecting object detection quality: data augmentation, novel training strategies, combinations of backbone models, 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.

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 yet satisfactorily address real-world challenges like those of Sect. 2.2. We see a number of long-standing challenges:

- Working in an open world: being robust to any number of environmental changes, being able to evolve or adapt.
- Object detection under constrained conditions: learning from weakly labeled data or few bounding box annotations, wearable devices, unseen object categories etc.
- Object detection in other modalities: 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. Object detection algorithms are unable, in general, to recognize object categories outside of their training dataset, although ideally there should be the ability to recognize novel object categories (Lake et al. 2015; Hariharan and Girshick 2017). Current detection datasets (Everingham et al. 2010; Russakovsky et al. 2015; Lin et al. 2014) contain only a few dozen to hundreds of categories, significantly fewer than those which can be recognized by humans. New larger-scale datasets (Hoffman et al. 2014; Singh et al. 2018a; Redmon and Farhadi 2017) 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 (Girshick et al. 2014), Fast RCNN (Girshick 2015), Faster RCNN (Ren et al. 2015), Mask RCNN (He et al. 2017)] and one-stage detectors [YOLO (Redmon et al. 2016), SSD (Liu et al. 2016)]. Region-based detectors have higher accuracy, one-stage detectors are generally faster and simpler. Object detectors depend heavily on the underlying backbone networks, which have been optimized for image classification, possibly causing a learning bias; 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 [AlexNet (Krizhevsky et al. 2012b)] to hundreds of layers [ResNet (He et al. 2016), DenseNet (Huang et al. 2017a)]. These networks have millions to hundreds of millions of parameters, requiring massive data and GPUs for training. In order reduce or remove network redundancy, there has been growing research interest in designing compact and lightweight networks (Chen et al. 2017a; Alvarez and Salzmann 2016; Huang et al. 2018; Howard et al. 2017; Lin et al. 2017c; Yu et al. 2018) and network acceleration (Cheng et al. 2018c; Hubara et al. 2016; Han et al. 2016; Li et al. 2017a, c; Wei et al. 2018).

(4) *Automatic Neural Architecture Search* Deep learning bypasses manual feature engineering which requires human experts with strong domain knowledge, however DCNNs require similarly significant expertise. It is natural to consider automated design of detection backbone architectures, such as the recent Automated Machine Learning (AutoML) (Quanming et al. 2018), which has been applied to image classification and object detection (Cai et al. 2018; Chen et al. 2019c; Ghiasi et al. 2019; Liu et al. 2018a; Zoph and Le 2016; Zoph et al. 2018).

(5) *Object Instance Segmentation* For a richer and more detailed understanding of image content, there is a need to tackle pixel-level object instance segmentation (Lin et al. 2014; He et al. 2017; Hu et al. 2018c), which can play an important role in potential applications that require the precise boundaries of individual objects.

(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 (Everingham et al. 2015; Lin et al. 2014; Russakovsky et al. 2015; Lin et al. 2014). However, fully supervised learning has serious limitations, particularly where the collection of bounding box annotations is labor intensive and where the number of images is large. Fully supervised learning is not scalable in the absence of fully labeled training data, so it is essential to understand how the power of CNNs can be leveraged where only weakly / partially annotated data are provided (Bilen and Vedaldi 2016; Diba et al. 2017; Shi et al. 2017).

(7) *Few / Zero 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. In contrast, humans (even children) can learn a visual concept quickly from very few given examples and can often generalize well (Biederman 1987b; Lake et al. 2015; FeiFei et al. 2006). Therefore, the ability to learn from only few examples, *few* shot detection, is very appealing (Chen et al. 2018a; Dong et al. 2018; Finn et al. 2017; Kang et al. 2018; Lake et al. 2015; Ren et al. 2018; Schwartz et al. 2019). Even more constrained, *zero* shot object detection localizes and recognizes object classes that have never been seen[16] before (Bansal et al. 2018; Demirel et al. 2018; Rahman et al. 2018b, a), essential for life-long learning machines that need to intelligently and incrementally discover new object categories.

(8) *Object Detection in Other Modalities* Most detectors are based on still 2D images; object detection in other modalities can be highly relevant in domains such as autonomous vehicles, unmanned aerial vehicles, and robotics. These modalities raise new challenges in effectively using depth (Chen et al. 2015c; Pepik et al. 2015; Xiang et al. 2014; Wu et al. 2015), video (Feichtenhofer et al. 2017; Kang et al. 2016), and point clouds (Qi et al. 2017, 2018).

(9) *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, and medical CT images (Rebuffi et al. 2017, 2018). Most such research focuses on image classification, rarely targeting object detection (Wang et al. 2019), and developed detectors are usually domain specific. Object detection independent of image domain and cross-domain object detection represent important future directions.

The research field of generic object detection is still far from complete. However given the breakthroughs over the past 5 years we are optimistic of future developments and opportunities.

Acknowledgements Open access funding provided by University of Oulu including Oulu University Hospital. The authors would like to thank the pioneering 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 anonymous 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.

[16] Although side information may be provided, such as a wikipedia page or an attributes vector.

References

Agrawal, P., Girshick, R., & Malik, J. (2014). Analyzing the performance of multilayer neural networks for object recognition. In *ECCV* (pp. 329–344).

Alexe, B., Deselaers, T., & Ferrari, V. (2010). What is an object? In *CVPR* (pp. 73–80).

Alexe, B., Deselaers, T., & Ferrari, V. (2012). Measuring the objectness of image windows. *IEEE TPAMI*, *34*(11), 2189–2202.

Alvarez, J., & Salzmann, M. (2016). Learning the number of neurons in deep networks. In *NIPS* (pp. 2270–2278).

Andreopoulos, A., & Tsotsos, J. (2013). 50 years of object recognition: Directions forward. *Computer Vision and Image Understanding*, *117*(8), 827–891.

Arbeláez, P., Hariharan, B., Gu, C., Gupta, S., Bourdev, L., & Malik, J. (2012). Semantic segmentation using regions and parts. In *CVPR* (pp. 3378–3385).

Arbeláez, P., Pont-Tuset, J., Barron, J., Marques, F., & Malik, J. (2014). Multiscale combinatorial grouping. In *CVPR* (pp. 328–335).

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.

Bansal, A., Sikka, K., Sharma, G., Chellappa, R., & Divakaran, A. (2018). Zero shot object detection. In *ECCV*.

Bar, M. (2004). Visual objects in context. *Nature Reviews Neuroscience*, *5*(8), 617–629.

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).

Belongie, S., Malik, J., & Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE TPAMI*, *24*(4), 509–522.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE TPAMI*, *35*(8), 1798–1828.

Biederman, I. (1972). *Perceiving real world scenes. IJCV*, *177*(7), 77–80.

Biederman, I. (1987a). Recognition by components: A theory of human image understanding. *Psychological Review*, *94*(2), 115.

Biederman, I. (1987b). Recognition by components: A theory of human image understanding. *Psychological Review*, *94*(2), 115.

Bilen, H., & Vedaldi, A. (2016). Weakly supervised deep detection networks. In *CVPR* (pp. 2846–2854).

Bodla, N., Singh, B., Chellappa, R., & Davis L. S. (2017). SoftNMS improving object detection with one line of code. In *ICCV* (pp. 5562–5570).

Borji, A., Cheng, M., Jiang, H., & Li, J. (2014). Salient object detection: A survey, 1, 1–26. arXiv:1411.5878v1.

Bourdev, L., & Brandt, J. (2005). Robust object detection via soft cascade. *CVPR*, *2*, 236–243.

Bruna, J., & Mallat, S. (2013). *Invariant scattering convolution networks. IEEE TPAMI*, *35*(8), 1872–1886.

Cai, Z., & Vasconcelos, N. (2018). Cascade RCNN: Delving into high quality object detection. In *CVPR*.

Cai, Z., Fan, Q., Feris, R., & Vasconcelos, N. (2016). A unified multi-scale deep convolutional neural network for fast object detection. In *ECCV* (pp. 354–370).

Cai, H., Yang, J., Zhang, W., Han, S., & Yu, Y. et al. (2018) Path-level network transformation for efficient architecture search. In *ICML*.

Carreira, J., & Sminchisescu, C. (2012). CMPC: Automatic object segmentation using constrained parametric mincuts. *IEEE TPAMI*, *34*(7), 1312–1328.

Chatfield, K., Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*.

Chavali, N., Agrawal, H., Mahendru, A., & Batra, D. (2016). Object proposal evaluation protocol is gameable. In *CVPR* (pp. 835–844).

Chellappa, R. (2016). The changing fortunes of pattern recognition and computer vision. *Image and Vision Computing*, *55*, 3–5.

Chen, G., Choi, W., Yu, X., Han, T., & Chandraker M. (2017a). Learning efficient object detection models with knowledge distillation. In *NIPS*.

Chen, H., Wang, Y., Wang, G., & Qiao, Y. (2018a). LSTD: A low shot transfer detector for object detection. In *AAAI*.

Chen, K., Pang, J., Wang, J., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Shi, J., Ouyang, W., et al. (2019a). Hybrid task cascade for instance segmentation. In *CVPR*.

Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. (2015a), Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *ICLR*.

Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. (2018b). DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE TPAMI*, *40*(4), 834–848.

Chen, Q., Song, Z., Dong, J., Huang, Z., Hua, Y., & Yan, S. (2015b). Contextualizing object detection and classification. *IEEE TPAMI*, *37*(1), 13–27.

Chen, X., & Gupta, A. (2017). Spatial memory for context reasoning in object detection. In *ICCV*.

Chen, X., Kundu, K., Zhu, Y., Berneshawi, A. G., Ma, H., Fidler, S., & Urtasun, R. (2015c) 3d object proposals for accurate object class detection. In *NIPS* (pp. 424–432).

Chen, Y., Li, J., Xiao, H., Jin, X., Yan, S., & Feng J. (2017b). Dual path networks. In *NIPS* (pp. 4467–4475).

Chen, Y., Rohrbach, M., Yan, Z., Yan, S., Feng, J., & Kalantidis, Y. (2019b), Graph based global reasoning networks. In *CVPR*.

Chen, Y., Yang, T., Zhang, X., Meng, G., Pan, C., & Sun, J. (2019c). DetNAS: Neural architecture search on object detection. arXiv:1903.10979.

Cheng, B., Wei, Y., Shi, H., Feris, R., Xiong, J., & Huang, T. (2018a). Decoupled classification refinement: Hard false positive suppression for object detection. arXiv:1810.04002.

Cheng, B., Wei, Y., Shi, H., Feris, R., Xiong, J., & Huang, T. (2018b). Revisiting RCNN: On awakening the classification power of faster RCNN. In *ECCV*.

Cheng, G., Zhou, P., & Han, J. (2016). RIFDCNN: Rotation invariant and fisher discriminative convolutional neural networks for object detection. In *CVPR* (pp. 2884–2893).

Cheng, M., Zhang, Z., Lin, W., & Torr, P. (2014). BING: Binarized normed gradients for objectness estimation at 300fps. In *CVPR* (pp. 3286–3293).

Cheng, Y., Wang, D., Zhou, P., & Zhang, T. (2018c). Model compression and acceleration for deep neural networks: The principles, progress, and challenges. *IEEE Signal Processing Magazine*, *35*(1), 126–136.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *CVPR* (pp. 1800–1807).

Cinbis, R., Verbeek, J., & Schmid, C. (2017). Weakly supervised object localization with multi-fold multiple instance learning. *IEEE TPAMI*, *39*(1), 189–203.

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*.

Dai, J., He, K., Li, Y., Ren, S., & Sun, J. (2016a). Instance sensitive fully convolutional networks. In *ECCV* (pp. 534–549).

Dai, J., He, K., & Sun J. (2016b). Instance aware semantic segmentation via multitask network cascades. In *CVPR* (pp. 3150–3158).

Dai, J., Li, Y., He, K., & Sun, J. (2016c). RFCN: Object detection via region based fully convolutional networks. In *NIPS* (pp. 379–387).

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., & Wei, Y. (2017). Deformable convolutional networks. In *ICCV*.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *CVPR*, *1*, 886–893.

Demirel, B., Cinbis, R. G., & Ikizler-Cinbis, N. (2018). Zero shot object detection by hybrid region embedding. In *BMVC*.

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).

Diba, A., Sharma, V., Pazandeh, A. M., Pirsiavash, H., & Van Gool L. (2017). Weakly supervised cascaded convolutional networks. In *CVPR* (Vol. 3, p. 9).

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: Cambridge University Press.

Ding, J., Xue, N., Long, Y., Xia, G., & Lu, Q. (2018). Learning RoI transformer for detecting oriented objects in aerial images. In *CVPR*.

Divvala, S., Hoiem, D., Hays, J., Efros, A., & Hebert, M. (2009). An empirical study of context in object detection. In *CVPR* (pp. 1271–1278).

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.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., et al. (2014). DeCAF: A deep convolutional activation feature for generic visual recognition. *ICML*, *32*, 647–655.

Dong, X., Zheng, L., Ma, F., Yang, Y., & Meng, D. (2018). Few-example object detection with model communication. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(7), 1641–1654.

Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., & Tian, Q. (2019). CenterNet: Keypoint triplets for object detection. arXiv:1904.08189.

Dvornik, N., Mairal, J., & Schmid, C. (2018). Modeling visual context is key to augmenting object detection datasets. In *ECCV* (pp. 364–380).

Dwibedi, D., Misra, I., & Hebert, M. (2017). Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *ICCV* (pp. 1301–1310).

Endres, I., & Hoiem, D. (2010). Category independent object proposals. In K. Daniilidis, P. Maragos, & N. Paragios (Eds.), *European Conference on Computer Vision* (pp. 575–588). Berlin: Springer.

Enzweiler, M., & Gavrila, D. M. (2009). Monocular pedestrian detection: Survey and experiments. *IEEE TPAMI*, *31*(12), 2179–2195.

Erhan, D., Szegedy, C., Toshev, A., & Anguelov, D. (2014). Scalable object detection using deep neural networks. In *CVPR* (pp. 2147–2154).

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.

Everingham, M., Gool, L. V., Williams, C., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *IJCV*, *88*(2), 303–338.

Feichtenhofer, C., Pinz, A., & Zisserman, A. (2017). Detect to track and track to detect. In *ICCV* (pp. 918–927).

FeiFei, L., Fergus, R., & Perona, P. (2006). One shot learning of object categories. *IEEE TPAMI*, *28*(4), 594–611.

Felzenszwalb, P., Girshick, R., & McAllester, D. (2010a). Cascade object detection with deformable part models. In *CVPR* (pp. 2241–2248).

Felzenszwalb, P., Girshick, R., McAllester, D., & Ramanan, D. (2010b). Object detection with discriminatively trained part based models. *IEEE TPAMI*, *32*(9), 1627–1645.

Felzenszwalb, P., McAllester, D., & Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *CVPR* (pp. 1–8).

Finn, C., Abbeel, P., & Levine, S. (2017). Model agnostic meta learning for fast adaptation of deep networks. In *ICML* (pp. 1126–1135).

Fischler, M., & Elschlager, R. (1973). The representation and matching of pictorial structures. *IEEE Transactions on Computers*, *100*(1), 67–92.

Fu, C.-Y., Liu, W., Ranga, A., Tyagi, A., & Berg, A. C. (2017). DSSD: Deconvolutional single shot detector. arXiv:1701.06659.

Galleguillos, C., & Belongie, S. (2010). Context based object categorization: A critical survey. *Computer Vision and Image Understanding*, *114*, 712–722.

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.

Ghiasi, G., Lin, T., Pang, R., & Le, Q. (2019). NASFPN: Learning scalable feature pyramid architecture for object detection. arXiv:1904.07392.

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).

Gidaris, S., & Komodakis, N. (2015). Object detection via a multiregion and semantic segmentation aware CNN model. In *ICCV* (pp. 1134–1142).

Gidaris, S., & Komodakis, N. (2016). Attend refine repeat: Active box proposal generation via in out localization. In *BMVC*.

Girshick, R. (2015). Fast R-CNN. In *ICCV* (pp. 1440–1448).

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR* (pp. 580–587).

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.

Girshick, R., Iandola, F., Darrell, T., & Malik, J. (2015). Deformable part models are convolutional neural networks. In *CVPR* (pp. 437–446).

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. Cambridge: MIT press.

Goodfellow, I., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *ICLR*.

Grauman, K., & Darrell, T. (2005). The pyramid match kernel: Discriminative classification with sets of image features. *ICCV*, *2*, 1458–1465.

Grauman, K., & Leibe, B. (2011). Visual object recognition. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *5*(2), 1–181.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., et al. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, *77*, 354–377.

Guillaumin, M., Küttel, D., & Ferrari, V. (2014). Imagenet autoannotation with segmentation propagation. *International Journal of Computer Vision*, *110*(3), 328–348.

Gupta, A., Vedaldi, A., & Zisserman, A. (2016). Synthetic data for text localisation in natural images. In *CVPR* (pp. 2315–2324).

Han, S., Dally, W. J., & Mao, H. (2016). Deep Compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *ICLR*.

Hariharan, B., Arbeláez, P., Girshick, R., & Malik, J. (2014). Simultaneous detection and segmentation. In *ECCV* (pp. 297–312).

Hariharan, B., Arbeláez, P., Girshick, R., & Malik, J. (2016). Object instance segmentation and fine-grained localization using hypercolumns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(4), 627–639.

Hariharan, B., & Girshick R. B. (2017). Low shot visual recognition by shrinking and hallucinating features. In *ICCV* (pp. 3037–3046).

Harzallah, H., Jurie, F., & Schmid, C. (2009). Combining efficient object localization and image classification. In *ICCV* (pp. 237–244).

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask RCNN. In *ICCV*.

He, K., Zhang, X., Ren, S., & Sun, J. (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV* (pp. 346–361).

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).
He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *CVPR* (pp. 770–778).
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).
He, Y., Zhu, C., Wang, J., Savvides, M., & Zhang, X. (2019). Bounding box regression with uncertainty for accurate object detection. In *CVPR*.
Hinton, G., & Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504–507.
Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv:1503.02531.
Hoffman, J., Guadarrama, S., Tzeng, E. S., Hu, R., Donahue, J., Girshick, R., Darrell, T., & Saenko, K. (2014). LSDA: Large scale detection through adaptation. In *NIPS* (pp. 3536–3544).
Hoiem, D., Chodpathumwan, Y., & Dai, Q. (2012). Diagnosing error in object detectors. In *ECCV* (pp. 340–353).
Hosang, J., Benenson, R., Dollár, P., & Schiele, B. (2016). What makes for effective detection proposals? *IEEE TPAMI*, *38*(4), 814–829.
Hosang, J., Benenson, R., & Schiele, B. (2017). Learning nonmaximum suppression. In *ICCV*.
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).
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*.
Hu, H., Gu, J., Zhang, Z., Dai, J., & Wei, Y. (2018a). Relation networks for object detection. In *CVPR*.
Hu, H., Lan, S., Jiang, Y., Cao, Z., & Sha, F. (2017). FastMask: Segment multiscale object candidates in one shot. In *CVPR* (pp. 991–999).
Hu, J., Shen, L., & Sun, G. (2018b). Squeeze and excitation networks. In *CVPR*.
Hu, P., & Ramanan, D. (2017). Finding tiny faces. In *CVPR* (pp. 1522–1530).
Hu, R., Dollár, P., He, K., Darrell, T., & Girshick, R. (2018c). Learning to segment every thing. In *CVPR*.
Huang, G., Liu, S., van der Maaten, L., & Weinberger, K. (2018). CondenseNet: An efficient densenet using learned group convolutions. In *CVPR*.
Huang, G., Liu, Z., Weinberger, K. Q., & van der Maaten, L. (2017a). Densely connected convolutional networks. In *CVPR*.
Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., & Murphy, K. (2017b). Speed/accuracy trade offs for modern convolutional object detectors. In *CVPR*.
Huang, Z., Huang, L., Gong, Y., Huang, C., & Wang, X. (2019). Mask scoring rcnn. In *CVPR*.
Hubara, I., Courbariaux, M., Soudry, D., ElYaniv, R., & Bengio, Y. (2016). Binarized neural networks. In *NIPS* (pp. 4107–4115).
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. arXiv:1602.07360.
ILSVRC detection challenge results. (2018). http://www.image-net.org/challenges/LSVRC/.
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).
Jaderberg, M., Simonyan, K., Zisserman, A., et al. (2015). Spatial transformer networks. In *NIPS* (pp. 2017–2025).
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).
Jiang, B., Luo, R., Mao, J., Xiao, T., & Jiang, Y. (2018). Acquisition of localization confidence for accurate object detection. In *ECCV* (pp. 784–799).
Kang, B., Liu, Z., Wang, X., Yu, F., Feng, J., & Darrell, T. (2018). Few shot object detection via feature reweighting. arXiv:1812.01866.
Kang, K., Ouyang, W., Li, H., & Wang, X. (2016). Object detection from video tubelets with convolutional neural networks. In *CVPR* (pp. 817–825).
Kim, A., Sharma, A., & Jacobs, D. (2014). Locally scale invariant convolutional neural networks. In *NIPS*.
Kim, K., Hong, S., Roh, B., Cheon, Y., & Park, M. (2016). PVANet: Deep but lightweight neural networks for real time object detection. In *NIPSW*.
Kim, Y, Kang, B.-N., & Kim, D. (2018). SAN: Learning relationship between convolutional features for multiscale object detection. In *ECCV* (pp. 316–331).
Kirillov, A., He, K., Girshick, R., Rother, C., & Dollár, P. (2018). Panoptic segmentation. arXiv:1801.00868.
Kong, T., Sun, F., Tan, C., Liu, H., & Huang, W. (2018). Deep feature pyramid reconfiguration for object detection. In *ECCV* (pp. 169–185).
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*.
Kong, T., Yao, A., Chen, Y., & Sun, F. (2016). HyperNet: Towards accurate region proposal generation and joint object detection. In *CVPR* (pp. 845–853).
Krähenbühl, P., & Koltun, V. (2014), Geodesic object proposals. In *ECCV*.
Krasin, I., Duerig, T., Alldrin, N., Ferrari, V., AbuElHaija, S., Kuznetsova, A., et al. (2017). OpenImages: A public dataset for large scale multilabel and multiclass image classification. Dataset available from https://storage.googleapis.com/openimages/web/index.html.
Krizhevsky, A., Sutskever, I., & Hinton, G. (2012a). ImageNet classification with deep convolutional neural networks. In *NIPS* (pp. 1097–1105).
Krizhevsky, A., Sutskever, I., & Hinton, G. (2012b). ImageNet classification with deep convolutional neural networks. In *NIPS* (pp. 1097–1105).
Kuo, W., Hariharan, B., & Malik, J. (2015). DeepBox: Learning objectness with convolutional networks. In *ICCV* (pp. 2479–2487).
Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., PontTuset, J., et al. (2018). The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. arXiv:1811.00982.
Lake, B., Salakhutdinov, R., & Tenenbaum, J. (2015). Human level concept learning through probabilistic program induction. *Science*, *350*(6266), 1332–1338.
Lampert, C. H., Blaschko, M. B., & Hofmann, T. (2008). Beyond sliding windows: Object localization by efficient subwindow search. In *CVPR* (pp. 1–8).
Law, H., & Deng, J. (2018). CornerNet: Detecting objects as paired keypoints. In *ECCV*.
Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *CVPR*, *2*, 2169–2178.
LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444.
LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Lee, C., Xie, S., Gallagher, P., Zhang, Z., & Tu, Z. (2015). Deeply supervised nets. In *Artificial intelligence and statistics* (pp. 562–570).

Lenc, K., & Vedaldi, A. (2015). R-CNN minus R. In *BMVC15*.

Lenc, K., & Vedaldi, A. (2018). Understanding image representations by measuring their equivariance and equivalence. In *IJCV*.

Li, B., Liu, Y., & Wang, X. (2019a). Gradient harmonized single stage detector. In *AAAI*.

Li, H., Kadav, A., Durdanovic, I., Samet, H., & Graf, H. P. (2017a). Pruning filters for efficient convnets. In *ICLR*.

Li, H., Lin, Z., Shen, X., Brandt, J., & Hua, G. (2015a). A convolutional neural network cascade for face detection. In *CVPR* (pp. 5325–5334).

Li, H., Liu, Y., Ouyang, W., & Wang, X. (2018a). Zoom out and in network with map attention decision for region proposal and object detection. In *IJCV*.

Li, J., Wei, Y., Liang, X., Dong, J., Xu, T., Feng, J., et al. (2017b). Attentive contexts for object detection. *IEEE Transactions on Multimedia*, *19*(5), 944–954.

Li, Q., Jin, S., & Yan, J. (2017c). Mimicking very efficient network for object detection. In *CVPR* (pp. 7341–7349).

Li, S. Z., & Zhang, Z. (2004). Floatboost learning and statistical face detection. *IEEE TPAMI*, *26*(9), 1112–1123.

Li, Y., Chen, Y., Wang, N., & Zhang, Z. (2019b). Scale aware trident networks for object detection. arXiv:1901.01892.

Li, Y., Ouyang, W., Zhou, B., Wang, K., & Wang, X. (2017d). Scene graph generation from objects, phrases and region captions. In *ICCV* (pp. 1261–1270).

Li, Y., Qi, H., Dai, J., Ji, X., & Wei, Y. (2017e). Fully convolutional instance aware semantic segmentation. In *CVPR* (pp. 4438–4446).

Li, Y., Wang, S., Tian, Q., & Ding, X. (2015b). Feature representation for statistical learning based object detection: A review. *Pattern Recognition*, *48*(11), 3542–3559.

Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., & Sun, J. (2018b). DetNet: A backbone network for object detection. In *ECCV*.

Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., & Sun, J. (2018c). Light head RCNN: In defense of two stage object detector. In *CVPR*.

Lin, T., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017a). Feature pyramid networks for object detection. In *CVPR*.

Lin, T., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017b). Focal loss for dense object detection. In *ICCV*.

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).

Lin, X., Zhao, C., & Pan, W. (2017c). Towards accurate binary convolutional neural network. In *NIPS* (pp. 344–352).

Litjens, G., Kooi, T., Bejnordi, B., Setio, A., Ciompi, F., Ghafoorian, M., et al. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, *42*, 60–88.

Liu, C., Zoph, B., Neumann, M., Shlens, J., Hua, W., Li, L., FeiFei, L., Yuille, A., Huang, J., & Murphy, K. (2018a). Progressive neural architecture search. In *ECCV* (pp. 19–34).

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.

Liu, S., Huang, D., & Wang, Y. (2018b). Receptive field block net for accurate and fast object detection. In *ECCV*.

Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018c). Path aggregation network for instance segmentation. In *CVPR* (pp. 8759–8768).

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).

Liu, Y., Wang, R., Shan, S., & Chen, X. (2018d). Structure inference net: Object detection using scene level context and instance level relationships. In *CVPR* (pp. 6985–6994).

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3431–3440).

Lowe, D. (1999). Object recognition from local scale invariant features. *ICCV*, *2*, 1150–1157.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *IJCV*, *60*(2), 91–110.

Loy, C., Lin, D., Ouyang, W., Xiong, Y., Yang, S., Huang, Q., et al. (2019). WIDER face and pedestrian challenge 2018: Methods and results. arXiv:1902.06854.

Lu, Y., Javidi, T., & Lazebnik, S. (2016). Adaptive object detection using adjacency and zoom prediction. In *CVPR* (pp. 2351–2359).

Luo, P., Wang, X., Shao, W., & Peng, Z. (2018). Towards understanding regularization in batch normalization. In *ICLR*.

Luo, P., Zhang, R., Ren, J., Peng, Z., & Li, J. (2019). Switchable normalization for learning-to-normalize deep representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.2019.2932062.

Malisiewicz, T., & Efros, A. (2009). Beyond categories: The visual memex model for reasoning about object relationships. In *NIPS*.

Ma, J., Shao, W., Ye, H., Wang, L., Wang, H., Zheng, Y., et al. (2018). Arbitrary oriented scene text detection via rotation proposals. *IEEE TMM*, *20*(11), 3111–3122.

Manen, S., Guillaumin, M., & Van Gool, L. (2013). Prime object proposals with randomized prim's algorithm. In *CVPR* (pp. 2536–2543).

Mikolajczyk, K., & Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE TPAMI*, *27*(10), 1615–1630.

Mordan, T., Thome, N., Henaff, G., & Cord, M. (2018). End to end learning of latent deformable part based representations for object detection. In *IJCV* (pp. 1–21).

MS COCO detection leaderboard. (2018). http://cocodataset.org/#detection-leaderboard.

Mundy, J. (2006). Object recognition in the geometric era: A retrospective. In J. Ponce, M. Hebert, C. Schmid, & A. Zisserman (Eds.), *Book toward category level object recognition* (pp. 3–28). Berlin: Springer.

Murase, H., & Nayar, S. (1995a). Visual learning and recognition of 3D objects from appearance. *IJCV*, *14*(1), 5–24.

Murase, H., & Nayar, S. (1995b). Visual learning and recognition of 3d objects from appearance. *IJCV*, *14*(1), 5–24.

Murphy, K., Torralba, A., & Freeman, W. (2003). Using the forest to see the trees: A graphical model relating features, objects and scenes. In *NIPS*.

Newell, A., Huang, Z., & Deng, J. (2017). Associative embedding: End to end learning for joint detection and grouping. In *NIPS* (pp. 2277–2287).

Newell, A., Yang, K., & Deng, J. (2016). Stacked hourglass networks for human pose estimation. In *ECCV* (pp. 483–499).

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.

Oliva, A., & Torralba, A. (2007). The role of context in object recognition. *Trends in cognitive sciences*, *11*(12), 520–527.

Opelt, A., Pinz, A., Fussenegger, M., & Auer, P. (2006). Generic object recognition with boosting. *IEEE TPAMI*, *28*(3), 416–431.

Oquab, M., Bottou, L., Laptev, I., & Sivic, J. (2014). Learning and transferring midlevel image representations using convolutional neural networks. In *CVPR* (pp. 1717–1724).

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).

Osuna, E., Freund, R., & Girosit, F. (1997). Training support vector machines: An application to face detection. In *CVPR* (pp. 130–136).

Ouyang, W., & Wang, X. (2013). Joint deep learning for pedestrian detection. In *ICCV* (pp. 2056–2063).

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).

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).

Ouyang, W., Wang, K., Zhu, X., & Wang, X. (2017a). Chained cascade network for object detection. In *ICCV*.

Ouyang, W., Zeng, X., Wang, X., Qiu, S., Luo, P., Tian, Y., et al. (2017b). DeepIDNet: Object detection with deformable part based convolutional neural networks. *IEEE TPAMI*, *39*(7), 1320–1334.

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.

PASCAL VOC detection leaderboard. (2018). http://host.robots.ox.ac.uk:8080/leaderboard/main_bootstrap.php

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*.

Peng, X., Sun, B., Ali, K., & Saenko, K. (2015). Learning deep object detectors from 3d models. In *ICCV* (pp. 1278–1286).

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).

Perronnin, F., Sánchez, J., & Mensink, T. (2010). Improving the fisher kernel for large scale image classification. In *ECCV* (pp. 143–156).

Pinheiro, P., Collobert, R., & Dollar, P. (2015). Learning to segment object candidates. In *NIPS* (pp. 1990–1998).

Pinheiro, P., Lin, T., Collobert, R., & Dollár, P. (2016). Learning to refine object segments. In *ECCV* (pp. 75–91).

Ponce, J., Hebert, M., Schmid, C., & Zisserman, A. (2007). *Toward category level object recognition*. Berlin: Springer.

Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., et al. (2018). A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys*, *51*(5), 92:1–92:36.

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).

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).

Quanming, Y., Mengshuo, W., Hugo, J. E., Isabelle, G., Yiqi, H., Yufeng, L., et al. (2018). Taking human out of learning applications: A survey on automated machine learning. arXiv:1810.13306.

Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., & Belongie, S. (2007). Objects in context. In *ICCV*.

Rahman, S., Khan, S., & Barnes, N. (2018a). Polarity loss for zero shot object detection. arXiv:1811.08982.

Rahman, S., Khan, S., & Porikli, F. (2018b). Zero shot object detection: Learning to simultaneously recognize and localize novel concepts. In *ACCV*.

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).

Rebuffi, S., Bilen, H., & Vedaldi, A. (2017). Learning multiple visual domains with residual adapters. In *Advances in neural information processing systems* (pp. 506–516).

Rebuffi, S., Bilen, H., & Vedaldi A. (2018). Efficient parametrization of multidomain deep neural networks. In *CVPR* (pp. 8119–8127).

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real time object detection. In *CVPR* (pp. 779–788).

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In *CVPR*.

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*.

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).

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.

Ren, S., He, K., Girshick, R., Zhang, X., & Sun, J. (2016). Object detection networks on convolutional feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(7), 1476–1481.

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*.

Rowley, H., Baluja, S., & Kanade, T. (1998). Neural network based face detection. *IEEE TPAMI*, *20*(1), 23–38.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). ImageNet large scale visual recognition challenge. *IJCV*, *115*(3), 211–252.

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.

Schmid, C., & Mohr, R. (1997). Local grayvalue invariants for image retrieval. *IEEE TPAMI*, *19*(5), 530–535.

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*.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2014). OverFeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR*.

Sermanet, P., Kavukcuoglu, K., Chintala, S., & LeCun, Y. (2013). Pedestrian detection with unsupervised multistage feature learning. In *CVPR* (pp. 3626–3633).

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).

Shelhamer, E., Long, J., & Darrell, T. (2017). Fully convolutional networks for semantic segmentation. *IEEE TPAMI*.

Shen, Z., Liu, Z., Li, J., Jiang, Y., Chen, Y., & Xue, X. (2017). DSOD: Learning deeply supervised object detectors from scratch. In *ICCV*.

Shi, X., Shan, S., Kan, M., Wu, S., & Chen, X. (2018). Real time rotation invariant face detection with progressive calibration networks. In *CVPR*.

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.

Shrivastava, A., & Gupta A. (2016), Contextual priming and feedback for Faster RCNN. In *ECCV* (pp. 330–348).

Shrivastava, A., Gupta, A., & Girshick, R. (2016). Training region based object detectors with online hard example mining. In *CVPR* (pp. 761–769).

Shrivastava, A., Sukthankar, R., Malik, J., & Gupta, A. (2017). Beyond skip connections: Top down modulation for object detection. In *CVPR*.

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large scale image recognition. In *ICLR*.

Singh, B., & Davis, L. (2018). An analysis of scale invariance in object detection-SNIP. In *CVPR*.

Singh, B., Li, H., Sharma, A., & Davis, L. S. (2018a). RFCN 3000 at 30fps: Decoupling detection and classification. In *CVPR*.

Singh, B., Najibi, M., & Davis, L. S. (2018b). SNIPER: Efficient multiscale training. arXiv:1805.09300.

Sivic, J., & Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. *International Conference on Computer Vision (ICCV)*, *2*, 1470–1477.

Sun, C., Shrivastava, A., Singh, S., & Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV* (pp. 843–852).

Sun, K., Xiao, B., Liu, D., & Wang, J. (2019a). Deep high resolution representation learning for human pose estimation. In *CVPR*.

Sun, K., Zhao, Y., Jiang, B., Cheng, T., Xiao, B., Liu, D., et al. (2019b). High resolution representations for labeling pixels and regions. *CoRR.*,. arXiv:1904.04514.

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).

Sun, Z., Bebis, G., & Miller, R. (2006). On road vehicle detection: A review. *IEEE TPAMI*, *28*(5), 694–711.

Sung, K., & Poggio, T. (1994). Learning and example selection for object and pattern detection. MIT AI Memo (1521).

Swain, M., & Ballard, D. (1991). *Color indexing. IJCV*, *7*(1), 11–32.

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).

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017). Inception v4, inception resnet and the impact of residual connections on learning. In *AAAI* (pp. 4278–4284).

Szegedy, C., Reed, S., Erhan, D., Anguelov, D., & Ioffe, S. (2014). Scalable, high quality object detection. arXiv:1412.1441.

Szegedy, C., Toshev, A., & Erhan, D. (2013). Deep neural networks for object detection. In *NIPS* (pp. 2553–2561).

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *CVPR* (pp. 2818–2826).

Torralba, A. (2003). *Contextual priming for object detection. IJCV*, *53*(2), 169–191.

Turk, M. A., & Pentland, A. (1991). Face recognition using eigenfaces. In *CVPR* (pp. 586–591).

Tuzel, O., Porikli, F., & Meer P. (2006). Region covariance: A fast descriptor for detection and classification. In *ECCV* (pp. 589–600).

TychsenSmith, L., & Petersson, L. (2017). DeNet: Scalable real time object detection with directed sparse sampling. In *ICCV*.

TychsenSmith, L., & Petersson, L. (2018). Improving object localization with fitness nms and bounded iou loss. In *CVPR*.

Uijlings, J., van de Sande, K., Gevers, T., & Smeulders, A. (2013). *Selective search for object recognition. IJCV*, *104*(2), 154–171.

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.

Van de Sande, K., Uijlings, J., Gevers, T., & Smeulders, A. (2011). Segmentation as selective search for object recognition. In *ICCV* (pp. 1879–1886).

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).

Vedaldi, A., Gulshan, V., Varma, M., & Zisserman, A. (2009). Multiple kernels for object detection. In *ICCV* (pp. 606–613).

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *CVPR*, *1*, 1–8.

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).

Wang, H., Wang, Q., Gao, M., Li, P., & Zuo, W. (2018). Multiscale location aware kernel representation for object detection. In *CVPR*.

Wang, X., Cai, Z., Gao, D., & Vasconcelos, N. (2019). Towards universal object detection by domain attention. arXiv:1904.04402.

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).

Wang, X., Shrivastava, A., & Gupta, A. (2017). A Fast RCNN: Hard positive generation via adversary for object detection. In *CVPR*.

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).

Woo, S., Hwang, S., & Kweon, I. (2018). StairNet: Top down semantic aggregation for accurate one shot detection. In *WACV* (pp. 1093–1102).

Worrall, D. E., Garbin, S. J., Turmukhambetov, D., & Brostow, G. J. (2017). Harmonic networks: Deep translation and rotation equivariance. In *CVPR* (Vol. 2).

Wu, Y., & He, K. (2018). Group normalization. In *ECCV* (pp. 3–19).

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2019). A comprehensive survey on graph neural networks. arXiv:1901.00596.

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).

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).

Xiang, Y., Mottaghi, R., & Savarese, S. (2014). Beyond PASCAL: A benchmark for 3D object detection in the wild. In *WACV* (pp. 75–82).

Xiao, R., Zhu, L., & Zhang, H. (2003). Boosting chain learning for object detection. In *ICCV* (pp. 709–715).

Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In *CVPR*.

Yang, B., Yan, J., Lei, Z., & Li, S. (2016a). CRAFT objects from images. In *CVPR* (pp. 6043–6051).

Yang, F., Choi, W., & Lin, Y. (2016b). Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers. In *CVPR* (pp. 2129–2137).

Yang, M., Kriegman, D., & Ahuja, N. (2002). Detecting faces in images: A survey. *IEEE TPAMI*, *24*(1), 34–58.

Ye, Q., & Doermann, D. (2015). Text detection and recognition in imagery: A survey. *IEEE TPAMI*, *37*(7), 1480–1500.

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? In *NIPS* (pp. 3320–3328).

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.

Yu, F., & Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122.

Yu, F., Koltun, V., & Funkhouser, T. (2017). Dilated residual networks. In *CVPR* (Vol. 2, p. 3).

Yu, R., Li, A., Chen, C., Lai, J., et al. (2018). NISP: Pruning networks using neuron importance score propagation. In *CVPR*.

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.

Zagoruyko, S., Lerer, A., Lin, T., Pinheiro, P., Gross, S., Chintala, S., & Dollár, P. (2016). A multipath network for object detection. In *BMVC*.

Zeiler, M., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *ECCV* (pp. 818–833).

Zeng, X., Ouyang, W., Yan, J., Li, H., Xiao, T., Wang, K., et al. (2017). Crafting gbd-net for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(9), 2109–2123.

Zeng, X., Ouyang, W., Yang, B., Yan, J., & Wang, X. (2016). Gated bidirectional cnn for object detection. In *ECCV* (pp. 354–369).

Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016a). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE SPL*, *23*(10), 1499–1503.
Zhang, L., Lin, L., Liang, X., & He, K. (2016b). Is faster RCNN doing well for pedestrian detection? In *ECCV* (pp. 443–457).
Zhang, S., Wen, L., Bian, X., Lei, Z., & Li, S. (2018a). Single shot refinement neural network for object detection. In *CVPR*.
Zhang, S., Yang, J., & Schiele, B. (2018b). Occluded pedestrian detection through guided attention in CNNs. In *CVPR* (pp. 2056–2063).
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).
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.
Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018c). ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*.
Zhang, Z., Geiger, J., Pohjalainen, J., Mousa, A. E., Jin, W., & Schuller, B. (2018d). Deep learning for environmentally robust speech recognition: An overview of recent developments. *ACM Transactions on Intelligent Systems and Technology*, *9*(5), 49:1–49:28.
Zhang, Z., Qiao, S., Xie, C., Shen, W., Wang, B., & Yuille, A. (2018e). Single shot object detection with enriched semantics. In *CVPR*.
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*.
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).
Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2015). Object detectors emerge in deep scene CNNs. In *ICLR*.
Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016a). Learning deep features for discriminative localization. In *CVPR* (pp. 2921–2929).
Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., & Torralba, A. (2017a). Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(6), 1452–1464.
Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., & Sun, M. (2018a). Graph neural networks: A review of methods and applications. arXiv:1812.08434.
Zhou, P., Ni, B., Geng, C., Hu, J., & Xu, Y. (2018b). Scale transferrable object detection. In *CVPR*.
Zhou, Y., Liu, L., Shao, L., & Mellor, M. (2016b). DAVE: A unified framework for fast vehicle detection and annotation. In *ECCV* (pp. 278–293).
Zhou, Y., Ye, Q., Qiu, Q., & Jiao, J. (2017b). Oriented response networks. In *CVPR* (pp. 4961–4970).
Zhu, X., Tuia, D., Mou, L., Xia, G., Zhang, L., Xu, F., et al. (2017). Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, *5*(4), 8–36.
Zhu, X., Vondrick, C., Fowlkes, C., & Ramanan, D. (2016a). Do we need more training data? *IJCV*, *119*(1), 76–92.
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).
Zhu, Y., Zhao, C., Wang, J., Zhao, X., Wu, Y., & Lu, H. (2017a). CoupleNet: Coupling global structure with local parts for object detection. In *ICCV*.
Zhu, Y., Zhou, Y., Ye, Q., Qiu, Q., & Jiao, J. (2017b). Soft proposal networks for weakly supervised object localization. In *ICCV* (pp. 1841–1850).
Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., & Hu, S. (2016b). Traffic sign detection and classification in the wild. In *CVPR* (pp. 2110–2118).
Zitnick, C., & Dollár, P. (2014). Edge boxes: Locating object proposals from edges. In *ECCV* (pp. 391–405).
Zoph, B., & Le, Q. (2016). Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578.
Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. (2018). Learning transferable architectures for scalable image recognition. In *CVPR* (pp. 8697–8710).