Multimedia Tools and Applications
https://doi.org/10.1007/s11042-020-08976-6

A review of object detection based on deep learning

Youzi Xiao[1] · Zhiqiang Tian[1] · Jiachen Yu[1] · Yinshu Zhang[1] · Shuai Liu[1] · Shaoyi Du[2] · Xuguang Lan[2]

Received: 25 April 2019 / Revised: 14 February 2020 / Accepted: 22 April 2020 /
Published online: 12 June 2020

Abstract
With the rapid development of deep learning techniques, deep convolutional neural networks (DCNNs) have become more important for object detection. Compared with traditional handcrafted feature-based methods, the deep learning-based object detection methods can learn both low-level and high-level image features. The image features learned through deep learning techniques are more representative than the handcrafted features. Therefore, this review paper focuses on the object detection algorithms based on deep convolutional neural networks, while the traditional object detection algorithms will be simply introduced as well. Through the review and analysis of deep learning-based object detection techniques in recent years, this work includes the following parts: backbone networks, loss functions and training strategies, classical object detection architectures, complex problems, datasets and evaluation metrics, applications and future development directions. We hope this review paper will be helpful for researchers in the field of object detection.

Keywords Object detection · Deep learning · Deep convolutional neural networks · Computer vision

1 Introduction

The essence of object detection is to locate and classify objects, which uses rectangular bounding boxes to locate the detected objects and classify the categories of the objects. Object detection has some relations with object classification, semantic segmentation and instance segmentation. The details are illustrated in Fig. 1. Object detection is an important area of computer vision and has important applications in scientific research and practical industrial production, such as face detection [215], text detection [94, 282], pedestrian detection [170, 274], logo detection [87, 108], video detection [102, 103], vehicle detection [23, 54], and medical image detection [145], the details are shown in Fig. 2. The limitation

✉ Zhiqiang Tian
zhiqiangtian@xjtu.edu.cn

[1] School of Software Engineering, Xi'an Jiaotong University, Xi'an, China

[2] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, China

Fig. 1 **a** Object classification needs to identify the category of objects in image. **b** Object detection not only needs to identify the category of objects, but also needs to locate the objects with rectangular bounding boxes. **c** Semantic segmentation only needs to predict the categories of each pixel, and does not need to distinguish the object instances. **d** Instance segmentation needs to predict both the categories of each pixel and object instances

of the computing resources, the datasets, and the basic theories have limited the development and application of deep neural networks in recent decades [122]. Therefore, in the field of computer vision, the traditional object detection algorithms were still popular. Traditional object detection algorithms include DPM [58], Selective Search [224], Oxford-MKL [228], and NLPR-HOGLBP [263], etc. The basic architecture of traditional object detection algorithms mainly divided into region selector, feature extractor, and classifier, which is demonstrated in Fig. 3.

Although the traditional object detection is relatively mature, it has its own inherent shortcomings. First, sliding-window [57] based region selection strategy has high computing complexity and high window redundancy. Second, the morphological diversity of the

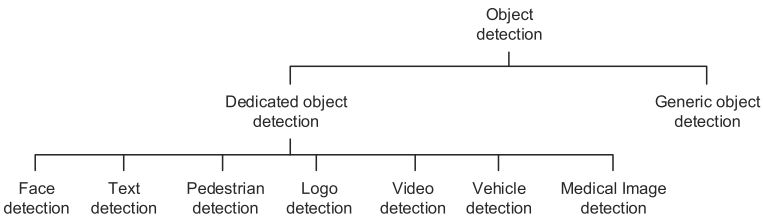

Fig. 2 The application of object detection can be divided into generic object detection and dedicated object detection

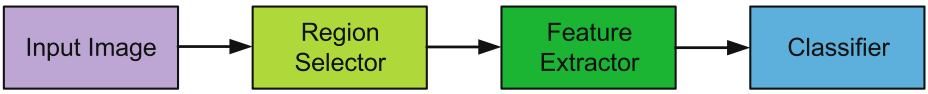

Fig. 3 The basic architecture of the traditional object detection algorithms. The region selector mainly uses sliding-windows of different sizes and ratios to slide on the image from left to right and top to bottom by a certain step size. The image blocks cropped by the sliding window are transformed to form an image with uniform size. The feature extractor mainly uses HOG [40], Haar [140], SIFT [155], and other algorithms to extract feature from image blocks. Finally, the classifier uses algorithms such as SVM [36] and Adaboot [229] to identify object category

appearances, the diversity of illumination changes, and the diversity of the background make it is difficult to design robust features manually. During 2010-2012, only small gains were obtained by building ensemble systems and employing minor variants of the existing methods [66]. Deep convolutional neural networks can learn the features of images from low-level to high-level, which are very robust [8, 105, 168]. Therefore, the researchers gradually turned their attention to the DCNNs.

With the increase of computing power and the number of image datasets [194], there are more opportunities for the development of the DCNNs-based object detection. In 2012, A. Krizhevsky et al. proposed a DCNN called AlexNet [116]. It won the competition of ILSVRC-2012 (top-5 error rates of 15.3%). This work triggered a wave of research and application of the deep convolutional neural networks. In 2014, R. Girshick et al. proposed a RCNN (Regions with CNN features) [66], which is a milestone in applying the DCNNs-based method for object detection. In 2015, J. Redmon et al. proposed an object detection system based on a single neural network called YOLO (You Only Look Once: Unified, Real-Time Object Detection) [187], which was presented in CVPR2016. With the publication of the series of papers, the DCNNs-based object detection methods break through the bottleneck of traditional object detection methods. Object detection has entered a period of using deep learning techniques. Some traditional object detection methods and DCNNs-based object detection methods are shown in Fig. 4.

From the development trend of the object detection in recent years. First, the accuracy of detection is continuously improved to satisfy the application of various complex scenarios. Second, the speed of detection is improved to satisfy real-time system applications while ensuring the accuracy. Therefore, attention must be paid to the trade-off between accuracy and speed [93, 136, 174] in the future research works. In order to get state-of-the-art results, the trade-off between accuracy and speed is quite important.

More than 300 papers in the field of object detection are cited in this review paper, most of which are based on deep learning. These include two published review papers, Deep Learning for Generic Object Detection: A Survey [147] and Object Detection with Deep Learning: A Review [285]. There are many improvements and rich parts to compare with them. In this review, the article highlights the pipeline of the object detection architectures. First, the backbone networks and loss functions are introduced, which clearly reflect the components and effects of the architecture. The introduction of specific object detection architectures make them easier for researchers to accept based on previous introductions. The structure of this review is more clearly and the content of each section is more reasonable compared to the two published review papers.

Five contributions are listed as follows:

(1). The development of deep convolutional neural networks are summarized, and the backbone networks used for object detection are compared in recent years.

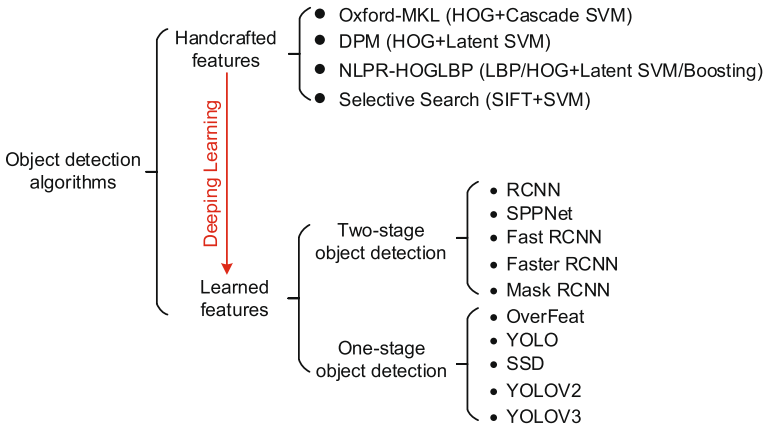

Fig. 4 Classical traditional object detection (such as Oxford-MKL [228], DPM [58], NLPR-HOGLBP [263], and Selective Search [224]) and object detection based on DCNNs. It is seen from the figure that the revival of deep learning transforms the handcrafted features of the object detection into learned features, which is the fundamental difference between the two. Two-stage object detection and one-stage object detection are elaborated in Section 4

(2). The network frameworks is analyzed and compared in detail, and the loss functions of object detection is summarized.
(3). Some guidance and advancements are provided in the future development of object detection.
(4). Difficulties and solutions to object detection are summarized.
(5). The applications of object detection are summarized and the technical details are fully analyzed.

The remainder of this review paper is organized as follows. In the Section 2, the architecture of the backbone networks are introduced in detail, and their performance and parameters are compared and analyzed. In the Section 3, the loss functions for object detection are summarized, and the loss function constructions and training strategies are implemented in the actual architecture. The Section 4 summarizes the milestone object detection architectures. The Section 5 summarizes some important complex problems in the field of object detection. The Section 6 summarizes and datasets and evaluation criteria. The Section 7 summarizes the applications of object detection. The Section 8 summarizes the future development directions of object detection.

2 Backbone network for object detection

A variety of DCNNs with powerful capabilities are proposed. The most of them have innovative architectures, which are shown in Fig. 5. The DCNNs are the backbone network for object detection (or classification, segmentation [37, 152]). In order to improve the performance of feature representation, the network architecture becomes more and more complicated (the network layer is deeper and the network parameters are increased). In the environment with limited computing power and storage, such as mobile [248], autonomous driving [24, 218], industrial production. Lightweight network structures are proposed,

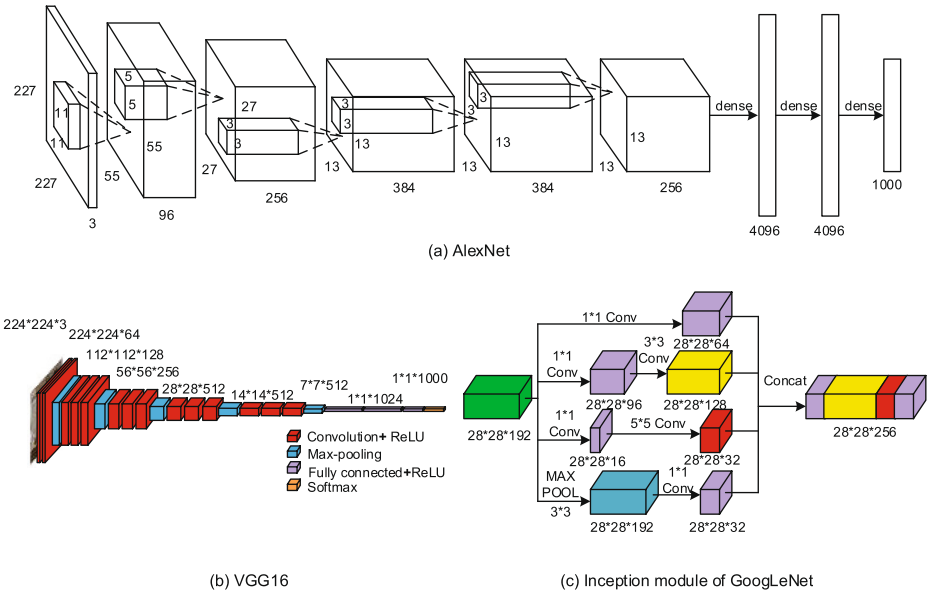

Fig. 5 Milestones of deep convolutional neural network architectures and network modules

which simplifies the network structure without reducing the feature representation capability. In Table 1, object detection backbone networks are listed. It can be found that the accuracy of the complex backbone networks(CBNs) can be improved by increasing the depth of the network. It can also be found that reducing the parameters in reasonable ways, which do not affect the accuracy of the lightweight backbone networks(LBNs).

2.1 Complex backbone network

AlexNet The first convolutional neural network was proposed by Yann LeCun based on previous research work in 1998, which is called LeNet-5. It has an average precision of 98% on the MNIST dataset [123]. LeNet-5 is a classical convolutional neural network used to identify handwritten numeric characters. Its emergence determines the basic architecture of the deep convolutional neural networks. The convolutional layer, pooling layer, and fully-connected layer in LetNet-5 are the basic components of the deep convolutional neural networks. This is the first time that a convolutional neural network can be available. However, it has not made great progress in the next decade. The expansion are limited by computing power. At the same time, the traditional machine learning algorithms such as SVM [36, 246] can achieve the same accuracy and even better. Therefore, the convolutional neural networks did not attract much attention. In 2012, AlexNet won the championship of ILSVRC-2012 competition. On the testing set of LSVRC-2010, they achieved top-1 and top-5 error rates of 37.5% and 17.0%. On the testing set of ILSVRC-2012, they achieved top-5 error rates of 15.3%, which is far higher than the second-place. AlexNet increases the depth and breadth of the LeNet-5 network architecture. It consists of five convolutional layers (Conv), three max-pooling layers and three fully-connected layers (FC), with a total of 60 million parameters [116]. The architecture is shown in Fig. 5a.

The great success of AlexNet is due to the following technologies:

Table 1 We summarize and compare the commonly used object detection backbone networks, highlighting and comparing complex backbone networks(CBNs) and lightweight backbone networks(LBNs)

	No.	DCNN	Top-1 Accuracy	Top-5 Accuracy	Params (M)	Mult-Adds (M)	Teamwork	Year	Delivered
CBNs	1	AlexNet [116]	57.20%	80.30%	60M	720M	UTORONTO	2012	NIPS2012
	2	ZFNet [266]	64.00%	85.20%	58M	-	NYU	2013	ECCV14
	3	VGGNet16 [207]	71.50%	89.80%	138M	15300M	Oxford & DeepMind	2015	ICLR15
	4	GoogLeNet [213]	69.80%	93.30%	6.8M	1550M	Google	2014	CVPR15
	5	InceptionV2 [97]	79.90%	95.20%	12M	1940M	Google	2015	ICML15
	6	InceptionV3 [214]	82.70%	96.50%	23.6M	5000M	Google	2015	CVPR16
	7	InceptionV4 [212]	83.50%	96.90%	41M	-	Google	2016	AAAI17
	8	ResNet50 [82]	79.30%	96.40%	23.4M	3832M	Microsoft	2015	CVPR16
	9	ResNet101 [82]	80.10%	96.40%	42M	-	Microsoft	2015	CVPR16
LBNs	10	SqueezeNet [96]	57.20%	80.30%	1.25M	1700M	Berkeley&Stanford	2016	ICLR17
	11	Xception	79.00%	94.50%	22.8M	-	Google	2017	CVPR17
	12	MobileNetV1 [88]	70.70%	89.50%	4.24M	575M	Google	2017	CVPR17
	13	MobileNetV2 [196]	72.00%	91.00%	3.4M	300M	Google	2018	-
	14	ShuffleNetV1 [250]	71.50%	-	3.4M	292M	Face++	2017	CVPR17
	15	ShuffleNetV2 [157]	-	-	-	-	Face++	2018	ECCV2018
	16	NASNet-A [297]	74.00%	91.60%	5.3M	564M	Google Brain	2018	CVPR17
	17	PeleeNet [237]	71.30%	90.30%	2.8M	508M	UWO	2018	CVPR18
	18	SqueezeNext [64]	67.50%	88.20%	3.2M	708M	Berkeley	2018	-

Table 1 (continued)

No.	DCNN	Top-1 Accuracy	Top-5 Accuracy	Params (M)	Mult-Adds (M)	Teamwork	Year	Delivered
19	MnasNet [216]	70.60%	89.50%	4.2M	317M	Google Brain	2018	-
20	MnasNet-92 [216]	74.80%	92.10%	4.4M	388M	Google Brain	2018	-
21	PNASNet [146]	74.20%	91.90%	5.1M	588M	JHU & Google AI & Stanford	2018	ECCV18

The Top-1 Accuracy and the Top-5 Accuracy represent the classification accuracy on the ImageNet dataset. Params indicate the number of network parameters. Mult-Adds indicate the number of multiply-add operations for a single image

(1). Data enhancement methods (such as horizontal flipping, random clipping, translational transformation, color illumination transformation) are used to extend the dataset and reduce overfitting.
(2). Traditional activation functions (such as Sigmoid and Tanh) are replaced by a ReLu activation function [163]. It solves the problem of gradient dispersion in deeper network.
(3). A part of neurons are randomly removed by using a Dropout [209] regularization method during training. It can reduce the complex inter-adaptive relationship of neurons and the probability of overfitting.
(4). Multi-GPUs parallel computing technology is used, which communicates between certain layers and speeds up network training.

Since the success of AlexNet, various DCNNs have emerged in the past few years, such as VGGNet [207], ZFNet [266], GoogLeNet [213], and ResNet [82]. Throughout the development of network structures, the ways to improve the performance of the network models include increasing depth of the networks. At the same time, excellent design of subtle topologies and bottlenecks to reduce the parameters of the network models and improve generalization ability.

ZFNet Researchers want a visualization technology of the convolutional layers, which is possible to intuitively analyze the changes in image feature maps at each layer. Matthew D. Zeiler et al. proposed a method for visualizing feature maps using unpooling layers and deconvolution [267, 268] layers in ZFNet [266]. The authors change the size of the convolution kernel of the first layer in AlexNet from 11×11 to 7×7 and adjusted the stride size from 4 to 2. These changes can preserve more low-level features. A small convolution kernel can reduce the downsampling rate, which is conducive to the location of large objects and the recognition of small objects [135].

GoogLeNet In the current popular two-stage object detectors, such as the RCNN series [65, 66, 191], the object detection is divided into two stages. First, the low-level features are used to locate the object, and the DCNN is used to classify the located objects. Therefore, the improvement of feature representation performance is advantageous for location and classification. The way to improve performance is to increase the number of layers in the network and the number of neurons in each layer. However, the larger the network size is, the more parameters there will be. The 1×1 convolution kernel proposed by Min Lin et al. in Network-in-Network [141] is introduced in GoogLeNet [213]. The 1×1 convolution kernel can not only reduce or increase the dimension, but also implement cross-channel information integration. The dimensionality reduction of the 1×1 convolution kernel can reduce the computational complexity while increasing the depth and width of the network. According to this ideology, the author proposed an Inception module [213] (see Fig. 5c) with dimension reductions. It use 9 Inception modules in the network to change the serial structure to parallel structure, and then replacing the fully connected layer with the average pooling layer. The required calculation parameters are reduces from $7\times7\times1024$ to $1\times1\times1024$.

VGGNet VGGNet increased the depth of AlexNet to 16-19 layers [207], which improves the feature representation of network. The mainstream network architectures are VGG16 and VGG19, and the architecture of VGG16 is shown in Fig. 5b. AlexNet and ZFNet use kernel of size 11×11 (stride of 4) and kernel of size 7×7 (stride of 2) in the first convolutional layer. The size of the convolution kernel is further reduced in VGGNet, and 3×3

convolution kernel (stride of 1) is used in each layer. The small kernel and stride are more conducive to extracting the location information of the object in the image. The small kernel instead of the large kernel has the advantage of increasing the depth of the network and keeping the receptive field unchanged. The feature representation capability of the network model is enhanced after reducing the parameters.

ResNet The gradient dispersion and the gradient explosion problems may occur with increasing the number of layers. These two problems are effectively solved on [207, 213], which enables the network to converge using stochastic gradient descent (SGD) algorithm at tens of layers. However, as the depth of the network continues to increase, there will be situations where the accuracy reaches saturation and then declines rapidly during training. This phenomenon is called degradation [82]. In order to solve this problem, Kaiming He et al. proposed a residual learning module. This can increase the depth of the network to hundreds of layers, so that the feature representation capability of the network is further enhanced. With the top-5 error rates of 3.57%, it won the first place in the ILSVRC2015 classification and detection.

The residual learning module is essentially shortcut connections, adding the result $\mathcal{F}(X)$ obtained from a layer or stacked layers to the input value X. The expected output value is $\mathcal{H}(X)=\mathcal{F}(X)+X$. Since the residual learning module solves the problem of training degradation, the depth of network is increased and the performance is continuously improved. ResNet50 and ResNet101 are widely used as backbone networks for object detection.

DetNet From the architecture and function of these classical DCNNs. The DCNNs are trained on classification task, which can be used as backbone networks for object detection and other tasks. Through the research of generic DCNNs and object detection task. Zeming Li et al. proposed a DetNet - A Backbone network for object detection [135], which makes up for the shortcoming of generic backbone networks in object detection task. The generic backbone networks use a large down-sampling rate, which ensures that the large receptive field is beneficial to the classification of images, but is not conducive to accurately locating large objects and recognizing small objects. The DetNet is compared with the traditional backbone network, which uses a dilated convolution [261, 262] instead of down-sampling the last few layers to ensure the larger receptive field and resolution.

Inspired by the architecture and ideology of classical DCNNs, researchers are constantly improving or integrating with each other based on these networks. InceptionV2 [97] inherits the ideology of InceptionV1 (GoogLeNet) [213], which uses 2 kernels of size 3×3 instead of 1 kernel of size 5×5, and proposes that Batch Normalization (BN) speeds up the learning rate of the network. Since the small kernel can improve the performance of networks, Christian Szegedy et al. proposed an InceptionV3 [214], which replaces $n\times n$ kernel by $1\times n$ kernel series concatenation $n\times1$ kernel. Similarly, ResNet [82] demonstrates the advantages of shortcut connections on deep networks. Therefore, an InceptionResNets [212] network is formed by combining the Inception networks and the shortcut connections. The research of DCNNs as the backbone networks is still in progress. Due to the limitation of the paper, such as ResNeXt [252], DenseNet [92], and SE ResNet [89] are not described it in detail.

2.2 Lightweight backbone network

The architecture of the complex backbone networks are described in Section 2.1. The main development direction is to deepen the depth of the network to improve the performance

of the network. For example, there are 7-layers for AlexNet, 16-layers for VGGNet, 22-layers for GoogLeNet, 50 to 152 layers for ResNet, and thousands of layers of ResNet and DenseNet [82, 92, 116, 207, 213]. This increases the size of the network parameters. Although some methods are used to reduce the parameters of the network, such as the kernel decomposition method is proposed in the InceptionV1/V2/V3 [97, 213, 214]. In short, the increase of model parameters brings about two problems, storage space and testing time. Since the backbone network accounts for about 90% of the calculation and storage of the object detection, these two problems hinder the application of the DCNNs-based object detection in real scenes. Where storage and computing resources are limited, such as face recognition [215, 240], autonomous driving [24, 218], mobile phone [216, 248], industrial production [245], embedded systems [158, 181]. In order to promote the process of industrialization of DCNNs, researchers proposed lightweight networks to reduce network parameters and ensure network performance.

SqueezeNet The traditional works use three operations to compress networks, which are singular value decomposition (SVD), network pruning, and deep compression [43, 65, 254]. However, SqueezeNet is not a kind of method to compress the network. It is a few parameters network architecture. The new network architecture is proposed in SqueezeNet, which is called the Fire Module [96]. The Fire Module consists of a squeeze layer and an expand layer, where the squeeze layer consists of only 1×1 convolutional filters, and the expand layer consists of 1×1 and 3×3 convolutional filters (kernels). The output of the expend layer is concatenated by the calculated feature map of the 1×1 convolution filters and the 3×3 convolution filters. Finally, the parameters size is reduced to less than 0.5MB using the deep compression technology [73]. The parameters size is reduced to 1/510 of the AlexNet parameters.

Xception Xception is an improvement over the InceptionV3 that replaces the convolution in the InceptionV3 with a depthwise separable convolution [34]. A depthwise separable convolution is proposed in the Xception, which is widely used by MobileNet [88, 196], ShuffleNet [157, 250], and other network architectures. Although the depthwise separable convolution reduces in computational complexity in the Xception, the implementation is not efficient enough in DCNNs. The structure is also constantly improving, which is explained in MobileNet.

MobileNet The SqueezeNet uses a bottleneck called Fire Module to build a lightweight backbone network [96]. In MobileNet, a convolution method different from the traditional convolution method is the depthwise separable convolution [88]. The depthwise separable convolution turns the standard convolution into a depthwise convolution and a 1×1 pointwise convolution. The depthwise convolution is that the feature channel is only operated with one convolution kernel. The number of convolution kernels is equal to the number of feature channels. The pointwise convolution is 1×1 convolution kernel. For example, the input feature map size is $D_K \times D_K \times M$, and the output feature map size is $D_F \times D_F \times N$. In the traditional convolution method, the calculation amount is $D_K \times D_K \times M \times N \times D_F \times D_F$. But in the depthwise separable convolution method, the calculation amount is $D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F$, so the both ratio is $\frac{1}{N} + \frac{1}{D_K^2}$. By this way, the amount of calculation is reduced.

MobileNetV2 Shortcut connections improve network performance in ResNet [82]. Therefore, the MobileNetV2 convolution block is formed by combining the depthwise separable

convolution of MobileNet with a shortcut connection [196]. In order to obtain more features, a 1×1 convolution kernel expansion channel number is added before the depthwise convolution. It use the Linear bottlenecks instead of the ReLU activation function to prevent the feature from being destroyed. So the core of MobileNetV2 is Inverted Residuals and Linear Bottlenecks.

ShuffleNet The depthwise separable convolution is proposed in Xception [34]. It transforms the traditional convolution method into a combination of depthwise convolution and pointwise convolution. This way reduces the amount of parameters and calculation, but the calculation of pointwise convolution is still large. In ShuffleNet [250], the authors proposed to use pointwise group convolution instead of pointwise convolution. Convolution is operated in each group, which reduces the calculation significantly. The channel shuffle is used to implement information exchange between groups. The residual block with depthwise separable convolution, pointwise group convolution, and channel shuffle forms the ShuffleNet unit. The shuffleNet can be built based on the ShuffleNet unit.

ShuffleNetV2 With the study of lightweight networks, researchers used speed instead of FLOPs to measure power. Then used four design guidelines for lightweight networks that weighed speed and accuracy. Based on these four guidelines, the researchers designed the ShuffleNetV2 network structure [157]. Channel split is introduced on the basis of ShuffleNetV1, and the feature input channel of each unit is divided into two parts, each of them consists of three convolutions. Unlike ShuffleNetV1, two 1×1 convolutions are used instead of two group convolutions. After the convolution operation, the two branches are concatenated. Finally, the Channel Shuffle operation is used to exchange information between the two branches.

PeleeNet The core idea of MobileNetV1/V2 and ShuffleNetV1/V2 is depthwise separable convolution [34], which reduces the amount of computation and storage space, but lacks efficient implementation. Based on DenseNet [92], the researchers proposed a PeleeNet [237] as the backbone network for object detection. In the PeleeNet, two-way dense layers are used to obtain receptive fields of different scales, and feature learning of small-scale objects and large-scale objects is simultaneously performed. The stem block before the first dense layer can improve the representation of the feature without increasing the amount of calculation. In order to be suitable for mobile applications, the dynamic number of channels in bottleneck layer, and transition layer without compression and composite function are also the highlights.

3 Loss function for object detection

In deep learning or machine learning, the loss function is also called the cost function. The main purpose of the loss function is to measure the deviation between the predicted value of the network and the true value of the sample. The smaller the value of the loss function, the better the training of the network model, which proves the convergence and robustness of the network model. The object detection is divided into object classification and object location. Therefore, the loss function includes classification loss ($Cls\ Loss$) and location loss ($Loc\ Loss$). The loss function of some classic object detection architectures are summarized in Table 2. The classification loss belongs to the classification, and the location loss belongs to the bounding-box regression. In recent works, many innovative methods have

Table 2 The compositions and characteristics of the loss functions of the classical object detection architectures

Classic object detection architectures	Classification		Bounding-box regression		Training strategy
	Classifier	Loss	Regressor	Loss	
RCNN, SPPNet	SVM	Hinge Loss	Bounding-box	L_2 Loss	Stage-wise training
Fast RCNN	Softmax	Cross entropy Loss	Bounding-box	smooth $_{L_1}$	Multi-task training
Faster RCNN	Softmax	Cross entropy Loss	Bounding-box	smooth $_{L_1}$	Multi-task training
RPN	Softmax	Binary cross entropy Loss	Bounding-box	smooth $_{L_1}$	Multi-task training
SSD	Softmax	Cross entropy Loss	Bounding-box	smooth $_{L_1}$	Multi-task training
YOLO	Softmax	L_2 Loss	Bounding-box	L_2 Loss	Multi-task training

been proposed in the loss function design and the loss-based training strategies for network architecture. Representative designs include a stage-wise training in the RCNN/SPPNet [66, 81] and a multi-task training in the Fast/Faster RCNN [65, 191]. The stage-wise training separates classification and location, while the multi-task training concentrates classification and location into a whole loss function. Moreover, the loss function can also achieve specific functions. For example, Xinlong Wang et al. proposed a Repulsion Loss to solve the dense occlusion problem [242]. Tsung-Yi Lin et al. proposed a Focal Loss to solve the class imbalance problem [142]. These results demonstrate that the design of the loss function has a positive effect on the robustness of the network model. Next, the classification losses and regression losses of object detection are briefly introduced. Then, the training strategies of the milestones of object detection and the specific loss function are described.

3.1 Classification loss

Hinge loss [6, 61] is a proxy function of the 0-1 loss function. It can be used as a loss for the max-margin problem in machine learning or deep learning, and can be extended to multi-class support vector machine (SVM) loss. The standard form of Hinge loss is listed as follows, which is suitable for binary classification.

$$\begin{aligned} L(y) &= \max(0, 1 - t \cdot y) \\ y &= w \cdot x + b \end{aligned} \tag{1}$$

where (w, b) are hyperplane parameters, x is the data vector that needs to be classified, $y \in [-1, 1]$ is the raw output of the classifier, not the predicted class label. And $t \in \{-1, 1\}$ represents the intended value.

The binary classification case does not apply to all actual situations, so the hinge loss needs to be extended to multiple classifications. It is defined as follows:

$$L(y) = \sum_{j \neq i} \max\left(0, y_j - y_i + 1\right) \tag{2}$$

The meaning of the above formula is to let the scores y_j of other classes minus the real class score y_i, and then sum them up. The smaller the value after summation is, the lower the score of the error class will be.

Cross entropy loss [41] is also called log loss, which is used in the softmax classifier. The function of softmax is to convert the $(k + 1) \times 1$ dimensional feature into the $(k + 1) \times 1$ dimensional probability distribution. The index value of the maximum probability is the category label of the predicted sample. Therefore, according to the characteristics of the softmax function, the cross entropy loss can be defined as follows:

$$L(p, u) = -\sum_{i=0}^{k} u_i \log p_i \tag{3}$$

where $p = (p_0, \cdots, p_k)$ is the probability distribution of $K+1$ categories calculated by the softmax, and u is the true category label. The function structure shows that the cross entropy loss is the distance between the predicted value and the target true value. The function structure has convex optimization, which has good convergence when the gradient descent. It is more suitable for multi-category classification than hinge loss.

If the multi-category task becomes a binary-category task, the cross entropy loss can be reduced to the **binary cross entropy loss**, which can be considered as a special form of multi-category. Its form is defined as follows:

$$L(p, u) = -u \log p - (1 - u) \log(1 - p) \tag{4}$$

Here, the category label $u \in \{0, 1\}$, p indicates the probability of the prediction belonging to category label $u = 1$.

The above three losses are mainly used for classifiers. Such as the use of the hinge loss in RCNN/SPPNet [66, 81], the use of cross entropy loss in Fast RCNN/Faster RCNN/SSD [65, 150, 191], and the use of binary cross entropy loss in YOLOV3 [189]. However, in YOLO [187], the classifier uses squared loss, which should be used for bounding-box regression. This shows that the design of the loss function is based on the network architecture and is not static.

3.2 Location loss

The **squared loss** is one of the basic loss functions of the bounding-box regression, also known as the L_2 loss [99]. It represents the sum of the squares of the differences between the target value and the predicted value. Its basic form is defined as follows:

$$L(y, f(x)) = (y - f(x))^2 \tag{5}$$

Here, y represents the target value corresponding to the input data x, and $f(x)$ represents the predicted value of the input data x obtained by mapping f. When input data $X = \{x_1, x_2, \cdots, x_n\}$, target value $Y = \{y_1, y_2, \cdots, y_n\}$ the loss function becomes:

$$L(Y, f(X)) = \sum_{i=1}^{n} (y_i - f(x_i))^2 \tag{6}$$

which is called the **Residual Sum of Squares (RSS)**. The mean value of RSS is usually used as the regression loss, which is called the **Mean Square Error (MSE)**.

$$L(Y, f(X)) = \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 \tag{7}$$

Another loss function for bounding-box regression is the **absolute loss**, called the L_1 [99]. The difference between the L_1 loss and the L_2 lossis that L_1 represents the sum of the absolute values of the difference between the target value and the predicted value. The basic form is expressed as follows:

$$L(y, f(x)) = |y - f(x)| \tag{8}$$

If there are n samples, the loss function becomes the following form, which can be called the **Sum of Absolute Differences (SAD)**.

$$L(Y, f(X)) = \sum_{i=1}^{n} |y_i - f(x_i)| \tag{9}$$

The mean value of SAD is usually used as the regression loss, which is called the **Mean Absolute Error (MAE)** [22, 247].

$$L(Y, f(X)) = \frac{1}{n} \sum_{i=1}^{n} |y_i - f(x_i)| \tag{10}$$

L_1 loss and L_2 loss [65] have their own advantages and disadvantages when used for bounding-box regression. The L_1 loss is more robust to outliers, but it has points where the derivative cannot be deduced, making the gradient descent inefficient. The gradient descent of L_2 loss is more accurate and simple to calculate, but more sensitive to outliers. Therefore, the advantages of L_1 loss and L_2 loss are combined in the design of the bounding-box regression loss function, the Section 3.3 shows the design method.

3.3 Loss-based training strategies

Stage-wise training method RCNN/SPPNet are pioneering approaches to object detection based on DCNNs, which is the stage-wise training method. First, the SVM algorithm is used for classification, and then the bounding-box regression [66, 81] is used to accurately locate the objects. Therefore, the stage-wise loss function is used in the training. After calculating the SVM loss function, the bounding-box regression loss function is calculated.

The loss function of the SVM classification algorithm is the Hinge loss with the $L2$ regularization term [162]. The function form is defined as follows:

$$L_{cls} = c \sum_i \max\left(0, 1 - p_i^* \cdot p_i\right) + \frac{1}{2} w^2 \tag{11}$$

Here, p_i^* represents the true category of the object, p_i represents the probability of the predicted object category, and i is the index of the mini-batch.

The bounding-box regression in RCNN/SPPNet uses the L_2 loss as the basic skeleton of the loss function. The main principle is to punish the distance deviation between the predicted bounding-box and the ground truth to optimize the robustness of the prediction. The function is defined as follows:

$$\begin{aligned} t_x^* &= \left(x^* - x\right)/w \quad , \quad t_y^* = \left(y^* - y\right)/h \\ t_w^* &= \log\left(w^*/w\right) \quad , \quad t_h^* = \log\left(h^*/h\right) \end{aligned} \tag{12}$$

$$L_{loc} = \sum_i \left(t_*^i - w_*^T \phi\left(t^i\right)\right)^2 \tag{13}$$

Here, the true coordinate is $t^* = (x^*, y^*, w^*, h^*)$, the predicted coordinate is $t = (x, y, w, h)$, where (x, y) represents the coordinate of the box center, (w, h) represents the width and height of the box. w_*^T is the learned parameter, and $\phi\left(t^i\right)$ is the feature vector.

Multi-task training method [20]. Stage-wise training method cannot achieve end-to-end training. By summarizing the shortcomings of RCNN, Fast RCNN uses softmax classifier instead of SVM classifier [65]. From the network structure analysis, each ROI outputs two feature vectors through two fully-connected layers, which input to the softmax classifier and bounding-box regressor in parallel. Finally, the classification and regression are integrated into a unified loss function, which is called the multi-task loss function. It combines classification loss and location loss, which improves the accuracy of the network model and implements end-to-end training of the network model. The multi-task loss function is defined as follows:

$$L\left(p, u, t^u, v\right) = L_{\text{cls}}(p, u) + \lambda[u \geq 1] L_{\text{loc}}\left(t^u, v\right) \tag{14}$$

$$L_{\text{cls}}(p, u) = -\log p_u \tag{15}$$

where, $p = (p_0, \cdots, p_k)$ is the probability distribution of $K + 1$ categories calculated by a softmax, and u is the true category. In L_{loc}, $t^u = \left(t_x^u, t_y^u, t_w^u, t_h^u\right)$ is the prediction offset to

true class u, which is converted by $t^k = \left(t_x^k, t_y^k, t_w^k, t_h^k\right)$ using the parameterization method. $v = \left(v_x, v_y, v_w, v_h\right)$ is the coordinates of the ground-truth of the true category u, where (x, y) represents the coordinate of the box center, (w, h) represents the width and height of the box. The Iverson bracket indicator function $[u \geq 1]$ can be used to eliminate the effects of background RoIs, when $u \geq 1$, the function is 1, otherwise it is 0. L_{loc} bounding-box loss function is shown as follows:

$$L_{\text{loc}}\left(t^u, v\right) = \sum_{i \in \{x,y,w,\text{h}\}} \text{smooth}_{L_1}\left(t_i^u - v_i\right) \tag{16}$$

In which

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{17}$$

The $\text{smooth}_{L_1}(x)$ combines the advantages of L_1 loss and L_2 loss. In L_1 loss and L_2 loss, x represents the difference between the target value and the predicted value. When $x > 1$, L_2 loss will increase the error, especially when there are outliers, the error will further increase, so very high penalty for large errors. Therefore, when $x > 1$, the L_1 loss with linear error increase can eliminate the sensitivity of the loss function to the outliers. When $x \leq 1$, the L_1 loss has no derivative points, which affect the convergence of the network model during the gradient descent, so the L_2 loss is selected. In short, $\text{smooth}_{L_1}(x)$ can eliminate the sensitivity to outliers, and the performance is more robust, which can avoid the situation of gradient explosion when RCNN or SPPNet is trained with L_2 loss.

The training region proposal network (RPN) [191] is required in the Faster RCNN. The loss function of the RPN is designed according to the multi-task loss function in the Fast RCNN. Different from the true category, only the binary category label is assigned to the anchor box in the RPN, indicating whether it is an object. With reference to equation (14), the loss function of the RPN is defined as follows:

$$L\left(\{p_i\}, \{t_i\}\right) = \frac{1}{N_{cls}} \sum_i L_{cls}\left(p_i, p_i^*\right) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}\left(t_i, t_i^*\right) \tag{18}$$

Here, p_i is the predicted probability of anchor i as the object. If the anchor is positive (object) then p_i^* is 1, if the anchor is negative (no object) then p_i^* is 0. The classification loss function L_{cls} reference to equation (4). Because of the specificity of RPN, make some minor changes to the bounding-box regression, which is defined as:

$$L_{reg}\left(t_i, t_i^*\right) = R\left(t_i - t_i^*\right) \tag{19}$$

R reference equation (15) in the definition of $smooth_{L_1}$. The improved parameterization method is used in the bounding-box regression of the anchor, which inherits the equation (12), but with minor differences. The definition method is defined as follows:

$$\begin{aligned} t_{\text{x}} &= (x - x_{\text{a}})/w_{\text{a}}, \quad t_{\text{y}} = (y - y_{\text{a}})/h_{\text{a}} \\ t_{\text{w}} &= \log(w/w_{\text{a}}), \quad t_{\text{h}} = \log(h/h_{\text{a}}) \\ t_{\text{x}}^* &= \left(x^* - x_{\text{a}}\right)/w_{\text{a}}, \quad t_{\text{y}}^* = \left(y^* - y_{\text{a}}\right)/h_{\text{a}} \\ t_{\text{w}}^* &= \log\left(w^*/w_{\text{a}}\right), \quad t_{\text{h}}^* = \log\left(h^*/h_{\text{a}}\right) \end{aligned} \tag{20}$$

Here, x, y, w, h represent the center coordinates of the anchor box, width and height respectively.

Unlike the two-stage detectors such as Fast/Faster RCNN, YOLO [187] is a single-stage detector. Therefore, the design of the loss function is different from the former. In the YOLO detector, a fully-image can output the bounding-boxes and class probabilities of

the objects through a single DCNN. It divides image into a $S \times S$ grid, and every grid is responsible for detecting the object of whose center falls into the grid cell. Each grid cell predicts B bounding-boxes and the confidence score of each bounding-box, and the confidence scores indicate the probability and accuracy of the bounding-boxes containing an object. The confidence score is defined as follows:

$$\text{Confidence score} = \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} \tag{21}$$

The Pr(Object) is 1 if it exists an object in the grid cell, and is 0 if there did not exist an object in the grid cell. $\Pr(\text{Class}_i | \text{Object})$ is the C conditional class probabilities to be predicted for each grid cell, which is for each grid cell, not B bounding-boxes. Finally, the test results yield the confidence scores of each bounding-box for class-specific objects, which as follows:

$$\Pr(\text{Class}_i | \text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}} \tag{22}$$

This gives us both confidence scores and accuracy for the class-specific of each bounding-box, as well as whether the bounding-boxes contain objects. In the training stage, although each grid cell can predict B bounding-boxes, only the bounding-box with the highest IoU of the object ground truth is used as the prediction. So, we define the multi-task loss function as follows:

$$\begin{aligned} Loss \;=\; & \lambda_{\mathbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(x_i - \hat{x}_i\right)^2 + \left(y_i - \hat{y}_i\right)^2 \right] \\ & + \lambda_{\mathbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2 \right] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i\right)^2 \\ & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i\right)^2 \\ & + \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c)\right)^2 \end{aligned} \tag{23}$$

where $\mathbb{1}_i^{\text{obj}}$ represents the grid cell i found the object, and $\mathbb{1}_{ij}^{\text{obj}}$ represents the $j\,th$ bounding-box in the grid cell i responsible for prediction. (x_i, y_i) is the center of the bounding-box that offsets from the grid cell boundary, (w_i, h_i) is the width and height of the bounding-box that normalized to the width and height of the input image. The purpose of the parameters λ_{coord} and λ_{noobj} are to reduce the impact of the grid cell that not object on the stability of the model.

4 The architectures of object detection

Object detection can be divided into object location and object classification. Since the DCNNs shows strong feature representation power [116, 122], the mainstream object detection architectures are based on DCNNs, which can be divided into two categories. One is the two-stage object detection architectures, which separate the object location task from the object classification task. It generates the region proposal first, and then classifies the region.

Table 3 Comparison of milestone two-stage object detectors and one-stage object detectors on the PASCAL VOC2007 test set

Method		Backbone network	Train set	Batch size	Input size	Proposals	Speed (fps)	GPU	mAP(%)	Published in
Two-stage object detectors	RCNN [66]	AlexNet	7	1	$\sim 1000 \times 600$	2000	0.1	Titan	58.5	CVPR14
		ZFNet	7	1	$\sim 1000 \times 600$	2000	0.07	Titan	59.2	
	SPPNet [81]	ZFNet	7	1	$\sim 1000 \times 600$	2000	2.6	Titan	60.9	ECCV14
	Fast RCNN [65]	VGGNet16	07+12	1	$\sim 1000 \times 600$	2000	0.5	Titan X	70	ICCV15
	Faster RCNN [191]	VGGNet16	07+12	1	$\sim 1000 \times 600$	6000	7	Titan X	73.2	NIPS15
		ResNet101	07+12	1	$\sim 1000 \times 600$	300	2.4	K40	76.4	
		ResNet101	07+12	1	$\sim 1000 \times 600$	300	5	Titan X	76.4	
		ZFNet	07+12	1	$\sim 1000 \times 600$	300	18	Titan X	62.1	
	RFCN [38]	ResNet101	07+12	1	$\sim 1000 \times 600$	300	9	Titan X	80.5	NIPS16
		ResNet101	07+12	1	$\sim 1000 \times 600$	300	5.8	K40	79.5	
One-stage object detectors	YOLOV1 [187]	VGGNet16	07+12	1	480×480	98	21	Titan X	66.4	CVPR16
		DarkNet19	07+12	1	480×480	98	45	Titan X	63.4	
	Fast YOLOV1 [187]	-	07+12	1	480×480	98	155	Titan X	52.7	
	YOLOV2 [188]	DarkNet19	07+12	-	480×480	-	59	Titan X	77.8	CVPR17
		DarkNet19	07+12	-	554×554	-	40	Titan X	78.6	
	SSD300 [150]	VGGNet16	07+12	1	300×300	8732	46	Titan X	74.3	ECCV16
		VGGNet16	07+12	8	300×300	8732	59	Titan X	74.3	
	SSD512 [150]	VGGNet16	07+12	1	512×512	24564	19	Titan X	76.8	
		VGGNet16	07+12	8	512×512	24564	22	Titan X	76.8	
	DSSD321 [59]	ResNet101	07+12	1	321×321	17080	9.5	Titan X	78.6	arXiv
		ResNet101	07+12	12	321×321	17080	13.6	Titan X	78.6	
	DSSD513 [59]	ResNet101	07+12	1	513×513	43688	5.5	Titan X	81.5	

Table 3 (continued)

Method	Backbone network	Train set	Batch size	Input size	Proposals	Speed (fps)	GPU	mAP(%)	Published in
	ResNet101	07+12	4	513 × 513	43688	6.6	Titan X	81.5	
FSSD300 [137]	VGGNet16	07+12+COCO	1	300 × 300	8732	65.8	1080Ti	82.7	arXiv
FSSD512 [137]	VGGNet16	07+12+COCO	1	512 × 512	24564	35.7	1080Ti	84.5	

'07' denotes PASCAL VOC2007 trainval, '07+12' denotes the union of PASCAL VOC2007 trainval and PASCAL VOC2012 trainval, '07+12+COCO' denotes the union of PASCAL VOC2007 trainval, PASCAL VOC2012 trainval, and MS COCO trainval35k. 'Batch size' is the test parameter

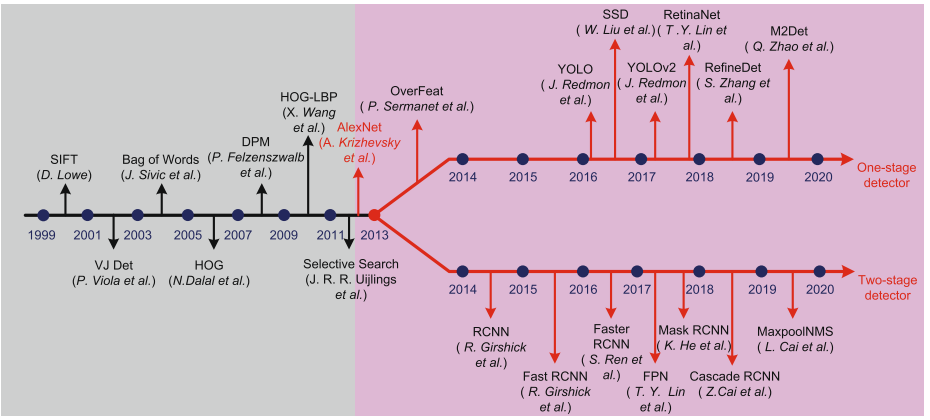

Fig. 6 The milestones of object detection evolution, in which AlexNet [116] serves as a watershed between traditional methods [58, 155, 208, 224, 229, 241] and DCNNs-based methods. The development of object detection based on DCNNs is mainly divided into one-stage detection architectures [142, 150, 187, 277, 283] and two-stage detection architectures [15, 16, 65, 66, 80, 144, 191]

The main advantage is the high detection accuracy and the main disadvantage is the slow detection speed. For example, RCNN [66], SPPNet [81], Fast RCNN [65], Faster RCNN [191], Mask RCNN [80] and RFCN [38] are all two-stage object detection architecture. Others are the one-stage object detection architectures that directly locates and classifies through DCNNs without separating into two parts. The one-stage object detection can generate the class probabilities and location coordinates of an object in a stage directly. It does not require the region proposal process, which is simpler than two-stage object detection. The main advantage is the high detection speed, but the detection accuracy is generally lower than two-stage object detection architecture. For example, OverFeat [197], YOLO series [187–189], SSD [150], DSSD [59], FSSD [137] and DSOD [199] belong to one-stage object detection. The performance parameters of some classic two-stage detectors and one-stage detectors are shown in Table 3. The milestones of object detection evolution are shown in Fig. 6. The highlights, properties, and shortcomings of the milestone object detection architectures are summarized in Table 4. The two-stage object detection architecture, the one-stage object detection architecture and the open source object detection platform are introduced below.

4.1 Two-stage object detection architecture

RCNN Inspired by AlexNet's great success in image feature extraction, R.Girshick et al. used DCNN as the feature extraction backbone network [116] instead of HOG [40, 58, 180], SIFT [155] and other traditional feature extraction algorithms, combined with the regional proposals algorithms (such as Selective Search [224], Objectness [1], category-independent object proposals [49], CPMC [19] and MCG [4]) to generate region proposals to form the RCNN (Regions with CNN features) [66] architecture, as shown in Fig. 7. From the pipeline of the RCNN architecture, the steps are as listed follows. Firstly, the selective search algorithm generates about 2000 category-independent region proposals. Secondly, the region proposals are inputted into the DCNN to extract 4096-dimensional feature as representation. Lastly, the features are classified using the SVM algorithm. Bounding-box

Table 4 Summarization of highlights, properties, and shortcomings of the milestone object detection architectures

Method	Highlights and properties	Shortcomings
RCNN [66]	Use DCNNs to extract image features; Use selection search algorithm to select 2k region proposals; Use SVM to classify regions; Use bounding box regressor to refine regions.	Training is too slow; Takes up a lot of space; No end-to-end training.
SPPNet [81]	Use DCNNs to extract the features of the entire image; Use selection search algorithm to extract 2k region proposals on the image, but map them to the feature maps; Use spatial pyramid pooling to input multi-scale image to DCNNs.	The use of selective search to extract region proposals is still slow; No end-to-end training.
Fast RCNN [65]	Use DCNNs to extract the features of the entire image; Use selection search algorithm to extract 2k region proposals on the image, but map them to the feature maps; Use the ROI Pooling layer to downsample the features of region proposals to obtain fixed-size feature maps; Use Multi-task loss function.	The use of selective search to extract region proposals is still slow; No end-to-end training.
Faster RCNN [191]	Use Region Proposal Network (RPN) to replace the selection search algorithm; The RPN shares feature maps with the backbone network; Can end-to-end training.	Poor performance for multi-scale objects and small objects; Detection speed cannot meet real-time requirements.
Mask RCNN [80]	Use ROIAlign pooling layer instead of ROI pooling layer, which improves detection accuracy; Combine training object detection and segmentation to improve detection accuracy; Conducive to small target detection.	Detection speed cannot meet real-time requirements.
FPN [144]	A multi-level feature fusion Feature Pyramid Network is proposed which is conducive to multi-scale object detection and small object detection.	Detection speed cannot meet real-time requirements.
YOLO [187]	Propose a novel single-stage detection network; Detection speed is fast and can meet real-time requirements.	Detection accuracy is not high, especially for dense objects and small objects.
YOLOV2 [188]	Multi-dataset joint training; new backbone network (DarkNet19); use k-means clustering algorithm to generate anchor box.	Complex training.

Table 4 (continued)

Method	Highlights and properties	Shortcomings
YOLOV3 [189]	Use multi-level feature fusion to improve the accuracy of multi-scale detection; New backbone network (DarkNet53).	As IoU increases, performance decreases.
SSD [150]	Multi-layer detection mechanism; Multi-scale anchors mechanism at different layers.	Not conducive to small object detection.
DSSD [59]	Multi-layer feature fusion mechanism; Up-sampling using deconvolution instead of simple linear interpolation; Improve the accuracy of small object detection.	Detection speed decreases relative to SSD.

regression and the greedy non-maximum suppression (NMS) [166] will be done to perform fine-tuning of bounding-box. In summary, RCNN has improved performance by 30% over traditional object detection algorithms [58, 192].

Although RCNN has pushed object detection into the era of neural networks, it still has three disadvantages that prevent application from entering the instance scenario. The three disadvantages are listed as follows:

(1). Each picture needs to be pre-fetched with about 2000 region proposals, which consumes a lot of storage spaces and I/O resources.
(2). In the case of using AlexNet [116] as the backbone network, the cropped/warped region block is deformed into 227×227 RGB image. This causes truncation or stretching of the object image to result in loss of object information.
(3). The separate extraction of each region proposals feature does not utilize the ability of DCNNs feature sharing, resulting in a large waste of computing resources.

SPPNet The process of cropping/warping the image in the RCNN is removed, and the spatial pyramid pooling (SPP) layer (similar to SPM [121]) is added after the last convolutional (Conv) layer. Therefore, the problem of missing object information caused by the cropped/warped image can be solved. Thus an image of arbitrary size can be input into the DCNNs to calculate 21-dimensional fixed-length feature vector for fully-connected (FC) layer [81]. The sharing of the entire image feature maps makes SPPNet test speed 10 to 100× faster than RCNN. The framework of SPPNet and RCNN is extremely same, so no end-to-end training is implemented. The convolutional (Conv) layer cannot be continued to train during fine-tuning in SPPNet, which limits the accuracy of network [65].

Fast RCNN By analyzing the disadvantages of SPPNet and RCNN in speed, space consumption, and training process. R.Girshick et al. proposed the RoI (Region of Interest) pooling layer, which is a single-level spatial pyramid pooling (SPP) [81]. In the Fast RCNN, the feature map of the image is calculated by the DCNNs. The selection search (SS) [224] algorithm is used to find the region proposals in the image and maps them onto feature maps. Then, the RoI pooling layer maps different feature regions to fixed-size feature vectors and inputs them to the fully-connected (FC) layer. Finally, the softmax predicts object categories and the bounding-box regression accurately locates the object location, and its architecture is depicted in Fig. 7. The Fast RCNN uses multi-task loss jointly train classification and

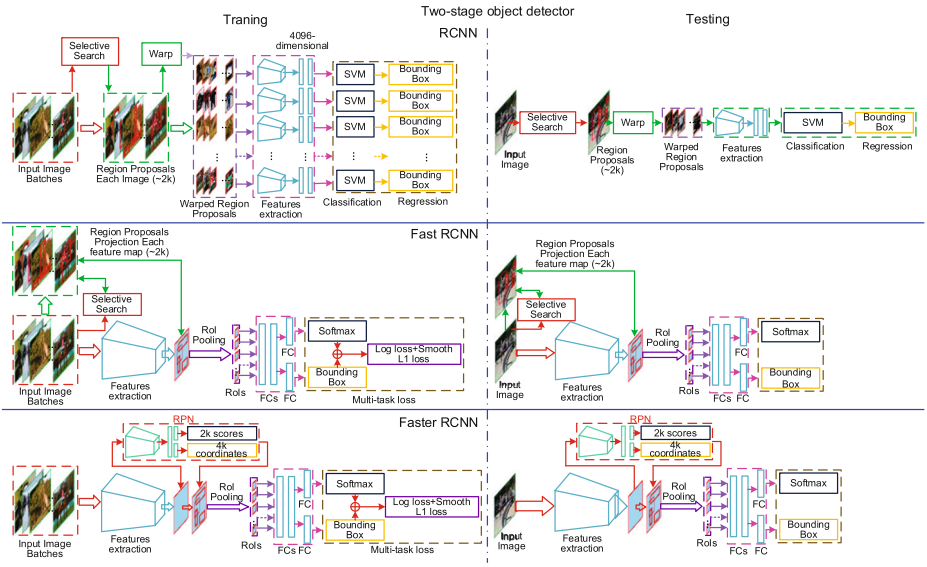

Fig. 7 The details of the classical two-stage object detectors include RCNN, Fast RCNN, and Faster RCNN. The left side is the training process and the right side is the testing process, and reflects the relationship between training and testing

bounding-box regression, so that two tasks share convolution features. Thus the stage-wise training of SVM+bounding-box regression (stage-wise training [66, 81, 294]) can be transformed into multi-task training [20, 177]. Because of these innovations, the advantages of Fast RCNN contrast with RCNN/SPPNet are as listed follows:

(1). Fast RCNN has higher accuracy than RCNN/SPPnet.
(2). Due to multi-task loss, the detector training is end-to-end.
(3). Fast RCNN training can update all network layers. It is superior to SPPNet that only updates the fully-connected (FC) layer.
(4). Hard disk storage is not required for feature caching.
(5). Training and testing are faster than RCNN/SPPNet.

Faster RCNN Although the Fast RCNN has a great improvement in accuracy and speed, the way to generate around 2000 region proposals/RoIs is the selective search algorithm [224, 225]. The selective search algorithm needs to search all the region proposals in the image and maps them into the feature maps, which is very time-consuming. In the test, Fast R-CNN took 2.3s to make predictions, 2s of which were used to generate 2000 RoIs. Therefore, the traditional region proposal algorithms [4, 114, 224, 225, 296] become the bottleneck of the object detection architecture. In order to solve this shortcoming, Shaoqing Ren et al. proposed a regional proposal network (RPN) in the Faster RCNN [191] (see Fig. 7) instead of the selective search algorithm to generate region proposals. The RPN integrates region proposals extraction into the DCNNs, which shares the convolution features of the full-image with the detection network. Since the region proposals share the convolution feature maps and the region proposal network (RPN) is implemented on the GPU, this process is nearly cost-free.

The RPN [191] is a fully convolutional network that is connected to the last convolutional layer of the backbone network. A feature map of any size is entered into the RPN, which outputs many rectangular object proposals with objectness scores. The RPN makes K predictions of different scales and aspect ratios for each sliding position in the feature map. Two fully-connected layers output $4 \times K$ coordinates through a box-regression layer (reg), and $2 \times K$ scores through a box-classification layer (cls). If the feature map size is $n \times n$, then output $n \times n \times k$ ROIs. RPN reduces the proposal number and improves the quality of regional proposals, so the location accuracy and speed of the object detection network (Common Datasets with only 300 proposals per image, 5fps on a GPU) get a big boost. In addition to RPN, researchers have proposed many region proposals based on DCNNs and have achieved good results in recent years, such as DeepBox [117], DeepMultiBox [50], SharpMask [178] and DeepProposal [63].

RFCN The algorithmic ideas and performances of the RCNN series determine the milestone in object detection. The architecture essentially consists of two subnets (Faster RCNN consists of three subnets, adding RPN [191]). The previous subnet is the backbone network for feature extraction, the next subnet completes the object detection: classification and location. The RoI pooling layer is inserted between two subnets, which converts the multi-scale feature map to the fixed-size feature map. But this step will destroy the translation invariance of the network, which is not conducive to the classification of the object. In order to balance the translational invariance and translational transformation. Jifeng Dai et al. proposed position-sensitive score maps in the RFCN [38], and the RFCN region detection is based on the fully convolutional calculation of the full-image. The RFCN shares fully-convolution and uses position-sensitive score maps/position-sensitive RoI pooling to blend the translational variance into the convolutional layers, which is beneficial for object location. All learnable layers have two characteristics: all of them are convolutional and shareable throughout the full-image; and they can encode spatial information for object detection. The author uses ResNet101 [82] as the backbone network, achieving 83.6% mAP on the PSCAL VOC 2007 testing set/82.0% mAP on the PASCAL VOC 2012 testing set, and faster than Faster R-CNN with ResNet101.

Mask RCNN Mask RCNN is an extension of Faster RCNN, which adds a Mask network branch for RoIs prediction segmentation parallel to object classification and bounding-box regression [80]. It can complete object detection and instance segmentation simultaneously. The Mask network is a streamlined version of the fully convolutional network used to generate split mask for each RoI. Due to the integer quantization of the RoI pooling [65, 191], the feature map region and the original image region is not aligned. Therefore, it produces a bias for accurately predicting pixel-level mask. Kaiming He et al. proposed a RoIAlign [80] layer instead of the RoI pooling layer, which uses bilinear interpolation to achieve pixel-level alignment. The RoIAlign also can improve the accuracy of the object detection branch. The experimental results show that the performance of Mask R-CNN using ResNet-101-FPN [82, 144] as the backbone network is much better than Faster RCNN with G-RMI. Mask RCNN got the championship of the COCO 2016 object detection challenge. Its AP is increased by 3 points compared to Faster RCNN with Inception-ResNet-v2-TDM [203, 212].

FPN Tsung-Yi Lin et al. proposed a Feature Pyramid Networks (FPN) in 2017 [144]. The FPN can be considered as a method of feature fusion and detection. Before FPN, the detectors mainly use top-level feature detection or independent detection in different feature

layers. These methods cannot take into account both classification information and location information, because the low-level semantic information are relatively small, but the object location information are rich, and the high-level semantic information are rich, but the object location information are relatively rough. Therefore, the FPN fuses different feature layers by way of top-down and lateral connection, and performs detection on the fused multi-layer feature layers. This method greatly enhances the detection performance. Faster RCNN with FPN achieves state-of-art results on the MS COCO dataset [144].

4.2 One-stage object detection architecture

OverFeat In 2013, Yann LeCun et al. proposed the famous OverFeat architecture [197], which utilized the feature sharing of DCNNs to integrate object classification and object location into one network architecture. The main idea of OverFeat is to extract the patch using the multi-scale fast sliding window [77] on the last pooling layer of DCNN. In order to predict the classification score for each patch and merge the patches according to the scores. In this way, the complex shape and multi-size problem of the object image are solved. OverFeat uses the classification and regression of DCNN to achieve the object classification and object localization. It is compared with RCNN [66], OverFeat has obvious advantages in speed, but lacks in accuracy.

YOLO Although Faster RCNN uses RPN to reduce the number of region proposals from around 2,000 (RCNN/Fast RCNN) to around 300, there are still overlaps between region proposals [191]. The inevitable overlap can lead to repetitive computation, making it difficult for the object detection architecture to break through the bottleneck of speed. In 2015, J.Redmon et al. proposed YOLO (Your Only Look Once), which is an end-to-end single neural network [187]. It can implement class probabilities and bounding-boxes regression directly from a full-image. YOLO divides the full-image into $S \times S$ grids. Each grid cell is responsible for the detection of the object center falling into the grid cell. Each grid cell predicts C class probabilities, B bounding-boxes and confidence scores, and the full-image is encoded to output $S \times S \times (5B + C)$ tensor. Figure 8 shows the architecture of YOLO.

The YOLO detection system consists of 24 convolutional layers and 2 fully-connected layers with a network entry of 448 × 448 image. In this paper, the network is trained using the PASCAL VOC dataset, so set S=7, B=2, C=20, and the final prediction code is a $7 \times 7 \times 7$ tensor. Up to 45 fps on the testing set, the speed has reached the requirements of real-time image processing, but YOLO has the following shortcomings to be improved:

(1). YOLO does not work well for dense small objects because a grid cell can only predict two bounding-boxes and can only belong to the same class.
(2). The generalization ability is weak for the case where a new aspect ratio occurs in the same type of object in the testing image.
(3). The shortcoming of loss function affects the detection effect.

YOLO9000/V2 Although YOLO achieves real-time object detection, it has a number of localization errors and low recall. In order to achieve higher accuracy, YOLOV2 [188] has the following improvements over YOLOV1:

(1). The introducing of batch normalization [97] speeds up network convergence and enhances network generalization capability.
(2). Train high resolution classifiers to accommodate higher resolution images [161].

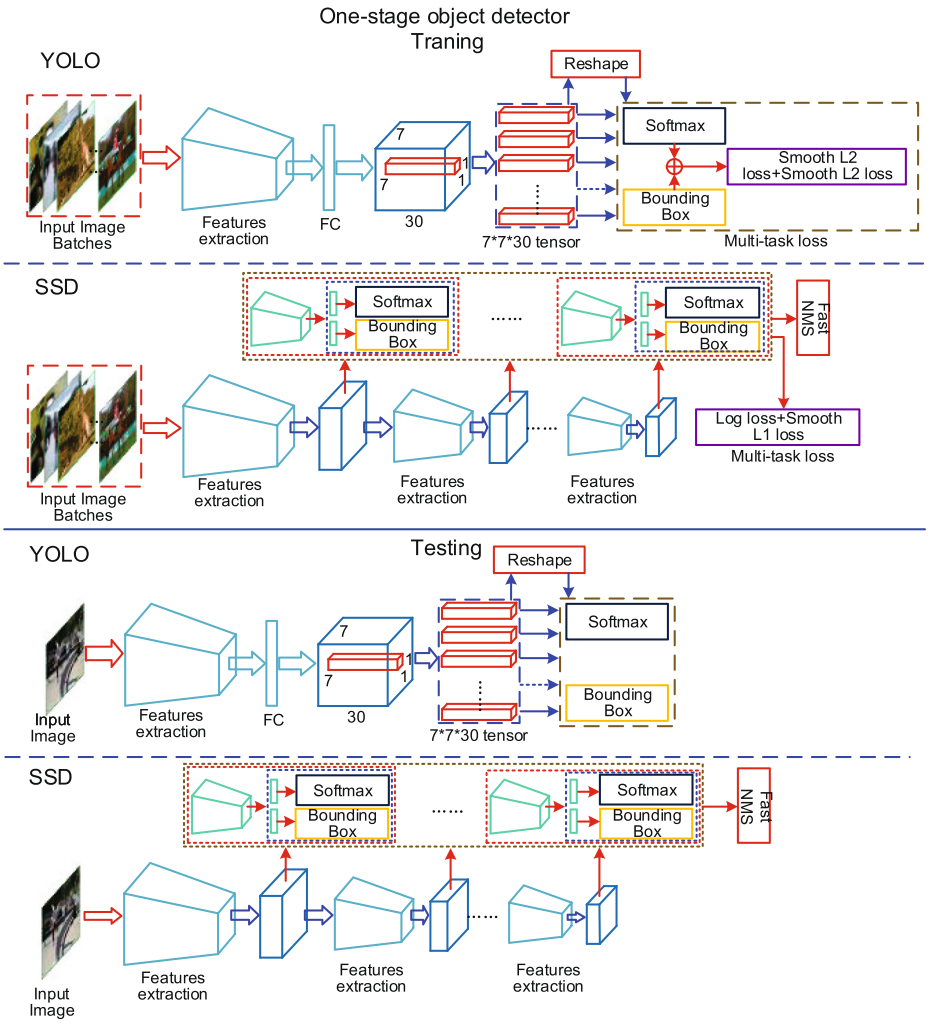

Fig. 8 The details of the classical one-stage object detectors include YOLO and SSD. The upper part is the training process and the lower part is the testing process, and reflects the relationship between training and testing

(3). In order to solve the weak generalization ability for various aspect ratios objects of YOLO. The idea of anchor in the Faster RCNN [191] is introduced in YOLOV2, and each grid cell can predict 3 scales and 3 aspect ratios.
(4). YOLOV2 uses the K-means [231] clustering algorithm to automatically find the prior bounding-boxes, which can make detection performance better.
(5). YOLOV2 limits the offset of ground truth relative to the grid cell coordinate between 0 and 1, which solves the instability of network model.

In order to achieve faster speed, authors proposed a Darknet-19 [186] backbone network based on the VGGNet [207]. The authors use the joint training mechanism of classification datasets (ImageNet [42]) and object detection datasets (COCO [143]) to train the network.

Therefore, the network learned more categories of object, the trained network is called YOLO9000.

YOLOV3 YOLOV3 [189] inherits the ideas of YOLOV1 and YOLOV2/9000, and has improved their shortcomings to achieve balance between speed and accuracy. To achieve this goal, the authors combine the residual block [82], feature pyramid network (FPN) [144], and binary cross entropy loss to upgrade YOLO to YOLOV3. These changes make the detection network suitable for more complex objects (more categories, multi-size objects).

SSD The RCNN series and YOLO have their own advantages in speed and accuracy. The RCNN series has high detection accuracy, but the speed is slow. Although the YOLO has fast detection speed, the generalization ability of the object with large dimensional change and the detection effect for small objects is weak. By drawing on the advantages of Faster RCNN and YOLO, Wei Liu et al. proposed Single Shot MultiBox Detector (SSD) [150]. SSD uses VGG16 as the backbone network for feature extraction, which replaces FC6/FC7 with Conv6/Conv7, and then add four convolutional layers (Conv8, Conv9, Conv10 and Conv11). The design idea of SSD network is hierarchical extraction of features, we can see from Fig. 8. The one-stage network is divided into 6 stages. Each stage extracts feature maps of different semantic levels and performs object classification and bounding-box regression. The multi-scale feature maps combined with the anchor mechanism in the Faster RCNN can adapt the SSD to the detection task of multi-scale objects. The accuracy of SSD512 with VGG16 is significantly better than Faster R-CNN, which is 3 times faster. The SSD300 runs at 59 fps faster than YOLO, with significant quality detection [150].

DSSD and FSSD In order to improve the ability of SSD to express low-level feature maps, DSSD [59] uses ResNet101 as the backbone network. The addition of deconvolution modules and skip-connection [82] enhance the representation of the low-level feature maps and achieve a certain degree of feature fusion. Similarly, FSSD combines low-level features into high-level features based on SSD, which significantly improves the accuracy.

4.3 Open source object detection platform

With the progress of research work on object detection technology in recent years, a large number of excellent object detection architectures based on DCNNs are proposed. These object detection architectures on a scalable, unified platform allows researchers to quickly and easily conduct experimental research. In order to meet the researchers' demand for high-quality and high-performance object detection platform. Facebook AI Research (FAIR) Institute opens up an object detection platform called Detectron [53]. Google opens up an object detection API System [217]. CUHK&SenseTime Joint Lab opens up an object detection library named Mmdetection [27]. These open source platforms integrate many landmark object detection architectures, backbone networks, a large number of benchmark results, and pre-trained models in the Model Zoo library. Summarization of the respective characteristics of the three open source platforms in Table 5.

5 Complex problem of object detection

Object detection encounters many obstacles in real-world scene applications. There are crowded occlusion, small objects detection, class imbalance, multi-scale object detection,

Table 5 Summarization of open source object detection platforms

No.	Object detection platform	Object detection algorithms	Backbone networks	Datasets	Deep learning framework	Source code	Teamwork
1	Detectron [53]	Mask R-CNN RetinaNet Faster R-CNN RPN Fast R-CNN R-FCN	ResNeXt ResNet VGG16 ResNet with FPN ResNeXt with FPN	COCO PASCAL VOC Cityscapes [35]	Caffe2 [13]	Python	FAIR
2	Mmdetection [27]	RPN Fast R-CNN Faster R-CNN Mask R-CNN Cascade R-CNN SSD RetinaNet	ResNet50-FPN ResNet101-FPN	COCO PASCAL VOC	PyTorch [179]	Python	CUHK & SenseTime
3	Object Detection API [217]	SSD R-FCN Faster RCNN Mask R-CNN	ResNet50 ResNet101 Inception V2 MobileNetV1 MobileNetV2 Inception-ResNet-v2	COCO PASCAL VOC KITTI [2] Open Images [113] AVA v2.1 [71] iNaturalist Species [227]	TensorFlow [219]	Python	Google

and redundant detection. Researchers propose lots of methods to respond to the challenges of these issues. The solutions make the object detection technology based on DCNNs to take an important step in practical applications.

5.1 Dense occlusion

Dense occlusion problem often occurs in pedestrian detection [45], autonomous driving [24, 31], and other practical application scenarios. It is divided into two situations, which are occlusion between objects of the same categories and the occlusion between objects of the different categories. Occlusion can lead to object information loss, such as missed detection and false detection. In traditional object detection algorithms, researchers can use the additional object information. Such as gray information, boundary information, and local features to overcome dense occlusion problem. This review paper focuses on methods based on DCNNs to deal with dense occlusion problem.

Influenced by the idea of Generative Adversarial Nets (GANs) [67], Xiaolong Wang et al. proposed A-Fast-RCNN by combining the GANs with the Fast RCNN. Based on GANs, the authors design an Adversarial Spatial Dropout Network (ASDN) [243], which can generate occlusion samples to train detection network. Through this step, it could make the detection network more robust to occlusions. ASDN is only a generator in the network, which plays a role in training process and does not participate in the testing process. A-Fast-RCNN improves robustness of detection network to occlusion objects by adding network modules. In recent researches, Face++ first solves crowed pedestrians detection from the perspective of loss function, which is named Repulsion Loss (RepLoss) [242]. This method is also suitable for generic object detection that is inspired by magnetic attraction and exclusion. The loss function consists of three parts: the attraction item L_{Attr}, the exclusion item L_{RepGT}, and the exclusion item L_{RepBox}, which are defined as follows:

$$L = L_{\text{Attr}} + \alpha * L_{\text{RepGT}} + \beta * L_{\text{RepBox}} \tag{24}$$

where weighting factors α and β balance L_{RepGT} and L_{RepBox}. Among the equation, the attracting item L_{Attr} denotes the loss between the predicted bounding-box and ground truth. $P = (l_P, t_P, w_P, h_P)$ are the coordinates of the proposal, $G = (l_G, t_G, w_G, h_G)$ are the coordinates of the ground truth, and l, t, w, and h represent the coordinates of the left-top points of the boxes and the widths and heights, respectively. Set $\mathcal{P}_+ = \{P\}$ represents all positive proposals, and set $\mathcal{G} = \{G\}$ represents all ground truths in one image. The representation of L_{Attr} is presented as follows:

$$G^P_{Attr} = \arg\max_{G \in \mathcal{G}} IoU(G, P) \tag{25}$$

$$L_{\text{Attr}} = \frac{\sum_{P \in \mathcal{P}_+} \text{Smooth}_{L1}\left(B^P, G^P_{Attr}\right)}{|\mathcal{P}_+|} \tag{26}$$

where $P \in \mathcal{P}_+$, B^P is the predicted box regressed from proposal P.

The exclusion item L_{RepGT} represents the loss of the predicted bounding-box and the ground truth of the adjacently same class. The representation is defined as follows:

$$G^P_{Rep} = \arg\max_{G \in \mathcal{G} \setminus \{G^P_{Attr}\}} IoU(G, P) \tag{27}$$

$$IoG(B, G) \triangleq \frac{area(B \cap G)}{area(G)}, IoG(B, G) \in [0, 1] \tag{28}$$

$$\text{Smooth}_{ln} = \begin{cases} -\ln(1-x) & x \leq \sigma \\ \frac{x-\sigma}{1-\sigma} - \ln(1-\sigma) & x > \sigma \end{cases} \tag{29}$$

$$L_{\text{RepGT}} = \frac{\sum_{P \in \mathcal{P}_+} \text{Smooth}_{ln}\left(IoG(B^P, G^P_{Rep})\right)}{|\mathcal{P}_+|} \tag{30}$$

where $IoG(B^P, G^P_{Rep})$ is the overlap between B^P and G^P_{Rep}, which is defined in the equation (28). $Smooth_{ln} \in (0, 1)$ is a continuously differentiable function, and $\sigma \in [0, 1)$ is used to adjust the sensitivity of the repulsion loss to the outliers.

The exclusion item L_{RepGT} represents the loss of the predicted bounding-box and the ground truth of the adjacently different class. The representation is defined as follows:

$$L_{\text{RepBox}} = \frac{\sum_{i \neq j} \text{Smooth}_{ln}\left(IoU\left(B^{P_i}, B^{P_j}\right)\right)}{\sum_{i \neq j} \mathbb{1}\left[IoU\left(B^{P_i}, B^{P_j}\right) > 0\right] + \epsilon} \tag{31}$$

where $\mathcal{P}_+$ is divided into $|\mathcal{G}|$ disjoint subsets based on each proposal object $\mathcal{P}_+ = \mathcal{P}_1 \cap \mathcal{P}_2 \cap \ldots \cap \mathcal{P}_{|\mathcal{G}|}$. Randomly sample from two different subsets for two proposals, $P_i \in \mathcal{P}_i$ and $P_j \in \mathcal{P}_j$ where $i, j = 1, 2, \ldots, |\mathcal{G}|$ and $i \neq j$. B^{P_i} and B^{P_j} indicate predicted bounding-boxes. ϵ is a small constant to prevent division by 0, $\mathbb{1}$ is an identity function.

A-Fast-RCNN and RepLoss prove the feasibility of solving the dense occlusion problem from the network architecture and the loss function. Then, Shifeng Zhan et al. proposed OR-CNN [276], which was published in ECCV2018. Based on Faster RCNN [191], OR-CNN adds compactness loss term based on regression loss. It is named Aggregation Loss, which can reduce the missed detection of overlapping objects. At the same time, the use of Part Occlusion-aware RoI Pooling Unit instead of RoI Pooling can reduce the impact of occlusion position on global features. The experimental results show that the results of these three algorithms in the Citypersons dataset [275] are satisfactory results.

5.2 Multi-scale objects detection

Multi-scale object detection is one of the most complicated difficulties in the field of object detection. The convolutional neural network is a typical hierarchical structure. Each layer abstracts the feature map of the image, and semantic information represented by the feature maps are different. This inherent property determines the detection of multi-scale objects based on DCNNs. For example, the RCNN [66] and the YOLO [187] only perform object classification and bounding-box regression on the last layer of feature maps. This leads to much loss of object feature representation information, which is obviously not conducive to the detection of multi-scale objects. For example, YOLO is not robust to small object detection, which is caused by the loss of small object features extracted in the last convolutional layer.

The combination of information fusion and hierarchical structures of DCNNs, researchers proposed the idea of multi-layer feature fusion and multi-layer detection to solve the problem of multi-scale object detection. By summarizing the relevant works, researchers have many achievements around this difficult problem, and many classical network architectures are proposed. These include FPN [144], SNIP [204], FSSD [137], SPPNet [81], Hypercolumns [76], HyperNet [112], ION [7], RON [111], SSD [150], DSSD [59], DSOD [199], MSCNN [14], RBFNet [148], RefineDet [277], STDN [288], DES [281], PFPNet [107], and so on.

According to the idea of multi-layer feature fusion, Hypercolumns fuses each layer of feature maps to obtain hypercolumn that is all nodes of the network corresponding

to the pixels and they connected in series as feature vectors. The hypercolumn contain more detailed location and classification information. Similarly, the skip layer feature is applied in ION [7] and HyperNet [112]. In HyperNet, the multi-level feature maps are extracted through the skip connections [82]. The multi-scale hyper-feature obtained by fusion includes high-level and low-level semantics, which improves the robustness of multi-scale object detection.

What about the idea of multi-layer detection? Due to the different representations of the semantic features of each layer, different feature maps can be used to detect different scales objects. The SSD [150] detects the feature maps generated by the last 6 convolutional layers, and finally uses NMS for post processing. Songtao Liu et al. introduced a Receptive Field Block (RFB) [148] in the SSD architecture. It uses multi-branch structure composed of convolutional layers of different sizes and dilated convolution [261, 262] to increase the receptive field. Finally, the convolution outputs of different sizes and dilation ratios are concatenated. The results prove that RFB can improve the robustness of detection. MSCNN [14] is an optimization based on Faster RCNN in detecting multi-scale objects, especially the small object detection. The multi-scale object proposal network is mainly composed of the multi-level network structure, which is the core of the MSCNN.

Multi-layer feature fusion and multi-layer detection can also be combined to improve the robustness of multi-scale object detection. In FPN [144], the bottom-up is down-sampling and the top-down is up-sampling. Then, it uses the lateral connection method to merge the down-sampling and the up-sampling feature map. Finally, it uses each merged layer to make prediction. In DSSD and FSSD, skip connections and feature fusion are also used to transform the architecture of SSD.

5.3 Class Imbalance

One of the reasons why the accuracy of the one-stage object detection is lower than the two-stage object detection is class imbalance. The region proposal of the two-stage object detection can effectively prevent class imbalance [142]. The class include hard positive example, hard negative example, easy positive example, and easy negative example. The number of hard examples is less than easy examples. Representative methods for solving this problem include OHEM [202], Focal Loss [142], CC-Net [169], and RON [111]. The main idea of OHEM is to screen out hard examples as training samples. The Focal Loss changes the weight of the examples through the loss function, which makes the network pay more attention to the training of hard examples. The CC-Net uses a cascaded network structure to process a large number of easy examples in previous stage and a small number of hard samples in latter stage. The RON uses the objectness prior map [111] to distinguish the foreground and background of the object, which keeps the positive and negative examples at certain percentage. Finally, the network training can be completed in combination with the Loss strategy.

Among them, the Focal Loss points out a new idea to solve the problem of class imbalance, which is from the perspective of loss function rather than network structure. Through the analysis of the one-stage detector, there is a problem of hard and easy (positive and negative) class imbalance, which results in low detector accuracy. Kaiming He et al. solved this problem from the perspective of loss function and proposed the Focal Loss, which is modified based on the cross entropy (CE) loss function.

5.4 Post-processing of redundant bounding-boxes

The post-processing of the redundant bounding-boxes is to de-duplicate the repeated bounding-boxes, and select the most accurate bounding-boxes as the results. The de-duplication operation of the bounding-boxes can be used for the intermediate process of object detection and the process of the final results. For example, Faster RCNN uses non-maximum suppression (NMS) to reduce the number of region proposals to 300 in the final step of the Region Proposal Network (RPN) [191]. This process reduces the amount of computation and does not compromise the final detection accuracy. YOLO and SSD use NMS to generate ultimate object results. NMS is the original algorithm for eliminating duplication. Inspired by the NMS algorithm, UMIACS proposed a Soft-NMS algorithm [10]. Based on the NMS and Soft-NMS algorithms, CMU and Face++ proposed a Softer-NMS algorithm [84]. Face++ published a IoU-Net [100] in 2018, which makes up for the shortcoming of NMS that uses only the classification confidence of bounding-box as the deduplication threshold. It introduces a localization confidence of bounding-box into the non-maximum suppression process, which can further improve the location accuracy of bounding-box.

5.4.1 NMS

The core idea of non-maximum suppression (NMS) [10, 166] is to use local maxima to suppress non-maximum values, and only retain the maxima. In the object detection, each bounding-box gets a classification and confidence after being classified. However, there will be overlaps in the bounding-boxes. In this case, NMS is needed to select the highest bounding-boxes in those neighborhoods and suppress the bounding-boxes with low confidence. The algorithm description is in Fig. 9a.

Although NMS [10] has a good promoting effect on the de-duplication of bounding-boxes, there are missed detection on the detection of overlapping (dense occlusion) objects, and the threshold of the algorithm is hard to set. The re-scoring function of the NMS algorithm is defined as follows and the parameters in the equation (32) are defined in Fig. 9a.

$$s_i = \begin{cases} s_i, & \text{iou}\,(\mathcal{M}, b_i) < N_t \\ 0, & \text{iou}\,(\mathcal{M}, b_i) \geq N_t \end{cases} \tag{32}$$

5.4.2 Soft-NMS

In order to deal with some of the shortcomings of NMS, UMIACS proposed the Soft-NMS algorithm [10] (described in Fig. 9b), which is published in ICCV2017. In the de-duplication process of the redundant bounding-boxes, the Soft-NMS does not delete (suppress) the bounding-boxes when the IoU is higher than a certain threshold. But decay the classification confidence of bounding-boxes to enter the next iteration. For decaying the classification confidence, the authors proposed a linear penalty function and a Gaussian penalty function. They are defined as follows and the parameters in the equation (33, 34) are defined in Fig. 9b.

$$s_i = \begin{cases} s_i, & \text{iou}\,(\mathcal{M}, b_i) < N_t \\ s_i\,(1 - \text{iou}\,(\mathcal{M}, b_i)), & \text{iou}\,(\mathcal{M}, b_i) \geq N_t \end{cases} \tag{33}$$

$$s_i = s_i e^{-\frac{\text{iou}(\mathcal{M}, b_i)^2}{\sigma}}, \forall b_i \notin \mathcal{D} \tag{34}$$

(a)

Input : $\mathcal{B} = \{b_1, \cdots, b_N\}$, $\mathcal{S} = \{s_1, \cdots, s_N\}$, N_t
$\mathcal{B}$ is the list of initial detection bounding-boxes
$\mathcal{S}$ contains corresponding detection confidence
N_t is the NMS threshold
Begin
$\mathcal{D} \leftarrow \{\}$
while $\mathcal{B} \neq empty$ **do**
$\mathrm{m} \leftarrow \mathrm{argmax}\, \mathcal{S}$
$\mathcal{M} \leftarrow b_m$
$\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{M}; \mathcal{B} \leftarrow \mathcal{B} - \mathcal{M}$
for $\boldsymbol{b_i}$ *in* $\mathcal{B}$ **do**
if $iou(\mathcal{M}, b_i) \geq N_t$ **then**
$\mathcal{B} \leftarrow \mathcal{B} - b_i; \mathcal{S} \leftarrow \mathcal{S} - s_i$
end
end
end
return $\mathcal{D}, \mathcal{S}$
end

(b)

Input : $\mathcal{B} = \{b_1, \cdots, b_N\}$, $\mathcal{S} = \{s_1, \cdots, s_N\}$, N_t
$\mathcal{B}$ is the list of initial detection bounding-boxes
$\mathcal{S}$ contains corresponding detection confidence
N_t is the NMS threshold
Begin
$\mathcal{D} \leftarrow \{\}$
while $\mathcal{B} \neq empty$ **do**
$\mathrm{m} \leftarrow \mathrm{argmax}\, \mathcal{S}$
$\mathcal{M} \leftarrow b_m$
$\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{M}; \mathcal{B} \leftarrow \mathcal{B} - \mathcal{M}$
for $\boldsymbol{b_i}$ *in* $\mathcal{B}$ **do**
$s_i \leftarrow s_i f(iou(\mathcal{M}, b_i))$
end
end
return $\mathcal{D}, \mathcal{S}$
end

(c)

Input : $\mathcal{B} = \{b_1, \cdots, b_N\}$, $\mathcal{S} = \{s_1, \cdots, s_N\}$, $\mathcal{C} = \{\sigma_1^2, \cdots, \sigma_N^2\}$, N_t
$\mathcal{B}$ is the list of initial detection bounding-boxes
$\mathcal{S}$ contains corresponding detection classification confidence
$\mathcal{T}$ contains corresponding detection localization confidence
N_t is the NMS threshold
Begin
$\mathcal{D} \leftarrow \{\}$
$\mathcal{T} \leftarrow \mathcal{B}$
while $\mathcal{T} \neq empty$ **do**
$m \leftarrow \mathrm{argmax}\, \mathcal{S}$
$\mathcal{M} \leftarrow b_m$
$\mathcal{T} \leftarrow \mathcal{T} - \mathcal{M}$
$idx \leftarrow iou(\mathcal{M}, b_i) \geq N_t$
$\mathcal{M} \leftarrow \mathcal{B}[idx] / \mathcal{C}[idx] / sum(\mathcal{C}[idx])$
$\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{M}$
end
return $\mathcal{D}, \mathcal{S}$
end

(d)

Input: $\mathcal{B} = \{b_1, \cdots, b_n\}, \mathcal{S}, \mathcal{T}, N_t$
$\mathcal{B}$ is a set of detected bounding boxes.
$\mathcal{S}$ and $\mathcal{T}$ are functions (neural networks) mapping bounding-boxes to their classification confidence and IoU estimation (localization confidence) respectively.
N_t is the NMS threshold.
Output: $\mathcal{D}$, the set of detected bounding boxes with classification scores.
Begin
$\mathcal{D} \leftarrow \emptyset$
while $\mathcal{B} \neq \emptyset$ **do**
$b_m \leftarrow \mathrm{argmax}\, \mathcal{T}(b_j)$
$\mathcal{B} \leftarrow \mathcal{B} \setminus \{b_m\}$
$s \leftarrow \mathcal{S}(b_m)$
for b_j *in* $\mathcal{B}$ **do**
if $iou(b_m, b_j) \geq N_t$ **then**
$s \leftarrow max\left(s, \mathcal{S}(b_j)\right)$
$\mathcal{B} \leftarrow \mathcal{B} \setminus \{b_j\}$
end
end
$\mathcal{D} \leftarrow \mathcal{D} \cup \{(b_m, s)\}$
end
return $\mathcal{D}$
end

Fig. 9 The pseudo code of the post-processing of the redundant bounding-boxes, where **a** is NMS, **b** is soft-NMS, **c** is softer-NMS, and **d** is IoU-guided NMS

5.4.3 IoU-Net

NMS and Soft-NMS sort bounding-boxes by the classification confidence, but the localization confidence of the bounding-boxes is not utilized. The lack of localization confidence may result in more accurate bounding-boxes being suppressed during NMS process. Therefore, Face++ proposed an IoU-Net [100], which introduces localization confidence into the NMS process. The IoU of the predicted bounding-box and the ground truth bounding-box is used as the localization confidence and sorting criterion instead of the classification confidence in the NMS process. Then the clustering-like rule is used to update the classification confidence, which is called IoU-guided NMS, and it is depicted in Fig. 9d.

5.4.4 Softer-NMS

Softer-NMS [84] introduces KL loss, Gaussian distribution, and Delta distribution into the bounding-box regression. To obtain the IoU and variances of the four coordinates of the

predicted bounding-box, which is called Bounding-Box Regression with KL Loss. The standard deviation of the bounding-box can be obtained by the variance of the coordinates, which can be used as the localization confidence. During the suppression process, Softer-NMS performs a weighted average of the localization confidence of the bounding-boxes above a certain threshold based on soft-NMS. This method can solve the problem that the classification confidence and the localization confidence are not positively correlated. The algorithm is described in Fig. 9c.

5.5 Detection speed

The speed of object detection is an important indicator of performance and is critical for applications. The purpose of the lightweight backbone networks mentioned in Section 2.2 is to speed up the speed of object detection. The compression technology of the network model can also improve the speed of object detection. The detailed technologies include network model pruning, network model quantification and network model distillation.

Network model pruning is a model compression method that cuts unimportant convolutional layers or connections to reduce the parameters of the model. Therefore, the core of pruning is to find unimportant convolutional filters or connections. Hao Li et al. proposed a method based on the weight value [125]. First, the absolute value of the weight of each filter is summed in the convolution, and then filter with the lowest value is deleted, which can reduce the parameters of the model. However, Jian-Hao Luo et al. believe that it is difficult to determine the importance of filter by the weight value. Therefore, a pruning method based on entropy is proposed, which uses entropy to determine the importance of filter [156]. Gwanak-Gu proposed to take random pruning method and then count the performance of each model to determine the local optimal pruning method [3]. Tien-Ju Yang et al. used the energy consumption of each layer to determine which layer to prune [258]. The layer with high energy consumption is selected for pruning, and the weight value is used to assist the judgment.

Network model quantification to reduce the code length of the weight. For example, 32 bit floating-point number can be represented by 1 bit floating-point number, which reduces the parameters. At present, model binarization is main research direction of quantification. For example, Binarized Neural Networks [249] not only quantify the weight value to 1 bit on the basis of BinnaryConnect, but also change the activation value to 1 bit. That is to reduce the memory consumption, and also simplify many multiply-accumulate operations into a bitwise operation XNOR-Count. Similarly, XNOR-Net [185] binarizes both the weight value and the activation function, which achieves 32× storage compression with a 58× speed increase.

Network model distillation is a kind of migration learning that uses a pre-trained complex model (called teacher model) to train a simple model (called student model) to achieve the effect of compressing. For the detection speed, Guobin Chen et al. used highly sophisticated detector models as teacher models to guide the learning process of efficient student models [25]. Yu Hao et al. proposed a new end-to-end object detection architecture that combines incremental learning with model distillation, which improves detection speed and improves detection accuracy [74].

5.6 Small object detection

Small object detection is one of the difficulties in object detection. The improvement of small object detection will promote the development of related applications, such as automatic driving, remote sensing image detection, industrial defect detection and medical image detection. At present, some classic detection methods (such as Faster RCNN, YOLO, SSD) are not ideal for detecting small objects. By analyzing the characteristics of small objects, there are mainly the following reasons:

(1). The small objects occupy small pixel size in the original image, which results in less feature information for detection. (Generally defined object resolution is less than 32×32 is small object)
(2). After the original image is extracted by the down-sampling of the backbone network, the information of the small objects location may be lost.
(3). Small objects are less in the original image, resulting in imbalances between small objects and medium or large objects.

In recent years, researchers propose some technical solutions for small object detection. Mate Kisantal et al. proposed a small object data augmentation method on the MS COCO dataset by over-sampling the image containing small objects and the copy-pasting strategies of small objects [160]. Tsung-Yi Lin et al. proposed a feature pyramid network, which improves the performance of multi-scale detection through multi-layer feature fusion and multi-layer detection, and also enhanced small object detection [144]. Jianan Li et al. proposed Perceptual Generative Adversarial Networks (GANs) to improve small object detection [128]. Perceptual GANs mine structural associations between objects of different scales to improve the feature representation of small objects. Chenyi Chen et al. redesigned the anchor sizes and introduced a context model based on Faster RCNN to improve small object detection. Jun Wang et al. proposed H-CNN to detect small ship objects in synthetic aperture radar (SAR) images [234]. Yantao Wei et al. proposed a multiscale patch-based contrast measure method for infrared small target detection, which effectively suppresses the interference of background clutter [244]. Liangkui et al. proposed a 7-layer DCNN to achieve automatic extraction of small object features and end-to-end suppression of clutter [139]. The problem of small object location loss can be solved by the combination of high-level features and low-level features, such as DSSD [59], Feature-Fused SSD [17], MDSSD [253] are based on this structure.

6 Datasets and evaluation criteria

The evaluation criteria on the dataset is the standard for evaluating the performance of the network. Table 6 summarizes the performance results of some object detection methods on the MS COCO dataset over the past five years. The datasets and evaluation criteria are summarized below.

6.1 Datasets

The training of DCNNs is inseparable from various image datasets. The datasets play an indelible role in the development of DCNNs. Whether it is image recognition, object detection or instance segmentation, each subject has its unique methods and specificity difference. Excellent datasets are important players in the current impressive progress in object

Table 6 Detection results of the detectors published in the top conferences or journals on the MS COCO test-dev(%) in recent years. *S*: small objects, *M*: medium objects, *L*: large objects

Method	Backbone network	Train set	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Published in
OHEM [202]	VGGNet16	trainval	22.6	42.5	22.2	5	23.7	37.9	CVPR16
ION [7]	VGGNet16	train	23.6	43.2	23.6	6.4	24.1	38.3	CVPR16
NoC [190]	ResNet101	train	27.2	48.4	27.6	-	-	-	TPAMI16
MPN [265]	VGGNet16	train	33.2	51.9	36.3	13.6	37.2	47.8	BMVC16
Faster RCNN w FPN [144]	ResNet101-FPN	trainval35k	36.2	59.1	39	18.2	39	48.2	CVPR17
TDM [203]	Inception-ResNetv2	trainval	37.3	57.8	39.8	17.1	40.3	52.1	CVPR17
RON320 [111]	VGGNet16	trainval	44.7	22.7	23.6	-	-	-	CVPR17
DeNet101(wide) [223]	ResNet101	trainval	33.8	53.4	36.1	12.3	36.1	50.8	ICCV17
CoupleNet [295]	ResNet101	trainval	34.4	54.8	37.2	13.4	38.1	52	ICCV17
RetinaNet [142]	ResNeXt101-FPN	trainval35k	40.8	61.1	44.1	24.1	44.2	51.2	ICCV17
Mask RCNN [80]	ResNeXt101	trainval35k	39.8	62.3	43.4	22.1	43.2	51.2	ICCV17
DSOD300 [199]	DS/64-192-48-1	trainval	29.3	47.3	30.6	9.4	31.5	47	ICCV17
SMN [29]	VGGNet16	trainval35k	31.6	52.2	33.2	14.4	35.7	45.8	ICCV17
SIN [151]	VGGNet16	train	23.2	44.5	22	7.3	24.5	36.3	CVPR18
STDN513 [287]	DenseNet169	trainval	31.8	51	33.6	14.4	36.1	43.4	CVPR18
RefineDet512+ [277]	ResNet101	trainval35k	41.8	62.9	45.7	25.6	45.1	54.1	CVPR18
D-RFCN + SNIP [32]	DPN98	trainval	45.7	67.3	51.1	29.3	48.8	57.1	CVPR18
Cascade R-CNN [16]	ResNet101	trainval	42.8	62.1	46.3	23.7	45.5	55.2	CVPR18
MLKP [233]	ResNet101	trainval35	28.6	52.4	31.6	10.8	33.4	45.1	CVPR18
RFBNet512-E [148]	VGGNet16	trainval35k	34.4	55.7	36.4	17.6	37	47.6	ECCV18
CornerNet511 [120]	Hourglass104	trainval35k	42.2	57.8	45.2	20.7	44.8	56.6	ECCV18
PFPNet-R512+ [107]	VGGNet16	trainval35k	39.4	61.5	42.6	25.3	42.3	48.8	ECCV18
SNIPER [204]	ResNet101	trainval35k	46.1	67	51.6	29.6	48.9	58.1	NIPS18

Table 6 (continued)

Method	Backbone network	Train set	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Published in
M2Det800 [283]	VGGNet16	trainval35k	44.2	64.6	49.3	29.2	47.9	55.1	AAAI19
R-DAD-v2 [5]	ResNet101	trainval35k	43.1	63.5	47.4	24.1	45.9	54.7	AAAI19
Libra R-CNN [172]	ResNeXt101-FPN	train	43	64	47	25.3	45.6	54.6	CVPR19
ExtremeNet [289]	Hourglass104	train	43.7	60.5	47	24.1	46.9	57.6	CVPR19
ScratchDet300+ [290]	Root-ResNet34	trainval35k	39.1	59.2	42.6	23.1	43.5	51	CVPR19
NAS-FPN1280 [62]	AmoebaNet-DropBlock	trainval35k	48.3	-	-	-	-	-	CVPR19
FSAF [260]	ResNeXt-64x4d-101-FPN	trainval35k	42.9	63.8	46.3	26.6	46.2	52.7	CVPR19
Cas-RetinaNet [270]	ResNet101	trainval35k	41.1	60.7	45	23.7	44.4	52.9	BMVC19
TridentNet [133]	ResNet101-Deformable	trainval35k	48.4	69.7	53.5	31.8	51.3	60.3	ICCV19
CenterNet511 [47]	Hourglass104	trainval35k	47	64.5	50.7	28.9	49.9	58.9	ICCV19
DAFS512 [132]	ResNet101	trainval35k	38.6	58.9	42.2	17.2	42.2	54.8	ICCV19
FRCNN-FD-WT [175]	ResNet101	trainval35k	42.1	63.4	45.7	21.8	45.1	57.1	ICCV19
RPDet [259]	ResNet101-DCN	trainval35k	46.5	67.4	50.9	30.3	49.7	57.1	ICCV19
FCOS [221]	ResNeXt101-64×4d-FPN	trainval35k	44.7	64.1	48.4	27.6	47.5	55.6	ICCV19
FreeAnchor [280]	ResNeXt101	train	44.8	64.3	48.4	27	47.9	56	NeurIPS19
NATS [176]	ResNeXt101-32×4d	train	41.6	64.3	45.2	24.9	45.5	54.8	NeurIPS19

detection. In the past 20 years, many excellent datasets have emerged. Due to the excellent performance of these datasets, some of them have become important indicators in the industry to measure the performance of algorithms. Relevant personnel revise and expand the datasets to enrich them. This can provide better support for our DCNNs and promote the continuous development of the field. For example, the release of the ImageNet dataset [42, 195] forms a unified standard for the training and testing of supervised learning-based convolutional neural networks. The training on the dataset makes it easy to form a unified evaluation standard and eliminate the differences in the local datasets. The increase in the number and quality of images in the datasets can certainly promote the development of deep learning.

The image datasets are abundant, ranging from about MB to TB and can be applied in object detection, image classification, and instance segmentation. They include MINST [123], CIFAR-10 [115], CIFAR-100 [115], ImageNet [42], Open Images [113], MS COCO [143], PASCAL VOC [51, 52], SUN [251], Tiny Images [222], and Places [286]. Figure 10 shows example images of four commonly used datasets.

The entry-level MNIST dataset is widely used for training and testing in the field of machine learning due to its single objects and small images. The classical LeNet is designed for the MNIST dataset [123]. Similar to the MNIST dataset, CIFAR-10 and CIFAR-100 are with moderate size, which is about 170MB. CIFAR is a great choice for image classification algorithms. These two datasets are a subset of Tiny Images in a very large size. CIFAR-10 is more diversified than MNIST handwritten dataset, but it only has ten categories, which has caused certain limitations. Therefore, researchers sort out CIFAR-100 based on it. CIFAR-100 is similar to CIFAR-10, which has 100 classes and 600 images per class. They abstract 20 superclasses from these 100 classes. So each image has two labels, a class label and a superclass label. It is worth mentioning that CIFAR-10 dataset is completely mutually exclusive between classes.

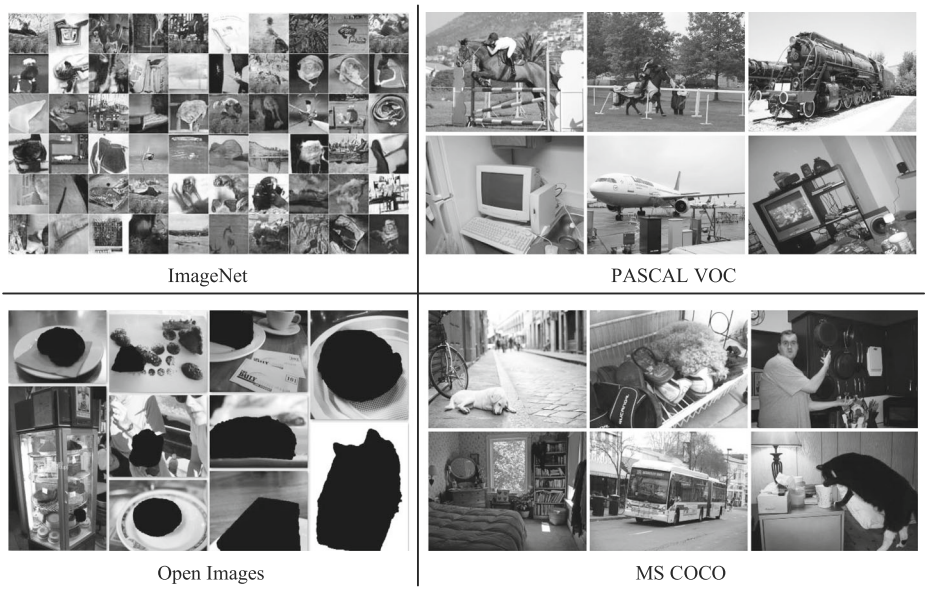

Fig. 10 Image examples of ImageNet, PASCAL VOC, Open Images and MS COCO

The larger Caltech datasets were released by the California Institute of Technology. Caltech datasets mainly include two categories, Caltech-101 (131MB) [56] and Caltech-256 (1.2GB) [70]. Caltech-101 has 102 classes, with 9145 images, which includes 101 normal image categories and a miscellaneous class (background class). Usually when in use, the last of the miscellaneous classes is removed. Caltech-256 is similar to Caltech -101, which contains 30607 images.

The PASCAL VOC dataset(2GB) [52], which pioneers the algorithm competition, provides a set of standardized datasets for image recognition, segmentation, and classification. The training set is presented as labelled images. Objects in the dataset include 4 categories, which are people, animals, vehicles, and indoor objects. The training set consists of a set of images. Each image corresponds to its annotation file (bounding box and the class label) one to one. Multiple objects of different categories may exist in one image. PASCAL VOC dataset has good image quality and complete labels, which is very suitable for evaluating algorithm performance. The proportion of training set and testing set is about 1:1. The category distribution in the image is also same. The SUN dataset(7GB) [251], focuses on object detection and scene recognition, which consists of 908 scene categories and 4,479 object categories. 313,884 objects are labeled with background.

The MS COCO dataset(40GB) [143] appears to be very large compared to the datasets above, which is sponsored by Microsoft. For image annotation information not only have a category and location information, and semantic text description of the image. Like ImageNet in the field of image classification and detection, MS-COCO dataset has become a yardstick in the evaluation of algorithm performance in the field of visual semantic understanding. Google's open source contains trained models are based on MS COCO dataset. The dataset is aimed at scene understanding and contains 91 classes, 328,000 images, and 2,500,000 labels. The MS COCO category is less comparable with ImageNet and SUN. But it has more images of each type, which makes it easier for the model to acquire stronger ability to analyze a certain type of object in a certain scene during training. The MS COCO dataset contains more images and classes than the PASCAL VOC mentioned earlier.

For another example, ImageNet dataset [42] is at TB level and widely applied in the visual field. Many researches in computer vision field are carried out based on ImageNet, such as image classification, object detection. It is maintained by a dedicated team. It has detailed dataset documentation, and is very easy to use. Some annotation problems of ImageNet will also be centrally fixed once a year and reissued, and the latest version is recommended. ImageNet is even praised as the benchmark of algorithm performance evaluation in the computer vision. It contains more than 14 million images and covers more than 20,000 categories. The well-known ILSVRC image classification and object detection challenge [195] is based on the ImageNet dataset.

Google released a dataset in 2016, which is named Open Images [113]. It has about 9 million images and about 6,000 category labels. Open Images became a new data support for the computer vision community to develop new models. At the same time, the large amount of image data in Open Images can guarantee the complete training of deep network model. Earlier, Google announced Open Images V4. It contains 15.4 million bounding-boxes for 600 categories on 1.9 million Images and is the largest existing dataset with object location annotations. Simultaneously the ECCV 2018 image challenge was held. The annotation work is heavily participated by professional staff, so as to guarantee the high accuracy and consistency of dataset annotation. In addition, the categories of images in Open Images are also quite diverse, and most scene images are complex scenarios that contain multiple objects. The overall size of the dataset is 1.5GB, because Open Images only provides the URLs of the images.

6.2 Evaluation criteria

How to evaluate the performance of the algorithms on the unified datasets is an important issue. At present, there are three main performance evaluation criteria: Precision, Recall, and Frame Rate. There are also other metrics, including IoU, PR curve, ROC curve, and AUC curve. They are closely related to each other.

IoU (Intersection-over-Union) is an important concept in object detection. It is the ratio of the intersection of the predicted bounding-boxes and the ground-truth bounding-boxes of the object and the union of the two. The IoU can also be seen as the similarity of the above two sets, expressed by the Jaccard index. Therefore, IoU can be used to measure the accuracy of object detection. If the IoU of object is closer to 1, the accuracy of detection is higher.

Based on the IoU, performance evaluation is measured by Precision and Recall. To obtain the Precision and the Recall of network model, the statistical True Positives (TP) and False Positives (FP), True Negatives (TN) and False Negatives (FN) in the validation dataset/testing dataset are needed. Here, IoU is used to determine whether the test results are correct. Assuming that the goal of the model is to detect cats from the dataset, it is necessary to determine whether the "cat" detected by the model is true (TP) or false (FP). Set $IoU = 0.5$ to determine whether the result is correct or not. When $IoU > 0.5$, the result is considered to be correct. Otherwise, it is wrong. In PASCAL VOC, the $IoU = 0.5$, while MS COCO competition changes the IoU to a composite value between 0.5 and 1.0. According to $IoU = 0.5$, the number of correct detections and the number of error detections can be obtained. Therefore, the precision and recall of each category of one image can be calculated. The formula is defined as follows:

$$P_{C_{ij}} = \frac{TP_{C_{ij}}}{TP_{C_{ij}} + FP_{C_{ij}}} \tag{35}$$

$$\text{Recall}_{C_{ij}} = \frac{TP_{C_{ij}}}{TP_{C_{ij}} + FN_{C_{ij}}} \tag{36}$$

where $P_{C_{ij}}$ represents the Precision of category C_i in the $j\ th$ image, while $\text{Recall}_{C_{ij}}$ represents the Recall of category C_i in the $j\ th$ image.

The Average Precision (AP) of the category C_i can be calculated:

$$AP_{C_i} = \frac{1}{m} \sum_{j=1}^{m} P_{C_{ij}} \tag{37}$$

There are multiple categories $\{C_1, C_2, \cdots, C_n\}$ for the dataset. Therefore, the mean Average Precision (mAP) of the entire category can be calculated, as follows:

$$mAP = \frac{1}{n} \sum_{i=1}^{n} AP_{C_i} \tag{38}$$

In the actual scene, it may has multiple categories of object detection. Therefore, mAP is used to describe the detection performance of the model for all object categories.

To more accurately evaluate the performance of the detector, Precision and Recall are used to construct the Precision-Recall curve. With the Recall value increasing, Precision can maintain a high level, which indicates that the detector performance is better. In the case of poor performance, the detector needs to sacrifice a lot of Precision to keep Recall at a high level. The use of Precision and Recall in the PR curve indicate that the curve is

more concerned with the positive case in the datasets. Since positive case predictions are the primary focus of the test, PR curves are widely used in many papers.

There is a Receiver Operating Characteristic (ROC) curve corresponding to the PR curve. The difference is that the ROC curve uses FPR (False positive rate) and TPR (True positive rate). The closer the ROC curve is to the upper left corner, the better the performance of the detector is. The coordinate (0, 1) represents the probability that the positive case is all arranged before the counterexample, which is the ideal state. If it is necessary to more intuitively indicate the quality of the detector, the area under the ROC curve (AUC) can be introduced. The value of AUC is the size of the area under the ROC curve. Typically, the AUC value is between 0.5 and 1.0, and a larger AUC value represents better performance.

When the proportion of positive and negative samples in the testing set is uneven, the ROC curve can describe the performance of detector more stably, but the PR curve is greatly affected under the same conditions. Therefore, ROC curve is a more balanced evaluation method than PR curve. Therefore, the choice of PR curve or ROC curve in application should be made according to actual needs. Here are some suggestions:

(1). If it is necessary to evaluate the overall performance of the detector, the ROC curve should be used.
(2). In the case of uneven categories, and if you want to remove the impact of category distribution on performance evaluation, the ROC curve should be used.
(3). If you need to test the impact of different categories on performance of the detector, the PR curve should be used.
(4). Most applications should focus on the detection of positive examples in the same category. Therefore, the PR curves should be used.
(5). When the category is imbalanced, the ROC curve tends to give a more optimistic result due to a higher tolerance for counterexamples.

7 Applications

As one of the three basic tasks of computer vision, object detection has a wide range of applications in real-world scenarios. In real-world application scenarios, object detection differs in technology implementation depending on the specific tasks. Important applications of object detection are reviewed in this section, including face detection [240], salient object detection [11], pedestrian detection [45], remote sensing image detection [193] and medical image detection [145].

7.1 Face detection

Face detection is the most important application area for object detection, which is the basis of face recognition, face alignment, gender recognition, and sentiment analysis. In real-world, face detection is a challenging detection task due to changes in face features, illumination, gestures, and occlusion.

The purpose of face detection is to determine whether there are faces and find location of the faces in pictures. The traditional face detection is mainly based on the sliding window and the handcrafted feature extractor, and the face template feature is used to perform sliding matching with the detected image feature to determine the position of the face. The representative method is the VJ detection algorithm designed by Viola and Jones in 2001 [230]. It uses Haar features and cascaded AdaBoost classifiers to construct the detector,

which greatly increases detection speed and accuracy. In addition, ACF [256] and DPM [57] also increase the performance of face detection. However, traditional face detection algorithms still have many problems. With the advent of the deep learning era, face detection based on deep learning shows strong performance.

The deep learning-based object detection algorithms achieve great success in general object detection, and many face detection algorithms evolve from these general object detection algorithms. Haoxiang Li et al. proposed a Cascade CNN [126], which contains multiple cascaded DCNN classifiers to solve the problem of sensitivity to illumination and angle in real-world scenarios to some extent. Zhang et al. proposed a multi-task face detection algorithm that uses a cascaded architecture like the Cascasde RCNN, called MTCNN [273]. It integrates face detection and face key detection into a framework in three parts. Similar to Cascade CNN, Faceness-Net [257] is a coarse-to-fine detection process that uses multiple DCNN-based network classifiers to detect faces. For improving the performance of the classifier, Hao Wang et al. proposed a Face RCNN based on the Faster RCNN and added center loss based on softmax [101]. In response to small objects and multi-scale problems in face detection, Peiyun Hu et al. proposed a hybrid-resolution model that processes image pyramids in a scale-invariant manner and uses a scaled hybrid detector [90]. M. Najib et al. proposed a SSH [165] to implement multi-scale face detection by detecting on different scale feature maps. Small objects and multi-scale face detection can also be handled by anchor strategies, such as FaceBoxes [278], S3FD [279] and ScaleFace [75].

7.2 Salient object detection

The role of salient object detection is to highlight the main object regions in the image, also known as the salient regions. As an important application of computer vision and object detection, salient object detection is widely used in image understanding, video understanding, computer graphics and robot navigation.

In the era of non-deep learning. Itti et al. proposed the earliest saliency model based on center-surround mechanisms to detect spatially discontinuous objects in the scene [98]. Liu et al. proposed a method of replacing saliency detection with binary segmentation, which promoted the development of saliency object detection [149]. Yu et al. proposed to determine the background score of each region based on the observations of the background and the salient regions [264].

Since DCNNs have strong feature representation capabilities, the introduction of DCNNs into salient object detection is a trend. R. Zhao et al. proposed an MCDL architecture [284] to extract local and global contexts based on multi-layer perceptron (MLP), and then classify the foreground and background. Multi-layer perceptron-based methods include superCNN [83], MAP [271], LEGS [235], and MDF [124]. Although the multi-layer perceptron-based methods improve in performance, it is insensitive to spatial information and is very time consuming. Currently, the most advanced salient object detection methods are based on the full convolutional networks. L. Wang et al. proposed a recurrent fully convolutional networks (RFCN) to further improve the detection performance [236]. P. Hu proposed a deep network set to generate a compact and uniform saliency map to distinguish pixels from the object boundary [91]. Based on the Deeplab algorithm, J. Zhang et al. proposed a DUS to learn the potential saliency and noise patterns through several pixel-supervised methods of heuristic saliency methods [272]. Specific detailed summary can also refer to the review [239].

7.3 Pedestrian detection

Pedestrian detection is widely used in intelligent surveillance, autonomous driving and robotic navigation. The problems faced by pedestrian detection are more complicated than general object detection. Because pedestrian objects have the characteristics of rigid and flexible objects at the same time, it is more susceptible to the influence of posture, dense occlusion, illumination and viewing angle, so the difficulty of pedestrian detection is also higher than that of face detection.

Traditional pedestrian detection is based on handcrafted feature extractors similar to traditional face detection. For example, HOG + SVM pedestrian detection algorithm proposed by Navneet Dalal et al. in CVPR2005 [39], which uses the orientation and intensity information of edge to describe the shape and appearance of the pedestrian.

DCNNs are more suitable for pedestrian detection because the feature representation ability of multi-layer nonlinear mapping of DCNNs is stronger than that of traditional handcrafted feature extractors. Faster RCNN is not well for pedestrian detection, Zhang et al. analyzed that the resolution of the feature map is too low, and there is no hard negative examples mining. Therefore, Zhang et al. used the RPN to process small objects and hard negative examples based on Faster RCNN, and then used random forests to classify proposal regions [274]. Also based on Faster RCNN, Jiayuan Mao et al. proposed a HyperLearner to enhance the recognition of pedestrians and backgrounds by modifying the scales of anchors [159]. For the multi-scale problem of pedestrians, Jianan Li et al. design two sub-networks for parallel detection based on the large-scale and small-scale differences, and then use the scale-aware to merge the two sub-networks [127]. In order to solve the occlusion problem, Yonglong Tian et al. proposed a DeepParts [220], which divides the human body into multiple parts for detection and then combines them. For occlusion problems, special loss functions can also be used, such as Repulsion Loss [242] proposed by Xinlong Wang et al. and Aggression Loss [276] proposed by Zhang et al..

7.4 Remote sensing image detection

Remote sensing image detection is mainly used in military reconnaissance, land and resources survey, urban planning and traffic navigation. The objects are varied, including aircraft, ships, vehicles, roads, airports, ports and various buildings. Remote sensing image detection mainly has the following difficulties:

(1). The large view of remote sensing images lead to high image resolution, which put high demand on the speed of object detection.
(2). The large view results in smaller objects size relative to image, so small object detection is also difficult for remote sensing images.
(3). The objects in natural images are mostly horizontal. While the remote sensing images are is taken overhead, the rotation invariance of the object is an important issue.
(4). The background of remote sensing image is quite complex.

At present, remote sensing image detection based on deep learning is devoted to solving these problems. Adam Van Etten proposed YOLT [226] based on high-speed YOLOv2, which improves the speed of high-resolution remote sensing image detection through two detections. At the same time, in order to solve the small object detection problem, the resolution of feature map is increased. Yang Long et al. proposed an unsupervised score-based bounding box regression (USB-BBR) combined with non-maximum suppression to optimize the bounding box and enhance the ability to locate small objects [153]. Jiangmiao Pang

et al. proposed an R^2-CNN to enhance the detection of small object remote sensing images by introducing attention mechanisms [173]. Li Ke et al. deal with rotational invariance by adding multi-angle anchors to the RPN [130]. Gong Cheng et al. proposed a rotation invariant layer to deal with the rotation invariance [33]. Chen Wang et al. proposed an end-to-end multi-scale visual attention network (MS-VANs) [232]. The core idea is to learn a visual attention network for the feature map of each scale, in order to highlight the objects and suppress the backgrounds.

7.5 Medical image detection

Medical image detection can assist doctors in accurately analyzing the lesion area, greatly improving the accuracy of medical diagnosis, and reducing the manual workload of doctors. Currently, the most abundant open dataset for medical images is https://grand-challenge.org/challenges/

Aryan Mobiny et al. applied the capsule network [119] to the detection of lung cancer, which improves the detection speed. Li et al. introduced the attention mechanism based on the DCNNs and applied it to glaucoma detection [131]. Kawahara et al. proposed multi-stream CNN to classify skin lesions, each of which works on images of different resolutions [106]. Kong et al. proposed a combination of LSTM-RNN and CNN to detect end-diastolic and end-systolic frames in the MRI (Magnetic Resonance Imaging) image of the heart [110]. Hwang et al. proposed a weakly supervised deep learning method to detect lesions in nodules and mammograms in chest X-rays [95].

8 Conclusions

This review paper makes a comprehensive and detailed review of the deep learning based object detection methods. It includes three aspects, which are backbone networks, detection architectures and loss functions. At the same time, it also presents a detailed analysis of the complex problems. Starting from the problems, we sum up the outstanding solutions emerging in recent years. We introduce the evaluation metrics and datasets, and summarize the important applications of the object detection. It is worth mentioning that we summarize the current open source platforms for the object detection, which aims to help researchers for choosing appropriate platform. Finally, we give some potential developing directions of the object detection.

From the developing tendency of the backbone network, we can see that the direction is to increase the number of the network layers and the number of the neurons in each layer. Researchers' pursuit of performance makes the network larger and larger, leading to explosive growth of network computing. At the same time, gradient dispersion and gradient explosion may occur, which is obviously unacceptable. In order to solve this problem, researchers usually adopt two methods: one is to use appropriate rules in the middle layer to remove some neurons, reduce the number of parameters, and to some extent avoid the occurrence of over-fitting [195, 209]; second, the convolution kernel decomposition method is used to replace the larger convolution kernel with the smaller convolution kernel. This can adjust the network depth and enhance the characterization of the network while keeping the receptive field constant and reducing the number of parameters [34, 212–214]. It is worth mentioning that ResNet [82] effectively solves the degradation problem by means of residual learning. Based on the differences between object detection and image classification,

Zeming Li et al. proposed the DetNet for object detection tasks, which makes the deep network more specialized. Moreover, it solves some deficiencies of non-dedicated network in the field of object detection and significantly improves the performance. InceptionV2 [97] and InceptionV3 [214] also provide us with new ideas from the perspective of batch processing and network splicing. In the actual application scenario, due to resource constraints, lightweight networks arise at the right moment. An important issue is how to reduce the computational load without losing performance. Overall, backbone networks of the object detection architecture account for 90% of computation and storage. Therefore, the design of efficient and lightweight backbone networks will be a key research direction in the future.

In this review paper, we summarize the one-stage and two-stage detectors at the level of network architecture. The two-stage architectures first generate the object region proposals, and then perform regression and classification on the region proposals, such as Faster RCNN [191]. It is better than the one-stage detectors in accuracy, such as YOLO [187]. But the speed of the two-stage architecture is significantly lower than the one-stage detectors. In practice, accuracy and speed need to be balanced. Generally, when the accuracy reaches a certain level, the costs brought by continuing improving the accuracy are unacceptable. In some fields, the real-time requirements make the speed of the object detection framework more important than the accuracy.

The Loss function plays an important role in deep learning and machine learning. In recent works, many innovative methods have been proposed based on the loss function, which includes the stage-wise loss function [66, 81] and the multi-task loss function [65, 191]. Multi-task loss function can achieve end-to-end training. The accuracy is higher than pipeline multi-stage loss function. It is worth mentioning that the Focal Loss [142] and Repulsion Loss [242] are respectively used to solve the complex problems of class imbalance and dense occlusion. Therefore, although some classic loss functions can usually be used, constructing more appropriate loss function will make the learning framework more robust.

At present, the development of object detection and even the whole artificial intelligence field is mainly data-driven. The continuous efforts of scholars, open source datasets have become very rich. Some standard datasets have become the benchmark for the performance of competition evaluation models. However, in the era of expanding application scenarios and big data explosion, there are relatively few datasets with good labels. Manual labeling of image training datasets can cost huge labor and time. Therefore, how to label data more effectively and use fewer samples for effective learning are the key issues in the field. The current limited learning is far from meeting the needs. How to enable the model to quickly and accurately detect objects is an important research direction in this field. However, the performance of unsupervised learning [18, 48, 78, 154, 183] is still unsatisfactory at present. How to use small and medium sized data for supervised learning to achieve high-precision results will be a hot topic. In addition, it is feasible to reduce the supervision cost by using the weak labels of image data. Based on the pre-training of large detection datasets [79, 206, 255], multi-category supervised learning is carried out on this basis. Based on this, researchers have made many constructive achievements in recent years: (1) Data enhancement and learning of small sample was first proposed by Feifei Li et al., which is called **one shot learning** [55]. To solve this problem, researchers focus on Siamese network [9] again. Siamese network [12] is a simple and powerful networks. In the field of face detection and recognition, Gregory Koch et al. combined one shot learning with Siamese neural network and converted the one-shot problem into the verification problem in image recognition by

learning the feature similarity between image pairs [109]. In addition, Hao Chen et al. proposed a new low-shot transfer detector (LSTD) [26]. They set up a kind of effective object domain detector by using rich source domain knowledge and few training samples. The design and implementation of LSTD can unify the advantages of SSD [150] and RCNN [66] into one deep framework. (2) The semantic information of known categories is used to detect unknown categories, which is known as **zero shot learning** [184]. This problem is obviously more challenging than one shot learning, which enables the machine to have certain reasoning ability. However, there are several problems in the domain of zero shot learning. The first is Domain shift problem. Since the feature dimension of the sample is often larger than the semantic dimension, the establishment of mapping often lose useful information. In order to preserve more information, the sample is usually mapped to the semantic space and reconstructed. The second is Hubness problem. Zero shot learning uses KNN, so hubness problem maybe appear. If ridge regression method is used, hubness problem can be aggravated. To alleviate the hubness problem, a mapping relation from semantic space to feature space can be established additionally [200]. The above method can get 76.5% accuracy on AWA dataset [184]. Also, generative model can be used to solve this problem, such as self-encoder, GAN [67]. It transforms the problem into supervised classification problem and avoid KNN operation. If there is semantic gap, the flow pattern formed by samples in feature space is inconsistent with that formed by categories in semantic space. Therefore, by adjusting their flow pattern to keep them consistent, then learning the mapping between them [134]. (3) **Incremental Learning** [21] is used to continue learning new data based on pre-trained models. Meanwhile, it ensures that the learned object detection ability is not lost. Therefore, the major problem in incremental learning is Catastrophic knowledge [201], balancing the relationship between new knowledge and old knowledge.

In the future, more efficient detection frameworks can improve the real-time and accuracy of embedded detection applications, which will make the application of object detection more wide. At present, due to the limited use of contextual information, it is necessary to consider how to organically combine such information with more refined object instance segmentation for future development. The research on the object detection of 3D image and depth image (RGB-D) are still scarce, which requires more attentions. We are optimistic about the development of this field. Finally, we present several promising directions based on the existing object detection technologies.

Object detection of high resolution images and videos Current object detection method are mainly proposed for small and medium size images. Mingfei Gao et al. introduced a general framework [60] to reduce calculation cost of object detection. At the same time, the accuracy of the detection of different scales object in high resolution image were significantly improved. In this architecture, the R-net uses coarse detection results to predict the potential accuracy gain of an analysis region at higher resolution. Then Q-net continues to select regions for amplification. Experiments on the Caltech Pedestrians dataset [45, 46] show that their approach reduces pixel calculation by more than 50% while maintaining high detection accuracy. The detection results are also verified on the YFCC100M [167] high-resolution testing dataset.

AR and VR scene detection At present, AR and VR technologies are widely used in the game industry, the film industry, and the military industry. Object detection and scene detection are also a very interesting direction [68, 164, 211]. For example, human visual and imaging devices can be overlapped in augmented reality scenarios. The object detection

combines the augmented reality scenarios can enable human-in-loop detection and control modes.

Depth image (RGB-D) detection At present, RGB-B image adds the depth information and is widely used in autonomous driving and robot vision. In the field of autonomous driving, object detection of RGB-D image [30, 31, 72, 118, 210] is very important. Accurate assessment of the location and orientation of different objects can directly affect scene understanding, motion state adjustment, and path planning. Recently, Martin Simon et al. proposed a Complex-YOLO [205], a network that detects 3D object point clouds in real-time. The network extends YOLOv2 [188] through a specific complex regression strategy to estimate multi-category 3D bounding-boxes in Cartesian space.

Video stream object detection It is mainly applied to mobile devices or security monitoring devices. The video stream object detection [85, 102–104] must consider not only the information of each frame, but also the relationship between each frame. Due to the limitations of computing resources, a reasonable network architecture must be designed to meet the accuracy, speed and storage space requirements of real-time video stream object detection. In the actual application scenario, it is also necessary to consider the redundant feature information between adjacent frames, frame blur or jitter and crowed occlusion. Therefore, the current object detection algorithms are difficult to obtain better results for video stream. To solve these problems, Evan Shelhamer et al. proposed a method to multiplex the features of the previous frame. Then, these features are sent to the lesser-computed parts to calculate the final features [198]. Xizhou Zhu et al. achieved good results by using optical flow to improve the speed and accuracy of video recognition [292, 293]. Based on this improvement, Xizhou Zhu et al. proposed a lightweight network structure, which is suitable for mobile video object detection. A lightweight object detector is applied to the sparse key frames, and the entire network can be trained end to end. The system achieves 60.2% mAP on mobile phone at 25.6 fps [291]. For the case where the object appearance deteriorates in some video frames, a typical solution is to enhance the characteristics of each frame by aggregating adjacent frames. However, due to the motion of the object and the viewpoint, the object features usually cannot be spatially corrected across frames. To meet this challenge, Wang et al. proposed an end-to-end full motion-aware network (MANet) model [238]. It uniformly calibrates the features of objects at the pixel level and the instance level in a unified framework, which obtains a good result on the ImageNet VID dataset.

Contextual information In a real-world scenario, the human visual system recognizes the object not only according to the properties of the object itself, but also according to the contextual information. The contextual information includes the background information, the environmental information, and the relationship between the objects. These information can improve the robustness of the detector for multi-scale objects, occlusion objects, and fuzzy objects. In recent years, some researches also make some substantial progress in this area [28, 29, 44, 171, 182, 294]. It is worth noting that the introduction of Long Short Term Memory Networks [86] and Recurrent Neural Networks [69, 122] into the processing of contextual information can help object detection. Representative network architectures include ION [7], ACCNN [129], Recurrent CNN [138], and GBDNet [269]. This aspect of research will have great significance for object detection, and definitely become a mainstream direction in the future.

We summarize the object detection algorithms based on DCNNs after the popularity of the deep learning. The traditional object detection algorithms are also mentioned in

this review. By summarizing the existing algorithms, technologies and architectures, it is a review of current development and a prospect of object detection. We believe that object detection will make great progress with the continuous development of basic theory and related hardware equipment in the future.

Acknowledgments This work was supported in part by NSFC under grant No. 61876148, No. 61866022, and No. 61703328. This work was also supported in part by the key project of Trico-Robot plan of NSFC under grant No. 91748208, key project of Shaanxi province No.2018ZDCXL-GY-06-07, the Fundamental Research Funds for the Central Universities No. XJJ2018254, and China Postdoctoral Science Foundation NO. 2018M631164.

References

1. Alexe B, Deselaers T, Ferrari V (2012) Measuring the objectness of image windows. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(11):2189–2202
2. Andreas G, Philip L, Raquel U (2012) Are we ready for autonomous driving? The KITTI vision benchmark suite. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 3354–3361
3. Anwar S, Sung W (2016) Coarse pruning of convolutional neural networks with random masks
4. Arbelaez P, Pont-Tuset J, Barron JT, Marques F, Malik J (2014) Multiscale combinatorial grouping. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 328–335
5. Bae SH (2019) Object detection based on region decomposition and assembly. In: Proceedings of the AAAI conference on artificial intelligence (AAAI)
6. Bartlett PL, Wegkamp MH (2008) Classification with a reject option using a hinge loss. J Mach Learn Res 9(8):1823–1840
7. Bell S, Lawrence Zitnick C, Bala K, Girshick R (2016) Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 2874–2883
8. Bengio Y, Courville A, Vincent P (2013) Representation learning: a review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(8):1798–1828
9. Bertinetto L, Valmadre J, Henriques JF, Vedaldi A, Torr PH (2016) Fully-convolutional siamese networks for object tracking. In: European conference on computer vision (ECCV), pp 850–865
10. Bodla N, Singh B, Chellappa R, Davis LS (2017) Soft-NMS–Improving object detection with one line of code. In: IEEE international conference on computer vision (ICCV), pp 5562–5570
11. Borji A, Cheng MM, Hou Q, Jiang H, Li J (2014) Salient object detection: a survey. Computational Visual Media, pp 1–34
12. Bromley J, Guyon I, LeCun Y, S?ckinger E, Shah R (1994) Signature verification using a siamese time delay neural network. In: Advances in neural information processing systems (NIPS), pp 737–744
13. Caffe2 (2020) A new lightweight, modular, and scalable deep learning framework. https://caffe2.ai/. Software available from caffe2.ai
14. Cai Z, Fan Q, Feris RS, Vasconcelos N (2016) A unified multi-scale deep convolutional neural network for fast object detection. In: European conference on computer vision (ECCV), pp 354–370
15. Cai L, Zhao B, Wang Z, Lin J, Foo CS, Aly MS, Chandrasekhar V (2019) MaxpoolNMS: getting rid of NMS bottlenecks in two-stage object detectors. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 9356–9364
16. Cai Z, Vasconcelos N (2018) Cascade r-cnn: delving into high quality object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 6154–6162
17. Cao G, Xie X, Yang W, Liao Q, Shi G, Wu J (2018) Feature-fused SSD: fast detection for small objects. In: Ninth international conference on graphic and image processing (ICGIP), p 106151
18. Caron M, Bojanowski P, Joulin A, Douze M (2018) Deep clustering for unsupervised learning of visual features. In: European conference on computer vision (ECCV), pp 139–156
19. Carreira J, Sminchisescu C (2011) CPMC: automatic object segmentation using constrained parametric min-cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence (7): 1312–1328
20. Caruana R (1997) Multitask learning. Mach Learn 28(1):41–75
21. Castro FM, Marin-Jimenez MJ, Guil N, Schmid C, Alahari K (2018) End-to-end incremental learning. In: Proceedings of the European conference on computer vision (ECCV), pp 233–248

22. Chai T, Draxler RR (2014) Root mean square error (RMSE) or mean absolute error (MAE)? Arguments against avoiding RMSE in the literature. Geosci Model Dev 7(3):1247–1250
23. Chen X, Xiang S, Liu C-L, Pan C-H (2013) Vehicle detection in satellite images by parallel deep convolutional neural networks. In: Asian conference on pattern recognition (ACPR), pp 181–185
24. Chen C, Seff A, Kornhauser A, Xiao J (2015) Deepdriving: learning affordance for direct perception in autonomous driving. In: Proceedings of the IEEE international conference on computer vision (CVPR), pp 2722–2730
25. Chen G, Choi W, Yu X, Han T, Chandraker M (2017) Learning efficient object detection models with knowledge distillation. In: Advances in neural information processing systems (NIPS), pp 742–751
26. Chen H, Wang Y, Wang G, Qiao Y (2018) LSTD: a low-shot transfer detector for object detection
27. Chen K, et al. (2020) Open MMLab Detection Toolbox (mmdetection). https://github.com/open-mmlab/mmdetection
28. Chen Q, Song Z, Dong J, Huang Z, Hua Y, Yan S (2015) Contextualizing object detection and classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 37(1):13–27
29. Chen X, Gupta A (2017) Spatial memory for context reasoning in object detection. In: IEEE international conference on computer vision (ICCV), pp 4106–4116
30. Chen X, Kundu K, Zhu Y, Berneshawi AG, Ma H, Fidler S, Urtasun R (2015) 3d object proposals for accurate object class detection. In: Advances in neural information processing systems (NIPS), pp 424–432
31. Chen X, Ma H, Wan J, Li B, Xia T (2017) Multi-view 3d object detection network for autonomous driving. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 6526–6534
32. Cheng B, Wei Y, Shi H, Feris R, Xiong J, Huang T (2018) Revisiting rcnn: on awakening the classification power of faster rcnn. In: Proceedings of the European conference on computer vision (ECCV), pp 453–468
33. Cheng G, Zhou P, Han J (2016) Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. IEEE Trans Geosci Remote Sens 54(12):7405–7415
34. Chollet F (2017) Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1800–1807
35. Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, Franke U, Roth S, Schiele B (2016) The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 3213–3223
36. Cortes C, Vapnik V (1995) Support vector machine. Mach Learn 20(3):273–297
37. Dai J, He K, Sun J (2016) Instance-aware semantic segmentation via multi-task network cascades. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 3150–3158
38. Dai J, Li Y, He K, Sun J (2016) R-FCN: object detection via region-based fully convolutional networks. In: Advances in neural information processing systems (NIPS), pp 379–387
39. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR
40. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 886–893
41. De Boer P-T, Kroese DP, Mannor S, Rubinstein RY (2005) A tutorial on the cross-entropy method. Ann Oper Res 134(1):19–67
42. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 248–255
43. Denton E, Zaremba W, Bruna J, LeCun Y, Fergus R (2014) Exploiting linear structure within convolutional networks for efficient evaluation. In: Advances in neural information processing systems (NIPS), pp 1269–1277
44. Divvala SK, Hoiem D, Hays JH, Efros AA, Hebert M (2009) An empirical study of context in object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1271–1278
45. Dollar P, Wojek C, Schiele B, Perona P (2009) Pedestrian detection: a benchmark. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 304–311
46. Dollar P, Wojek C, Schiele B, Perona P (2012) Pedestrian detection: an evaluation of the state of the art. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(4):743–761
47. Duan K, Bai S, Xie L, Qi H, Huang Q, Tian Q (2019) Centernet: keypoint triplets for object detection. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 6569–6578

48. Dundar A, Jin J, Culurciello E (2016) Convolutional clustering for unsupervised learning. In: International conference on learning representations (ICLR
49. Endres I, Hoiem D (2010) Category independent object proposals. In: European conference on computer vision (ECCV), pp 575–588
50. Erhan D, Szegedy C, Toshev A, Anguelov D (2014) Scalable object detection using deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 2147–2154
51. Everingham M, Eslami SA, Van Gool L, Williams CK, Winn J, Zisserman A (2015) The pascal visual object classes challenge: a retrospective. Int J Comput Vis 111(1):98–136
52. Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. Int J Comput Vis 88(2):303–338
53. Facebook Al Research (2020) FAIR's research platform for object detection research (Detectron). https://github.com/facebookresearch/Detectron
54. Fan Q, Brown L, Smith J (2016) A closer look at faster R-CNN for vehicle detection. In: IEEE intelligent vehicles symposium (IV), pp 124–129
55. Fei-Fei L, Fergus R, Perona P (2006) One-shot learning of object categories. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(4):594–611
56. Fei-Fei L, Fergus R, Perona P (2007) Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. Computer Vision Image Understanding 106(1):59–70
57. Felzenszwalb P, McAllester D, Ramanan D (2008) A discriminatively trained, multiscale, deformable part model. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1–8
58. Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D (2010) Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(9):1627–1645
59. Fu C-Y, Liu W, Ranga A, Tyagi A, Berg AC (2017) DSSD: deconvolutional single shot detector. arXiv:170106659
60. Gao M, Yu R, Li A, Morariu VI, Davis LS (2018) Dynamic zoom-in network for fast object detection in large images. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 6926–6935
61. Gentile C, Warmuth MK (1999) Linear hinge loss and average margin. In: Advances in neural information processing systems (NIPS), pp 225–231
62. Ghiasi G, Lin TY, Le QV (2019) Nas-fpn: learning scalable feature pyramid architecture for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 7036–7045
63. Ghodrati A, Diba A, Pedersoli M, Tuytelaars T, Van Gool L (2015) Deepproposal: hunting objects by cascading deep convolutional layers. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 2578–2586
64. Gholami A, Kwon K, Wu B, Tai Z, Yue X, Jin P, Zhao S, Keutzer K (2018) SqueezeNext: hardware-aware neural network design. In: IEEE conference on computer vision and pattern recognition workshops (CVPRW), pp 1638–1647
65. Girshick R (2015) Fast R-CNN. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 1440–1448
66. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 580–587
67. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Advances in neural information processing systems (NIPS), pp 2672–2680
68. Gordon RS, Perez M (2018) Safety for wearable virtual reality devices via object detection and tracking. US Patent Application
69. Graves A, Mohamed A-R, Hinton G (2013) Speech recognition with deep recurrent neural networks. In: IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 6645–6649
70. Griffin G, Holub A, Perona P (2007) Caltech-256 object category dataset. In: Technical Report of California Institute
71. Gu C, Sun C, Ross D, Vondrick C, Pantofaru C, Li Y, Vijayanarasimhan S, Toderici G, Ricco S, Sukthankar R (2018) AVA: a video dataset of spatio-temporally localized atomic visual actions. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 6047–6056

72. Gupta S, Girshick R, Arbelaez P, Malik J (2014) Learning rich features from RGB-D images for object detection and segmentation. In: European conference on computer vision (ECCV), pp 345–360
73. Han S, Mao H, Dally WJ (2016) Deep compression: compressing deep neural networks with pruning trained quantization and huffman coding. In: International conference on learning representations (ICLR)
74. Hao Y, Fu Y, Jiang YG, Tian Q (2019, July) An end-to-end architecture for class-incremental object detection with knowledge distillation. In: IEEE international conference on multimedia and expo (ICME), pp 1–6
75. Hao Z, Liu Y, Qin H, Yan J, Li X, Hu X (2017) Scale-aware face detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 6186–6195
76. Hariharan B, Arbelaez P, Girshick R, Malik J (2017) Object instance segmentation and fine-grained localization using hypercolumns. IEEE Transactions on Pattern Analysis and Machine Intelligence (4): 627–639
77. Harzallah H, Jurie F, Schmid C (2009) Combining efficient object localization and image classification. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 237–244
78. Hastie T, Tibshirani R, Friedman J (2009) Unsupervised learning. In: The elements of statistical learning. Springer, Berlin, pp 485–585
79. He K, Girshick R, Dollar P (2018) Rethinking ImageNet Pre-training. arXiv:181108883
80. He K, Gkioxari G, Dollar P, Girshick R (2017) Mask R-CNN. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 2980–2988
81. He KM, Zhang XY, Ren SQ, Sun J (2014) Spatial pyramid pooling in deep convolutional networks for visual recognition. In: European conference on computer vision (ECCV), pp 346–361
82. He KM, Zhang XY, Ren SQ, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 770–778
83. He S, Lau RW, Liu W, Huang Z, Yang Q (2015) Supercnn: a superpixelwise convolutional neural network for salient object detection. Int J Comput Vis 115(3):330–344
84. He Y, Zhang X, Savvides M, Kitani K (2018) Softer-NMS: rethinking bounding box regression for accurate object detection. arXiv:180908545
85. Hetang C, Qin H, Liu S, Yan J (2017) Impression network for video object detection. arXiv:171205896
86. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
87. Hoi SC, Wu X, Liu H, Wu Y, Wang H, Xue H, Wu Q (2015) Logo-net: large-scale deep logo detection and brand recognition with deep region-based convolutional networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 46(5):2403–2412
88. Howard AG, 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: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)
89. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 7132–7141
90. Hu P, Ramanan D (2017) Finding tiny faces. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 951–959
91. Hu P, Shuai B, Liu J, Wang G (2017) Deep level sets for salient object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 2300–2309
92. Huang G, Liu Z, van der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 2261–2269
93. Huang J, Rathod V, Sun C, Zhu ML, Korattikara A, Fathi A, Fischer I, Wojna Z, Song Y, Guadarrama S, Murphy K (2017) Speed/accuracy trade-offs for modern convolutional object detectors. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 3296–3297
94. Huang W, Qiao Y, Tang X (2014) Robust scene text detection with convolution neural network induced mser trees. In: European conference on computer vision (ECCV), pp 497–511
95. Hwang S, Kim HE (2016) Self-transfer learning for weakly supervised lesion localization. In: International conference on medical image computing and computer-assisted intervention, pp 239–246
96. Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K (2017) Squeezenet: alexnet-level accuracy with 50x fewer parameters and ¡ 0.5 mb model size. In: International conference on learning representations (ICLR)
97. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning (ICML), pp 448–456
98. Itti L, Koch C, Niebur E (1998) A model of saliency-based visual attention for rapid scene analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence (11): 1254–1259

99. Janocha K, Czarnecki WM (2017) On loss functions for deep neural networks in classification. arXiv:170205659
100. Jiang B, Luo R, Mao J, Xiao T, Jiang Y (2018) Acquisition of localization confidence for accurate object detection. In: European conference on computer vision (ECCV), pp 8–14
101. Jiang H, Learned-Miller E (2017) Face detection with the faster R-CNN. In: IEEE international conference on automatic face and gesture recognition, pp 650–657
102. Kang K, Li H, Yan J, Zeng X, Yang B, Xiao T, Zhang C, Wang Z, Wang R, Wang X (2018) T-cnn: tubelets with convolutional neural networks for object detection from videos. IEEE Transactions on Circuits Systems for Video Technology 28(10):2896–2907
103. Kang K, Ouyang W, Li H, Wang X (2016) Object detection from video tubelets with convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 817–825
104. Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Fei-Fei L (2014) Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1725–1732
105. Kavukcuoglu K, Sermanet P, Boureau Y-L, Gregor K, Mathieu M, Cun YL (2010) Learning convolutional feature hierarchies for visual recognition. In: Advances in neural information processing systems (NIPS), pp 1090–1098
106. Kawahara J, Hamarneh G (2016) Multi-resolution-tract CNN with hybrid pretrained and skin-lesion trained layers. In: International workshop on machine learning in medical imaging, pp 164–171
107. Kim S-W, Kook H-K, Sun J-Y, Kang M-C, Ko S-J (2018) Parallel feature pyramid network for object detection. In: European conference on computer vision (ECCV), pp 234–250
108. Kleban J, Xie X, Ma W-Y (2008) Spatial pyramid mining for logo detection in natural scenes. In: IEEE international conference on multimedia and expo (ICME), pp 1077–1080
109. Koch G, Zemel R, Salakhutdinov R (2015) Siamese neural networks for one-shot image recognition. In: International conference on machine learning (ICML
110. Kong B, Zhan Y, Shin M, Denny T, Zhang S (2016) Recognizing end-diastole and end-systole frames via deep temporal regression network. In: International conference on medical image computing and computer-assisted intervention, pp 264–272
111. Kong T, Sun F, Yao A, Liu H, Lu M, Chen Y (2017) Ron: reverse connection with objectness prior networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 5244–5252
112. Kong T, Yao A, Chen Y, Sun F (2016) Hypernet: towards accurate region proposal generation and joint object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 845–853
113. Krasin I, Duerig T, Alldrin N, Veit A, Abu-El-Haija S, Belongie S, Cai D, Feng Z, Ferrari V, Gomes V (2016) Openimages: a public dataset for large-scale multi-label and multi-class image classification. Dataset available from https://githubcom/openimages 2(6):7
114. Krhenbuhl P, Koltun V (2014) Geodesic object proposals. In: European conference on computer vision (ECCV), pp 725–739
115. Krizhevsky A, Hinton G (2009) Learning multiple layers of features from tiny images. In: Technical Report of University of Toronto
116. Krizhevsky A, Sutskever I, Hinton GE (2017) Imagenet classification with deep convolutional neural networks. Commun ACM 60(6):84–90
117. Kuo W, Hariharan B, Malik J (2015) Deepbox: learning objectness with convolutional networks. In: Proceedings of the IEEE international conference on computer vision (CVPR), pp 2479–2487
118. Lai K, Bo L, Ren X, Fox D (2011) A large-scale hierarchical multi-view rgb-d object dataset. In: IEEE international conference on robotics and automation (ICRA), pp 1817–1824
119. LaLonde R, Bagci U (2018) Capsules for object segmentation. arXiv:1804.04241
120. Law H, Deng J (2018) Cornernet: detecting objects as paired keypoints. In: Proceedings of the European conference on computer vision (ECCV), pp 734–750
121. Lazebnik S, Schmid C, Ponce J (2006) Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 2169–2178
122. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
123. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. In: Proceedings of the IEEE, pp 2278–2324
124. Li G, Yu Y (2015) Visual saliency based on multiscale deep features. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 5455–5463

125. Li H, Kadav A, Durdanovic I, Samet H, Graf HP (2016) Pruning filters for efficient convnets. arXiv:1608.08710
126. Li H, Lin Z, Shen X, Brandt J, Hua G (2015) A convolutional neural network cascade for face detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 5325–5334
127. Li J, Liang X, Shen S, Xu T, Feng J, Yan S (2017) Scale-aware fast R-CNN for pedestrian detection. IEEE Transactions on Multimedia 20(4):985–996
128. Li J, Liang X, Wei Y, Xu T, Feng J, Yan S (2017) Perceptual generative adversarial networks for small object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1222–1230
129. Li J, Wei Y, Liang X, Dong J, Xu T, Feng J, Yan S (2017) Attentive contexts for object detection. IEEE Transactions on Multimedia 19(5):944–954
130. Li K, Cheng G, Bu S, You X (2017) Rotation-insensitive and context-augmented object detection in remote sensing images. IEEE Trans Geosci Remote Sens 56(4):2337–2348
131. Li L, Xu M, Wang X, Jiang L, Liu H (2019) Attention based glaucoma detection: a large-scale database and CNN model. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 10571–10580
132. Li S, Yang L, Huang J, Hua XS, Zhang L (2019) Dynamic anchor feature selection for single-shot object detection. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 6609–6618
133. Li Y, Chen Y, Wang N, Zhang Z (2019) Scale-aware trident networks for object detection. In: Proceedings of the IEEE international conference on computer vision (ICCV)
134. Li Y, Wang D, Hu H, Lin Y, Zhuang Y (2017) Zero-shot recognition using dual visual-semantic mapping paths. In: Proceedings of the IEEE international conference on computer vision (CVPR), pp 5207–5215
135. Li Z, Peng C, Yu G, Zhang X, Deng Y, Sun J (2018) DetNet: design backbone for object detection. In: European conference on computer vision (ECCV), pp 334–350
136. Li Z, Peng C, Yu G, Zhang X, Deng Y, Sun J (2018) Light-head R-CNN: in defense of two-stage object detector. In: Proceedings of the IEEE international conference on computer vision (CVPR)
137. Li Z, Zhou F (2017) FSSD: feature fusion single shot multibox detector. arXiv:171200960
138. Liang M, Hu X (2015) Recurrent convolutional neural network for object recognition. In: Proceedings of the IEEE international conference on computer vision (CVPR), pp 3367–3375
139. Liangkui L, Shaoyou W, Zhongxing T (2018) Using deep learning to detect small targets in infrared oversampling images. J Syst Eng Electron 29(5):947–952
140. Lienhart R, Maydt J (2002) An extended set of haar-like features for rapid object detection. In: Proceedings of the international conference on image processing (ICIP), pp 1–1
141. Lin M, Chen Q, Yan S (2014) Network in network. In: International conference on learning representations (ICLR)
142. Lin T, Goyal P, Girshick R, He K, Dollr P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 2999–3007
143. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollar P, Zitnick CL (2014) Microsoft coco: common objects in context. In: European conference on computer vision (ECCV), pp 740–755
144. Lin TY, Dollar P, Girshick R, He KM, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 936–944
145. Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, Van Der Laak JA, Van Ginneken B, Sanchez CI (2017) A survey on deep learning in medical image analysis. Med Image Anal 42:60–88
146. Liu C, Zoph B, Neumann M, Shlens J, Hua W, Li L-J, Fei-Fei L, Yuille A, Huang J, Murphy K (2018) Progressive neural architecture search. In: European conference on computer vision (ECCV), pp 19–34
147. Liu L, Ouyang W, Wang X, Fieguth P, Chen J, Liu X, Pietik?inen M (2018) Deep learning for generic object detection: a survey. International Journal of Computer Vision
148. Liu S, Huang D, Wang Y (2018) Receptive field block net for accurate and fast object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)
149. Liu T, Yuan Z, Sun J, Wang J, Zheng N, Tang X, Shum HY (2010) Learning to detect a salient object. IEEE Transactions on Pattern analysis and Machine Intelligence 33(2):353–367
150. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2016) SSD: single shot multibox detector. In: European conference on computer vision (ECCV), pp 21–37
151. Liu Y, Wang R, Shan S, Chen X (2018) Structure inference net: object detection using scene-level context and instance-level relationships. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 6985–6994

152. 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 (CVPR), pp 3431–3440
153. Long Y, Gong Y, Xiao Z, Liu Q (2017) Accurate object localization in remote sensing images based on convolutional neural networks. IEEE Trans Geosci Remote Sens 55(5):2486–2498
154. Lotter W, Kreiman G, Cox D (2017) Deep predictive coding networks for video prediction and unsupervised learning. International conference on learning representations (ICLR)
155. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
156. Luo JH, Wu J (2017) An entropy-based pruning method for cnn compression. arXiv:1706.05791
157. Ma N, Zhang X, Zheng H-T, Sun J (2018) ShuffleNet v2: practical guidelines for efficient cnn architecture design. In: European conference on computer vision (ECCV), pp 116–131
158. Mao H, Yao S, Tang T, Li B, Yao J, Wang Y (2018) Towards real-time object detection on embedded systems. IEEE Transactions on Emerging Topics in Computing 6(3):417–431
159. Mao J, Xiao T, Jiang Y, Cao Z (2017) What can help pedestrian detection? In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 3127–3136
160. Mate K, Zbigniew W, Jakub M, Jacek N, Kyunghyun C (2019) Augmentation for small object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR
161. Miller GA, Beckwith R, Fellbaum C, Gross D, Miller KJ (1990) Introduction to WordNet: an on-line lexical database. Int J Lexicogr 3(4):235–244
162. Moore R, DeNero J (2013) L1 and L2 regularization for multiclass hinge loss models. Symposium on Machine Learning in Speech and Language Processing
163. Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. In: Proceedings of the international conference on machine learning (ICML), pp 807–814
164. Najafi H, Genc Y (2010) Fast object detection for augmented reality systems. US Patent Application
165. Najibi M, Samangouei P, Chellappa R, Davis LS (2017) Ssh: single stage headless face detector. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 4875–4884
166. Neubeck A, Van Gool L (2006) Efficient non-maximum suppression. In: International conference on pattern recognition (ICPR), pp 850–855
167. Ni K, Pearce R, Boakye K, Van Essen B, Borth D, Chen B, Wang E (2015) Large-scale deep learning on the YFCC100M dataset. arXiv:150203409
168. Oquab M, Bottou L, Laptev I, Sivic J (2014) Learning and transferring mid-level image representations using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1717–1724
169. Ouyang W, Wang K, Zhu X, Wang X (2017) Learning chained deep features and classifiers for cascade in object detection. In: Proceedings of the IEEE international conference on computer vision (ICCV)
170. Ouyang W, Wang X (2013) Joint deep learning for pedestrian detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 2056–2063
171. Ouyang W, Wang X, Zeng X, Qiu S, Luo P, Tian Y, Li H, Yang S, Wang Z, Loy C-C (2015) Deepid-net: deformable deep convolutional neural networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 2403–2412
172. Pang J, Chen K, Shi J, Feng H, Ouyang W, Lin D (2019) Libra r-cnn: towards balanced learning for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 821–830
173. Pang J, Li C, Shi J, Xu Z, Feng H (2019) R^2-CNN: fast tiny object detection in large-scale remote sensing images. IEEE Transactions on Geoscience and Remote Sensing
174. Peng C, Xiao T, Li Z, Jiang Y, Zhang X, Jia K, Yu G, Sun J (2018) MegDet: a large mini-batch object detector. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 6181–6189
175. Peng J, Sun M, Zhang Z, Tan T, Yan J (2019) POD: practical object detection with scale-sensitive network. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 9607–9616
176. Peng J, Sun M, ZHANG Z X, Tan T, Yan J (2019) Efficient neural architecture transformation search in channel-level for object detection. In: Advances in neural information processing systems (NeurIPS), pp 14290–14299
177. Pentina A, Sharmanska V, Lampert CH (2015) Curriculum learning of multiple tasks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 5492–5500
178. Pinheiro PO, Collobert R, Dollar P (2015) Learning to segment object candidates. In: Advances in neural information processing systems (NIPS), pp 1990–1998

179. PyTorch (2020) Tensors and dynamic neural networks in python with strong GPU acceleration https://pytorch.org/. Software available from pytorch.org
180. Qiang Z, Mei-Chen Y, Kwang-Ting C, Shai A (2006) Fast human detection using a cascade of histograms of oriented gradients. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1491–1498
181. Qiu J, Wang J, Yao S, Guo K, Li B, Zhou E, Yu J, Tang T, Xu N, Song S (2016) Going deeper with embedded fpga platform for convolutional neural network. In: Proceedings of the ACM/SIGDA international symposium on field-programmable gate arrays, pp 26–35
182. Rabinovich A, Vedaldi A, Galleguillos C, Wiewiora E, Belongie S (2007) Objects in context. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 1–8
183. Radford A, Metz L, Chintala S (2016) Unsupervised representation learning with deep convolutional generative adversarial networks. In: International conference on learning representations (ICLR)
184. Rahman S, Khan S, Porikli F (2018) Zero-shot object detection: learning to simultaneously recognize and localize novel concepts. In: European conference on computer vision (ECCV
185. Rastegari M, Ordonez V, Redmon J, Farhadi A (2016) Xnor-net: imagenet classification using binary convolutional neural networks. In: European conference on computer vision (ECCV), pp 525–542
186. Redmon J (2013) Darknet: open source neural networks in c. http://pjreddie.com/darknet
187. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 779–788
188. Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 6517–6525
189. Redmon J, Farhadi A (2018) YOLOv3: an incremental improvement. arXiv:180402767
190. 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
191. Ren SQ, He KM, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in neural information processing systems (NIPS), pp 91–99
192. Ren X, Ramanan D (2013) Histograms of sparse codes for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 3246–3253
193. Ren Y, Zhu C, Xiao S (2018) Small object detection in optical remote sensing images via modified faster R-CNN. Appl Sci 8(5):813
194. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323(6088):533–536
195. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252
196. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C (2018) Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 4510–4520
197. Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, LeCun Y (2014) Overfeat: integrated recognition localization and detection using convolutional networks. In: The international conference on learning representations (ICLR)
198. Shelhamer E, Rakelly K, Hoffman J, Darrell T (2016) Clockwork convnets for video semantic segmentation. In: European conference on computer vision (ECCV), pp 852–868
199. Shen Z, Liu Z, Li J, Jiang Y-G, Chen Y, Xue X (2017) DSOD: learning deeply supervised object detectors from scratch. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 1937–1945
200. Shigeto Y, Suzuki I, Hara K, Shimbo M, Matsumoto Y (2015) Ridge regression, hubness, and zero-shot learning. In: Joint European conference on machine learning and knowledge discovery in databases (ECML PKDD), pp 135–151
201. Shmelkov K, Schmid C, Alahari K (2017) Incremental learning of object detectors without catastrophic forgetting. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 3420–3429
202. Shrivastava A, Gupta A, Girshick R (2016) Training region-based object detectors with online hard example mining. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 761–769
203. Shrivastava A, Sukthankar R, Malik J, Gupta A (2017) Beyond skip connections: top-down modulation for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)
204. Singh B, Davis LS (2018) An analysis of scale invariance in object detection SNIP. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 3578–3587

205. Simon M, Milz S, Amende K, Gross H-M (2018) Complex-YOLO: an euler-region-proposal for real-time 3d object detection on point clouds. In: European Conference on Computer Vision Workshops
206. Simon M, Rodner E, Denzler J (2016) Imagenet pre-trained models with batch normalization. arXiv:161201452
207. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: International conference on learning representations (ICLR)
208. Sivic J, Zisserman A (2003) Video Google: a text retrieval approach to object matching in videos. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 1470–1477
209. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 15(1):1929–1958
210. Sturm J, Engelhard N, Endres F, Burgard W, Cremers D (2012) A benchmark for the evaluation of RGB-D SLAM systems. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 573–580
211. Sun B, Saenko K (2014) From virtual to reality: fast adaptation of virtual object detectors to real domains. In: British machine vision conference (BMVC
212. Szegedy C, Ioffe S, Vanhoucke V, Alemi A (2017) Inception-v4 inception-resnet and the impact of residual connections on learning. In: AAAI conference on artificial intelligence
213. Szegedy C, Liu W, Jia YQ, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1–9
214. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 2818–2826
215. Taigman Y, Yang M, Ranzato MA, Wolf L (2014) Deepface: closing the gap to human-level performance in face verification. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1701–1708
216. Tan M, Chen B, Pang R, Vasudevan V, Le QV (2018) MnasNet: platform-aware neural architecture search for mobile. arXiv:180711626
217. Tanner G (2020) Object detection API System. https://github.com/tensorflow/models
218. Teichmann M, Weber M, Zoellner M, Cipolla R, Urtasun R (2018) Multinet: real-time joint semantic reasoning for autonomous driving. In: IEEE intelligent vehicles symposium (IV), pp 1013–1020
219. TensorFlow (2020) Large-Scale machine learning on heterogeneous distributed systems. https://www.tensorflow.org/. Software available from tensorflow.org
220. Tian Y, Luo P, Wang X, Tang X (2015) Deep learning strong parts for pedestrian detection. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 1904–1912
221. Tian Z, Shen C, Chen H, He T (2019) FCOS: fully convolutional one-stage object detection. In: Proceedings of the IEEE international conference on computer vision (ICCV
222. Torralba A, Fergus R, Freeman WT (2008) 80 million tiny images: a large data set for nonparametric object and scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(11):1958–1970
223. Tychsen-Smith L, Petersson L (2017) Denet: scalable real-time object detection with directed sparse sampling. Objects in context. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 428–436
224. Uijlings JRR, van de Sande KEA, Gevers T, Smeulders AWM (2013) Selective search for object recognition. International Journal of Computer Vision (IJCV) 104(2):154–171
225. van de Sande KEA, Uijlings JRR, Gevers T, Smeulders AWM (2011) Segmentation as selective search for object recognition. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 1879–1886
226. Van Etten A (2018) You only look twice: rapid multi-scale object detection in satellite imagery. arXiv:1805.09512
227. Van Horn G, Mac Aodha O, Song Y, Cui Y, Sun C, Shepard A, Adam H, Perona P, Belongie S (2018) The inaturalist species classification and detection dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 8769–8778
228. Vedaldi A, Gulshan V, Varma M, Zisserman A (2009) Multiple kernels for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 606–613
229. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1–1
230. Viola P, Jones MJ (2004) Robust real-time face detection. International journal of computer vision 57(2):137–154

231. Wagstaff K, Cardie C, Rogers S, Schr?dl S (2001) Constrained k-means clustering with background knowledge. In: International conference on machine learning (ICML), pp 577–584
232. Wang C, Bai X, Wang S, Zhou J, Ren P (2018) Multiscale visual attention networks for object detection in VHR remote sensing images. IEEE Geosci Remote Sens Lett 16(2):310–314
233. Wang H, Wang Q, Gao M, Li P, Zuo W (2018) Multi-scale location-aware kernel representation for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1248–1257
234. Wang J, Zheng T, Lei P, Bai X (2019) A hierarchical convolution neural network (CNN)-based ship target detection method in spaceborne SAR imagery. Remote Sens 11(6):620
235. Wang L, Lu H, Ruan X, Yang MH (2015) Deep networks for saliency detection via local estimation and global search. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 3183–3192
236. Wang L, Wang L, Lu H, Zhang P, Ruan X (2016) Saliency detection with recurrent fully convolutional networks. In: European conference on computer vision (ECCV), pp 825–841
237. Wang RJ, Li X, Ao S, Ling CX (2018) Pelee: a real-time object detection system on mobile devices. In: Advances in neural information processing systems (NIPS)
238. Wang S, Zhou Y, Yan J, Deng Z (2018) Fully motion-aware network for video object detection. In: European conference on computer vision (ECCV), pp 542–557
239. Wang W, Lai Q, Fu H, Shen J, Ling H (2019) Salient object detection in the deep learning era: an in-depth survey. arXiv:1904.09146
240. Wang WH, Yang J, Xiao JW, Li S, Zhou DX (2015) Face recognition based on deep learning. In: International conference on human-centered computing (HCC), pp 812–820
241. Wang X, Han T, Yan S (2009) An HOG-LBP human detector with partial occlusion handling. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 32–39
242. Wang X, Xiao T, Jiang Y, Shao S, Sun J, Shen C (2018) Repulsion loss: detecting pedestrians in a crowd. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 7774–7783
243. Wang XL, Shrivastava A, Gupta A (2017) A-Fast-RCNN: hard positive generation via adversary for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 3039–3048
244. Wei Y, You X, Li H (2016) Multiscale patch-based contrast measure for small infrared target detection. Pattern Recogn 58:216–226
245. Weimer D, Scholz-Reiter B, Shpitalni M (2016) Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. CIRP Annals-Manufacturing Technology 65(1):417–420
246. Weston J, Watkins C (1999) Support vector machines for multi-class pattern recognition. In: European symposium on artificial neural networks (ESANN), pp 219–224
247. Willmott CJ, Matsuura K (2005) Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Climate Research 30(1):79–82
248. Wu J, Leng C, Wang Y, Hu Q, Cheng J (2016) Quantized convolutional neural networks for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 4820–4828
249. Wu X, Wu Y, Zhao Y (2016) Binarized neural networks on the imagenet classification task. arXiv:1604.03058
250. Xiangyu Z, Xinyu Z, Mengxiao L, Jian S (2017) ShuffleNet: an extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 6848–6856
251. Xiao J, Ehinger KA, Hays J, Torralba A, Oliva A (2016) Sun database: exploring a large collection of scene categories. Int J Comput Vis 119(1):3–22
252. Xie S, Girshick R, Dollar P, Tu Z, He K (2017) Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 5987–5995
253. Xu M, Cui L, Lv P, Jiang X, Niu J, Zhou B, Wang M (2018) Mdssd: Multi-scale deconvolutional single shot detector for small objects. arXiv:1805.07009
254. Xue J, Li JY, Gong YF (2013) Restructuring of deep neural network acoustic models with singular value decomposition. In: Annual conference of the international speech communication association, pp 2364–2368
255. Yanai K, Kawano Y (2015) Food image recognition using deep convolutional network with pre-training and fine-tuning. In: IEEE international conference on multimedia and expo workshops (ICMEW), pp 1–6

256. Yang B, Yan J, Lei Z, Li SZ (2014) Aggregate channel features for multi-view face detection. In: IEEE international joint conference on biometrics (IJCB), pp 1–8
257. Yang S, Luo P, Loy CC, Tang X (2017) Faceness-net: face detection through deep facial part responses. IEEE Transactions on Pattern Analysis and Machine Intelligence 40(8):1845–1859
258. Yang TJ, Chen YH, Sze V (2017) Designing energy-efficient convolutional neural networks using energy-aware pruning. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 5687–5695
259. Yang Z, Liu S, Hu H, Wang L, Lin S (2019) Reppoints: point set representation for object detection. In: Proceedings of the IEEE international conference on computer vision (ICCV
260. Yildirim G, Susstrunk S (2014) FASA: fast, accurate, and size-aware salient object detection. In: Asian conference on computer vision (ACCV), pp 514–528
261. Yu F, Koltun V (2016) Multi-scale context aggregation by dilated convolutions. In: International conference on learning representations (ICLR)
262. Yu F, Koltun V, Funkhouser TA (2017) Dilated residual networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 636–644
263. Yu Y, Zhang J, Huang Y, Zheng S, Ren W, Wang C, Huang K, Tan T (2010) Object detection by context and boosted HOG-LBP. In: European conference on computer vision workshop on PASCAL VOC
264. Yu Z, Wong HS (2007) A rule based technique for extraction of visual attention regions based on real-time clustering. IEEE Transactions on Multimedia 9(4):766–784
265. Zagoruyko S, Lerer A, Lin TY, Pinheiro PO, Gross S, Chintala S, Dollar P (2016) A multipath network for object detection. arXiv:1604.02135
266. Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: European conference on computer vision (ECCV), pp 818–833
267. Zeiler MD, Krishnan D, Taylor GW, Fergus R (2010) Deconvolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 2528–2535
268. Zeiler MD, Taylor GW, Fergus R (2011) Adaptive deconvolutional networks for mid and high level feature learning. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 2018–2025
269. Zeng X, Ouyang W, Yang B, Yan J, Wang X (2016) Gated bi-directional cnn for object detection. In: European conference on computer vision (ECCV), pp 354–369
270. Zhang H, Chang H, Ma B, Shan S, Chen X (2019) Cascade RetinaNet: maintaining consistency for single-stage object detection. In: The British machine vision conference (BMVC)
271. Zhang J, Sclaroff S, Lin Z, Shen X, Price B, Mech R (2016) Unconstrained salient object detection via proposal subset optimization. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 5733–5742
272. Zhang J, Zhang T, Dai Y, Harandi M, Hartley R (2018) Deep unsupervised saliency detection: a multiple noisy labeling perspective. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 9029–9038
273. Zhang K, Zhang Z, Li Z, Qiao Y (2016) Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters 23(10):1499–1503
274. Zhang L, Lin L, Liang X, He K (2016) Is faster r-cnn doing well for pedestrian detection? In: European conference on computer vision (ECCV), pp 443–457
275. Zhang S, Benenson R, Schiele B (2017) Citypersons: a diverse dataset for pedestrian detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 4457–4465
276. Zhang S, Wen L, Bian X, Lei Z, Li SZ (2018) Occlusion-aware R-CNN: detecting pedestrians in a crowd. In: European conference on computer vision (ECCV), pp 637–653
277. Zhang S, Wen L, Bian X, Lei Z, Li SZ (2018) Single-shot refinement neural network for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 4203–4212
278. Zhang S, Zhu X, Lei Z, Shi H, Wang X, Li SZ (2017) Faceboxes: a CPU real-time face detector with high accuracy. In: IEEE international joint conference on biometrics (IJCB), pp 1–9
279. Zhang S, Zhu X, Lei Z, Shi H, Wang X, Li SZ (2017) S3fd: single shot scale-invariant face detector. In: Proceedings of the IEEE international conference on computer vision (CVPR), pp 192–201
280. Zhang X, Wan F, Liu C, Ji R, Ye Q (2019) Freeanchor: learning to match anchors for visual object detection. In: Advances in neural information processing systems (NeurIPS), pp 147–155
281. Zhang Z, Qiao S, Xie C, Shen W, Wang B, Yuille AL (2018) Single-shot object detection with enriched semantics. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 5813–5821

282. Zhang Z, Zhang C, Shen W, Yao C, Liu W, Bai X (2016) Multi-oriented text detection with fully convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 4159–4167
283. Zhao Q, Sheng T, Wang Y, Tang Z, Chen Y, Cai L, Ling H (2019) M2det: a single-shot object detector based on multi-level feature pyramid network. In: Proceedings of the AAAI conference on artificial intelligence (AAAI), pp 9259–9266
284. Zhao R, Ouyang W, Li H, Wang X (2015) Saliency detection by multi-context deep learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1265–1274
285. Zhao Z-Q, Zheng P, Xu S-T, Wu X (2018) Object detection with deep learning: a review. IEEE Transactions on Neural Networks and Learning Systems, pp 1–21
286. Zhou B, Lapedriza A, Khosla A, Oliva A, Torralba A (2018) Places: a 10 million image database for scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 40(6):1452–1464
287. Zhou P, Ni B, Geng C, Hu J, Xu Y (2018) Scale-transferrable object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 528–537
288. Zhou P, Ni BB, Geng C, Hu JG, Xu Y (2018) Scale-transferrable object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 528–537
289. Zhou X, Zhuo J, Krahenbuhl P (2019) Bottom-up object detection by grouping extreme and center points. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 850–859
290. Zhu R, Zhang S, Wang X, Wen L, Shi H, Bo L, Mei T (2019) ScratchDet: training single-shot object detectors from scratch. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 2268–2277
291. Zhu X, Dai J, Yuan L, Wei Y (2018) Towards high performance video object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 7210–7218
292. Zhu X, Wang Y, Dai J, Yuan L, Wei Y (2017) Flow-guided feature aggregation for video object detection. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 408–417
293. Zhu X, Xiong Y, Dai J, Yuan L, Wei Y (2017) Deep feature flow for video recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 4141–4150
294. Zhu Y, Urtasun R, Salakhutdinov R, Fidler S (2015) SegDeepM: exploiting segmentation and context in deep neural networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 4703–4711
295. Zhu Y, Zhao C, Wang J, Zhao X, Wu Y, Lu H (2017) Couplenet: coupling global structure with local parts for object detection. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 4126–4134
296. Zitnick CL, Dollar P (2014) Edge boxes: locating object proposals from edges. In: European conference on computer vision (ECCV), pp 391–405
297. Zoph B, Vasudevan V, Shlens J, Le QV (2018) Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 8697–8710

Youzi Xiao received his Bachelor degree from Xi'an Shiyou University in 2015. He is currently a master student in the School of Software Engineering in Xi'an Jiaotong University. His research interests include computer vision, pattern recognition.

Zhiqiang Tian is an associate professor at Xi'an Jiaotong University. He received the B.S. degree in Automation Control from the Northeastern University in 2004, the M.S. and Ph.D. degrees in Control Science and Engineering from Xi'an Jiaotong University in 2007 and 2013, respectively. He was a postdoctoral fellow in the Department of Radiology and Imaging Sciences of Emory University from 2014 to 2017. His research interests are image/video processing, computer vision, multimedia, and medical image analysis.

Jiachen Yu received his Bachelor degree from Xi'an Jiaotong University in 2017. He is currently a master student in the School of Software Engineering in Xi'an Jiaotong University. His research interests include computer vision, machine learning.

Yinshu Zhang received her Bachelor degree from Northeast Forestry University in 2017. She is currently a master student in the School of Software Engineering in Xi'an Jiaotong University. Her research interests include generative adversarial networks, computer vision.

Shuai Liu received the B.S.degree in Observation & Control Engineering and Instrumentation, and Ph.D. degree in Circuit and System from Xidian University, Xi'an, China, in 2009 and 2017 respectively. She is currently a faculty member at the School of Software Engineering, Xi'an Jiaotong University, Xi'an, China. Her major research interests include statistical machine learning and bayesian deep network.

Shaoyi Du received double Bachelor degrees in computational mathematics and in computer science in 2002 and received his MS degree in applied mathematics in 2005 and PhD degree in pattern recognition and intelligence system at Xi'an Jiaotong University. He was a postdoc research fellow in computer science in Xi'an Jiaotong University from 2009 to 2011, and a visiting scholar at the University of North Carolina at Chapel Hill from 2013 to 2014. He is currently a professor in Institute of Artificial Intelligence and Robotics at Xi'an Jiaotong University. His research interests include image registration, intelligence vehicle, medical image analysis, face image analysis.

Xuguang Lan received the Ph.D degree in control science and engineering at Xi'an Jiaotong University in 2005. Currently he is a professor of the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. His research interests include image/video coding, processing and communication, P2P technology, and VLSI design.