Multimedia Tools and Applications (2019) 78:27809–27847
https://doi.org/10.1007/s11042-019-07898-2

Recent progresses on object detection: a brief review

Hao Zhang[1] · **Xianggong Hong**[1]

Received: 4 October 2018 / Revised: 19 April 2019 / Accepted: 16 June 2019 /
Published online: 26 June 2019

Abstract
Object detection, aiming at locating objects from a large number of specific categories in natural images, is a fundamental but challenging task in the field of computer vision. Recent years have seen significant progress of object detection using deep CNN mainly due to its robust feature representation ability. The goal of this paper is to provide a simple but comprehensive survey of the recent improvements in object detection in the era of deep learning. More than 100 key contributions are investigated mainly from five directions: architecture diagram, contextual reasoning, multi-layer exploiting, training strategy, and others which includes some other progress like real-time object detectors and works borrowing the idea from RNN and GAN. We discuss comprehensive but straightforward experimental comparisons under widely used benchmarks and metrics. This review finishes by providing promising trends for future research.

Keywords Object detection · Deep convolutional neural networks (CNNs) · Recent progress · Computer vision

1 Introduction

Object detection is a fundamental and challenging task in computer vision. It can be treated as a combination of classification and localization, but multiple objects with different scales should be detected and classified at the same time within one image. Object detection has received wide attention and has been applied in many other fields like autonomous driving [15], surveillance [4], *etc*.

Early object detection approaches adopted the sliding-window paradigm. Hand-crafted features like HOG [12] and SIFT [59] are applied and classifiers detect objects on dense image grids. Based on multi-scale, deformable models, DPM [23] and its descendants have been the leading methods on PASCAL VOC [19] for many years.

✉ Xianggong Hong
fengshenglanshan@163.com

Hao Zhang
haozhang@email.ncu.edu.cn

[1] School of Information Engineering, Nanchang University, Nanchang, Jiangxi, 330031, China

With the availability of large-scale training data like ImageNet [14] and the advance of high-performance GPUs, it's easy to train large models especially deep convolutional neural networks such as AlexNet [45]. Many CNN-based methods have been proposed to boost the performance of object detection for the powerful feature representation ability of CNN. The history of CNNs for vision recognition can date back to LeNet [46] in the 1980s, which is applied to document recognition. Due to the computation constraint at that time and the appearance of the other simple but efficient methods like support vector machine (SVM), deep learning has not been applied broadly yet. Until a milestone method, AlexNet, won the first place in ILSVRC 2012, deep learning represented by CNN returns to the public view and object detection steps into the era of deep learning.

According to [28], object detection can be roughly sorted into three kinds: object detection (OD), salient object detection (SOD) [42] and category-specific object detection (COD). We focus on category-specific object detection, and we use object detection instead of COD through this paper. Object detectors based on CNN can be categorized into two types: *two-stage* and *one-stage*. In the two-stage method, a set of region proposals are firstly generated by Selective Search [85] or EdgeBoxes [107] or region proposal network (RPN) [71], then the certain object locations and the corresponding category labels are determined by using convolutional networks (the below part of Fig. 1). On the contrary, the one-stage approaches use a single feed-forward convolutional network [68] or a reduced convolutional network with extra multi-scale layers [56] to directly predict object classes and object bounding boxes (the above part of Fig. 1).

The difference between one-stage method and two-stage method is whether there is a region proposal stage before the detection part or not, but we summarize the two methods

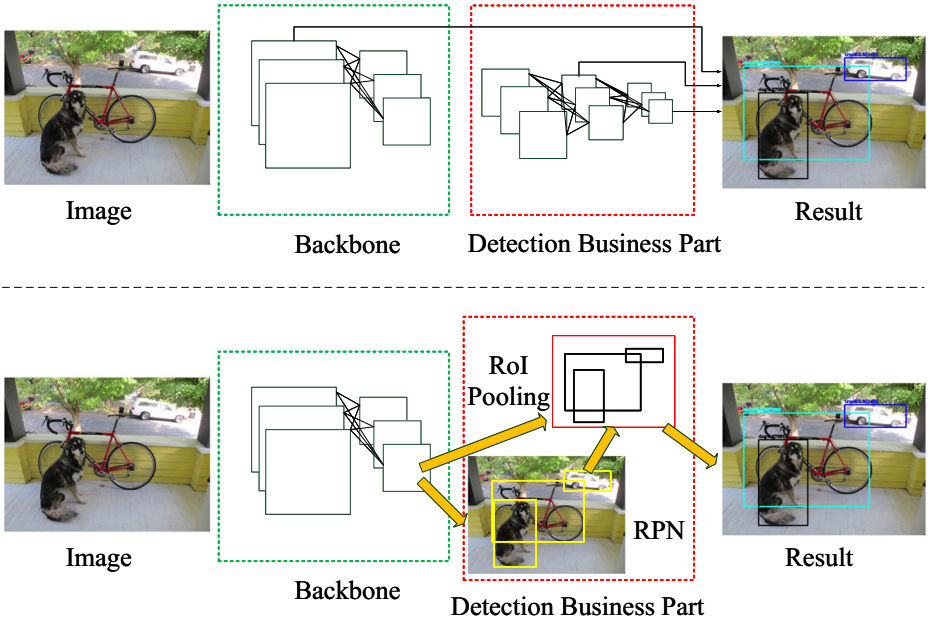


Fig. 1 Illustration of *backbone* and *detection business part*. The diagram above is a concise version of SSD [56] and the below is from Faster R-CNN [71], which are the representative models of *one-stage* and *two-stage* respectively. The backbone networks are usually borrowed from classification task, and a reduced VGG-16 [79] is employed in both the two networks

regardless of the region proposal, and we divide the whole detection process into two parts: *backbone* and *detection business part*, see Fig. 1. The backbone networks are usually designed and trained for ImageNet [14] classification task. The robust feature extraction ability and transfer learning help the later detection business part detect and classify the objects more accurate. The detection business part is integrated into the backbone after the feature map generated in the previous layer.

As the borrowed backbone network is a fundamental part of an object detector, its performance on classification will affect the accuracy on object detection. Since AlexNet [45] became the winning of ImageNet [14] in 2012, the last few years had witnessed significant progress in the accuracy improvement on this authoritative dataset. Intuitively, these architectures focus on the accuracy improvement while they need to solve the problems arising with the increasing of the depth of network (*e.g.*, a 152-layer ResNet [30] surpasses a 16-layer VGG-Net [79] with three times depth but lower complexity). On the other hand, some methods [8, 35, 40, 100] pay close attention to the model size and running speed for applying their applications on devices with memory and computing speed constraint (*e.g.*, SqueezeNet [40] achieves comparable accuracy compared to AlexNet [45] with 50× fewer parameters and less than 0.5MB model size with model compression). More details about these architectures that employed in object detection as backbone will be discussed in Section 2.1.

Object detection business part, as mentioned above, is the vital part of object detection pipeline which turns the feature maps output from backbone to class labels and bounding boxes. Similar to the backbone networks, *two-stage* approaches focus on improving the accuracy on standard evaluation datasets like PASCAL VOC [18, 19] and MS COCO [52]. Differently, *one-stage* methods perform better in speed perspective, such as YOLO [68] can process an image at 45 frames per second (FPS) with the accuracy outperforms R-CNN [24] and its second version YOLO9000 [69] outperforms Faster R-CNN [71] with ResNet while still running significantly fast. Some of the two mainstream methods [24, 56, 68, 73] will be introduced in Section 2.2.

The main purpose of this paper is about the recent progresses of object detection in deep learning era, mainly about the *detection business part*. As shown in Fig. 2, we firstly introduce some representative models of detection business part in Section 2.2. Then, in Section 3, we investigate recent progresses of the detection business part mainly based on R-CNN [24], YOLO [68], and SSD [56] from five perspectives: *architecture diagram*, *contextual reasoning*, *multi-layer exploiting*, *training strategy* and *others*. Methods focusing on architecture diagram try to design new network by replacing their *backbone* or changing the components of the original detectors like using *position-sensitive RoI pooling* or using *deformable convolution* instead of standard convolution or adopting *multi-stage or cascade* structure. Approaches *concatenating multi-scale features*, *using top-down pathway and conduct skip connections* and combining both will be classified into multi-layer exploiting part.

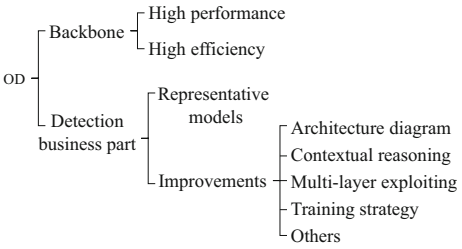


Fig. 2 Architecture of the main part of this review

Contextual information plays an important role in object detection especially for object detection of *small objects*. Some advanced techniques like *concatenating features*, *using a semantic segmentation subnet* utilized for introducing context into detection will be discussed in this part. Different training problems like *class imbalance*, *non-maximum suppression*, *large mini-batch* and *etc.* will be solved with different training strategies. Additionally, new ideas borrowed from the other fields like *Recurrent Neural Network (RNN)*, *generative adversarial networks (GAN)*, *weakly supervised learning*, and *new backbone* for object detection and *real-time object detectors* will be introduced at last.

Benchmarks and metrics for object detection will be talked in Section 4 and comprehensive but straightforward comparisons of object detectors in this paper will be given in Section 5. Finally, we conclude this work in Section 6 and offer some promising trends for future research.

The main contributions of this paper are summarised as:

1. Plenty of literatures are investigated in this brief review to give a comprehensive understanding of object detection in the deep learning era.
2. Mainly focusing on the recent progress on object detectors using deep CNNs, we review these works through a unique methodology from five main parts: architecture diagram, contextual reasoning, multi-layer exploiting, training strategy, and others.
3. Comprehensive comparisons of these detectors are summarised to offer an intuitive understanding of differences between them. Some promising trends for future research are provided after a short summary of this review.

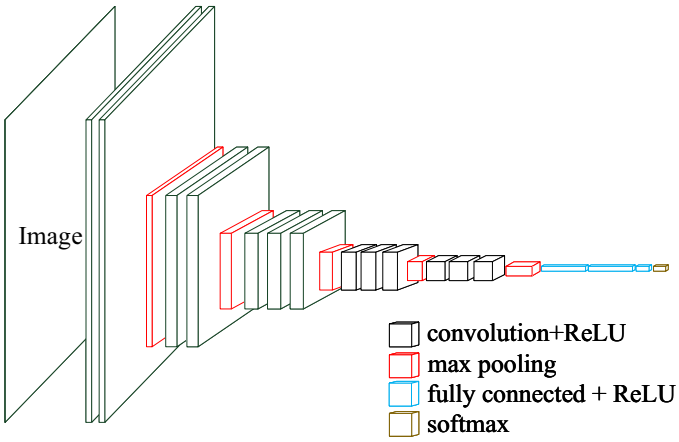
2 Backbone and detection business part

In this section, we go deep into the improvements on the backbone network and two main streams of the detection business part respectively. Firstly, we investigate the backbone from both accuracy and speed perspective. Then we introduce the main methods of the two main streams like R-CNN [24], OverFeat [73], YOLO [68] and SSD [56].

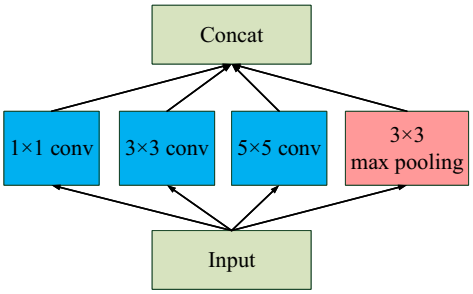
2.1 Backbone

Deep learning has got a lot of attention since AlexNet [45] won first place in the challenge of ImageNet [14] in 2012. Great improvements have been achieved both in the accuracy [38] and speed [37] of image classification. In this part, we briefly introduce some of the advanced classification architectures that have been widely applied in object detection as *backbone*, which are utilized to extract features. The development of backbone can be divided into several stages which are represented by some classic network design principles (see Fig. 3). The first is *repeat* which stacks structure with the same topology and makes the entire network becomes a modular structure. This technique starts from AlexNet and VGG [79] (Fig. 3a) and is adopted by almost all the later works. The second is *multi-path* which first appears in Inception [82] module (Fig. 3b). The input from the previous layer is divided into different paths to transform by filters with different kernel sizes, and finally, the output is concatenated by a 1×1 convolutional layer. The last is the *skip-connection* (Fig. 3c) which starts from Highway Network [81] and becomes a standard principle from ResNet [30]. It constructs the connection between high-level and low-level feature information which changes the original single linear structure.

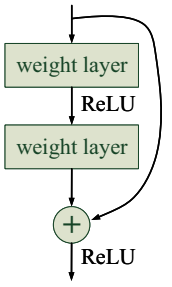
AlexNet: AlexNet [45] consists of five convolutional layers and three fully connected layers. It is a milestone study of deep learning and computer vision for introducing some



(a) Architecture of VGG-16



(b) Inception Module



(c) Skip connection

Fig. 3 Development of backbone. **a** VGG-16 [79] *repeats* the basic components of CNN: *convolution layer, pooling layer and fully connected layer*. **b** Inception module [82] uses *multi-path* and different kernel size of convolution. **c** *skip-connection* [30] connects high-level and low-level features

advanced techniques like training the network with graphics processing unit (GPU) for speeding up the operation of convolution parallelly and using the dropout to prevent from overfitting.

VGGNet: VGGNet [79] won second place in the classification task and the first place in location task in the competition of ILSVRC 2014. The small receptive field is utilized in the whole network for fewer parameters. It has two versions: VGG-16 and VGG-19. VGG-16 has been widely used because of its simple architecture, which has 13 convolutional layers, five pooling layers, and three fully connected layers.

GoogLeNet: To solve the overfitting and computing problem arising with the increasing size of the network, Inception module was introduced in GoogLeNet [82]. Using different kernel sizes of filters in the same layer helps preserve the spatial information and reduce the parameters. It has 22 layers, which is almost three times deeper than AlexNet, but it has 12 times fewer parameters than AlexNet.

ResNet/ResNeXt: ResNet [30] is one of the most successful CNNs and has been exploited in many applications including the very famous AlphaGo [78]. The idea of the

ResNet is simple yet effective, which each layer should not learn unreferenced functions but learn residual functions with references to the layer's inputs. This kind of learning makes it easier to train much deeper networks efficiently. ResNet has different architectures: ResNet-50, ResNet-101 and ResNet-152. ResNeXt [93] is the upgraded version of ResNet. It is constructed by repeating a building block that aggregates a set of transformations with the same topology. It demonstrates that it's more effective to increase the size of the set of transformations (cardinality) than to increase the depth and width. Moreover, a 101-layer ResNeXt can achieve better accuracy than ResNet-200 but with only 50% complexity.

DenseNet: Inspired by the shortcut connection of ResNet, DenseNet [38] connects each layer in the network with every other layer in a feed-forward fashion with $L(L+2)/2$ direct connections. In addition to the original features (alleviating the vanishing-gradient problem and reducing the number of parameters) of the shortcut connection, this design has new features that strengthen feature propagation and encourage feature reuse. Besides, with the help of the bottleneck layer, translation layer, and small growth rate, the network becomes narrow which can prevent from overfitting.

The models above mainly focus on the accuracy improvement of the classification by increasing the depth and width of the network. On the other hand, some architectures are putting their attention on the model size while maintaining considerable accuracy so that they can be utilized on the devices with memory and computation speed constraints.

MobileNets: MobileNet [35] is a lightweight deep neural network proposed by Google for embedded devices such as mobile phones. The core of network designs, separable convolution, can effectively reduce the number of parameters and computation at the expense of lesser performance. Separable convolution replaces traditional convolution operations with two-step convolution operations: depth-wise convolution and point-wise convolution. Subsequent MobileNet-v2 [72] mainly adds residual structure, and adds a layer of point-wise convolution before depth-wise convolution, which optimizes the bandwidth usage and further improves the performance on embedded devices.

Xception: Xception [8] is an improvement to Inception v3 [83], mainly using Depth-wise Separable Convolution to replace the original Inception v3 convolution operation, in the premise of little increase in network complexity to improve the effectiveness of the model. Xception separates the tasks related to learning space from the tasks related to learning channels by adding groups to the convolution layer, which dramatically reduces the theoretical computation complexity and the size of the model.

SqueezeNet: Based on three architecture design strategies: (1) replace 3×3 filters with 1×1 filters; (2) decrease the number of input channels to 3×3 filters; (3) downsample late in the network so that convolution layers have large activation maps, SqueezeNet [40] is a small CNN architecture. Fire module which consists of squeeze convolution layer and expand layer is used to reduce the parameter number. With further compression, the model size of SqueezeNet can be compressed to less than 0.5 MB which is $510\times$ smaller than AlexNet [45] while it can achieve AlexNet-level accuracy with $50\times$ fewer parameters.

ShuffleNet: ShuffleNet [100] utilizes two new mechanisms, point-wise group convolution, and channel shuffle, to reduce computation cost while maintaining accuracy. Experiments show that it is an extremely computation-efficient CNN architecture with comparable accuracy. Channel split is introduced in the upgrade version, ShuffleNet v2 [61], to speed up the network. Some practical guidelines for efficient network design are proposed in this work, and ShuffleNet v2 achieves a trade-off between speed and accuracy.

In addition to these models mentioned above, there are also some noticeable architectures [37, 97]. ZFNet [97] presents a method of deconvolution for visualization of convolution network, which can analyze the effect of convolution network and guide the improvement of

the network. Based on AlexNet network, ZFNet obtains a better result. SE block in SENet [37] is designed by explicitly modeling the interdependence between channels and adaptively recalibrating the channel response. The core of the SENet is squeezing and excitation operation.

2.2 Detection business part

The detection business part can be divided into two main streams according to whether there's an independent region proposal stage or not. In the part, we present some of the representative models of two-stage approach and one-stage approach.

OverFeat: Overfeat [73], one of the first advances in using deep learning for object detection, integrates three tasks of image classification, location, and detection into a framework to boost the accuracy and won the first place in the ILSVRC2013 localization competition. OverFeat is based on the multi-scale sliding-window algorithm, which is an intuitive search method of object detection.

R-CNNs: R-CNN [24], one of the most famous region-base convolutional neural networks, is the first to use deep CNN to extract feature for object detection. Firstly, it generates about $2k$ object candidates named region proposals through Selective Search [85]. Then these proposals are resized to the fixed size to fit the input size of the CNN like AlexNet. A fixed length of feature vectors is generated by the CNN and finally classified using class-specific linear support vector machines (SVMs). This simple yet effective pipeline has reached state-of-the-art performance on the benchmark datasets with momentous performance boost over all previous models, which are mainly based on DPM [23] while the computation for every region proposal is very time-consuming. The whole detection pipeline of R-CNN is shown in the above part of Fig. 4. To solve the computation and the limited image input size problem, SPPnet [29] (the middle part of Fig. 4) introduces spatial pyramid pooling to relax the constraint of the fixed input size due to the fully connected layers. More importantly, SPPnet extracts the feature maps from the entire image independent of the region proposal stage. Then it matches the proposals through spatial pyramid pooling (SPP) and generates a fixed-length vector regardless of the input size. Finally, the fixed-length representation is input into the last two fully connected layers and then classified by category-specific linear SVMs. SPPnet speeds up the R-CNN method 24-102× faster with better or comparable accuracy. Fast R-CNN [25] inherits the spatial pyramid pooling from SPPnet but modifies it as Region of Interest (ROI) Pooling which can be seen as a single-level SPP (the lower left corner of Fig. 4). It uses the bounding-box regressor instead of linear SVMs and utilizes a multi-task loss which makes the network can be trained in a single stage and no extra storage is required for feature caching during the training. This method can train a very deep detection network with a backbone VGG16 [79], testing 9× faster than R-CNN [24] and 3× faster than SPPnet [29]. At test time, the detection network processes one image in 0.3s (excluding object proposal time). Faster R-CNN [71] replaces the Selective Search [85] in the region proposal stage with the region proposal network (RPN) (the right corner of Fig. 4) which is built by several convolutional layers, which makes the network completely trainable end-to-end. With the RPN, Faster R-CNN can process an image in 0.2 seconds (including region proposal), which is 250× faster than R-CNN and 10× than Fast R-CNN, almost toward real-time. It is noticeable that the backbone of R-CNN is AlexNet [45], and SPPnet is based on ZF-5 [97] while Fast R-CNN and Faster R-CNN adopt VGG-16 [79].

YOLOs: Focusing on real-time object detection, YOLO [68] borrows ideas from the design of the architecture of GoogLeNet [82]. The input image is divided into $S \times S$ grid

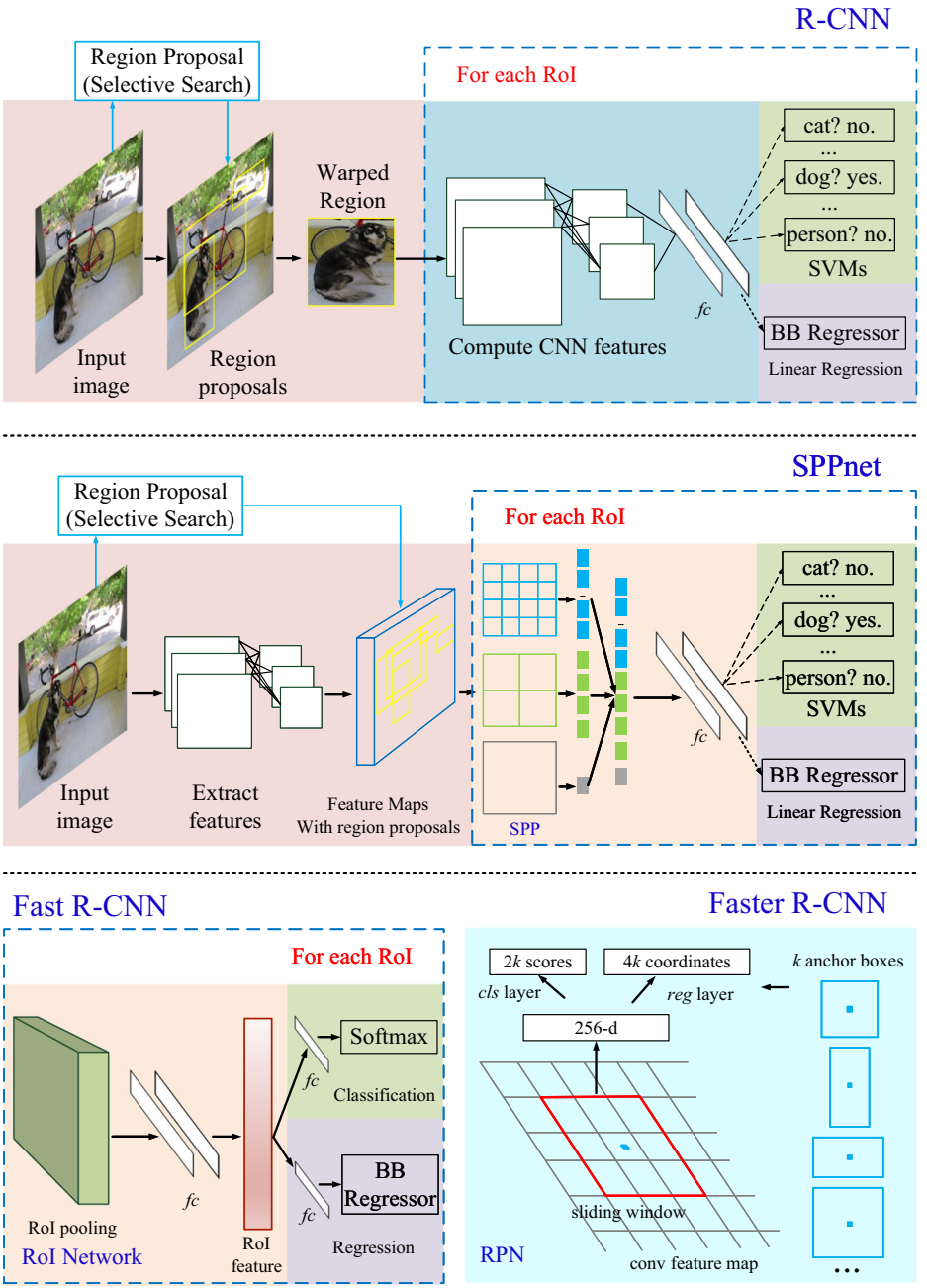


Fig. 4 Some representative object detectors of *two-stage*. R-CNN [24] is a milestone detector using deep CNN to extract features instead of handcrafted features. SPPnet [29] modifies R-CNN by extracting features on the whole input image and adopting spatial pyramid pooling to fulfil the arbitrary size of input image. Fast R-CNN [25] replaces the SPP in SPPnet with RoI network and Faster R-CNN [71] uses Region Proposal Network (RPN) instead of the original Selective Search for generating region proposals

and the grid where the center of the object lies in charge of the prediction of the object. Each grid cell outputs B bounding boxes and confidence scores for those boxes, as well as C class probabilities. The unified framework runs at 45 frames per second with the performance outperforming DPM [23] and R-CNN [24]. The architecture of YOLO can be seen in the above part of Fig. 5. To improve the precision and recall of object localization, YOLOv2 [69] adopts some advanced methods to make the detection better, stronger and faster. Briefly, the idea of anchor box is introduced from Faster R-CNN [71] and the network architecture is altered to fit the modification where the fully connected layer of the output layer is replaced by a convolutional layer. Using WordTree and joint training method, the authors train YOLOv2 simultaneously on the MS COCO [52] detection dataset and the ImageNet classification dataset. YOLOv2 gets 78.6 mAP at the speed of 40 frames per second, outperforming state-of-the-art methods like Faster R-CNN with ResNet and SSD while still running significantly faster. Based on Darknet-53 [67], which is as accurate as ResNet-101 or ResNet-152 [30] but much faster, YOLOv3 [70] makes an incremental improvement not only on the accuracy perspective but also speed. Multi-scale prediction employed to get more meaningful semantic information from the upsampled features and finer-grained information from the earlier feature map. At the image size of 320 × 320, YOLOv3 runs as accurate as SSD [56] but three times faster. It achieves similar performance but 3.8× faster compared to RetinaNet [54].

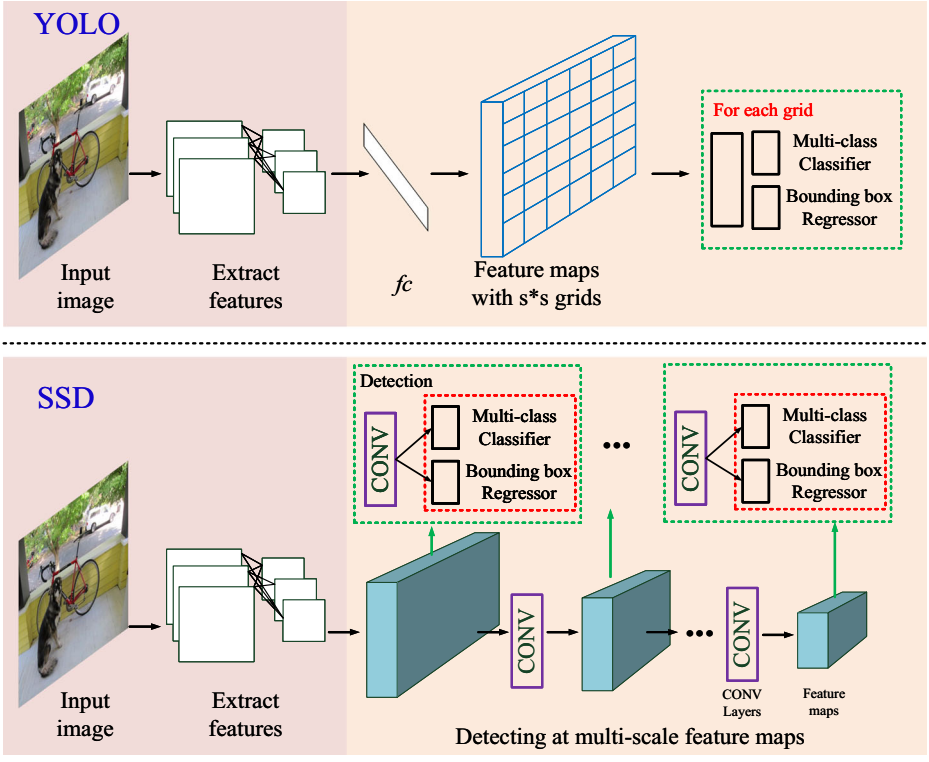


Fig. 5 Detection pipeline of representative models of *one-stage*. YOLO [68] divides the feature maps into $S \times S$ grids and detects each grid. SSD [56] detects objects at multi-scale feature maps

SSD: As one of the most successful one-stage approaches, single shot detector (SSD) [56] (the below of Fig. 5) discretizes the output space of the bounding boxes into a set of default boxes. Each feature map is located at different aspect ratios and scales in the default boxes. These default bounding boxes are essentially equivalent to Faster R-CNN's anchor boxes [71]. At prediction time, scores are generated for each object class in each default box and the box is adjusted to match the object shape. The network combines predictions from multiple feature maps with various resolutions to deal with objects of different sizes. SSD uses the reduced VGG-16 [79] like Faster R-CNN but with a better performance.

3 Improvements

3.1 Architecture diagram

The main idea of architecture design is to use deeper backbone which not only leads to improvements in classification but also in object detection. For example, Faster R-CNN [71] can achieve over 3% mAP boost by replacing the original VGG-16 [79] backbone with ResNet-101 [30]. While changing the components of CNN (*e.g.* using a deformable convolution or position-sensitive RoI pooling) or using a multi-stage structure can also get a new architecture diagram (shown in Fig. 6). For example, Mask R-CNN [31], R-FCN [10] and its descendant [80], and CoupleNet change the original RoI pooling in Fast R-CNN [25], DeepID-Net [64], DCN [11] replaces the convolution with deformable convolution to

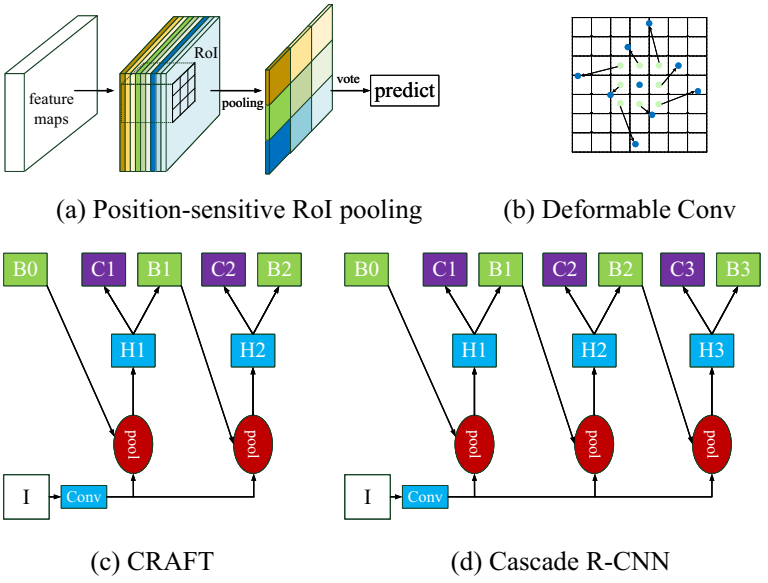


Fig. 6 The improvements of architecture of object detectors. **a** Deformable convolution used in DCN [11]. **b** Position-sensitive RoI pooling introduced in R-FCN [10] to keep the spatial information for better localization. **c** and **d** multi-stage structure where "I" is input image, "Conv" backbone convolutions, "pool" region-wise feature extraction, "H" network head, "B" bounding box, and "C" classification. "B0" is proposals in all architectures. **c** CRAFT [94] adopts it to generate compact and better localized proposals and reduce false positive in classification. **d** Cascade R-CNN [6] consists a sequence of detectors trained with increasing IoU thresholds

fit the scale variance of objects, CRAFT [94], Cascade R-CNN [6], ME-RCNN [47], STDN [105] and [50, 95] adopt a cascade structure to boost accuracy.

Mask R-CNN [31] is designed for instance segmentation and object detection, which modifies a little to Faster R-CNN by replacing RoI with RoIAlign. Just adding a branch for an object mask in parallel with the branch for bounding box regression, Mask R-CNN can be generalized to other tasks easily, *e.g.*, human pose estimation and person keypoint detection. As the third version of R-CNN [24], Faster R-CNN [71] achieves a speed-accuracy trade-off for using the Region Proposal Network (RPN) to generate object proposals for detecting. Taking Faster R-CNN as the baseline, R-FCN [10] borrows idea from fully convolutional neural network (FCN) [58] which is applied for semantic segmentation. Firstly, feature maps are generated by a modified 101-layer ResNet [30] whose last average pooling layer and fully connected layer are discarded and the output of the last convolutional layer is resized to 1024-d from 2048-d. Then an RPN proposes candidate RoIs based these feature maps. A position-sensitive RoI (PSRoI) pooling (shown in Fig. 6a) is designed to keep the spatial information of object regions and generates scores for each RoI (the size of RoI is 3×3). The result is generated by averaging voting on each RoI. The use of a fully convolutional network can avoid the loss of spatial information that brought by the fully connected layer applied to classification. Online Hard Example Mining (OHEM) [76] (after-mentioned) is applied during the training. CoupleNet [106] proposes the idea to couple the global structure with local information for object detection. The object proposals generated by the Region Proposal Network (RPN) are input into the coupling module with two branches. The local part information of the object is encoded by the position-sensitive RoI (PSRoI) pooling in one branch, and the global and context information are captured by the RoI pooling in the other branch. Different coupling strategies and normalization ways are investigated to make full use of the complementary advantages between the global and local branches.

In the architecture of DeepID-Net [64], a new deformation constrained pooling (def-pooling) layer models the deformation of object parts with geometric constraints and penalty. Besides, a new pre-training strategy is introduced to learn more suitable feature representations for the object detection task with excellent generalization capability. To fit the geometric variations adaptively, two modules named deformable convolution (see Fig. 6b), and deformable RoI pooling are introduced in DCN [11] to enhance the transformation modeling capability of CNNs. The two modules augment the spatial sampling locations with additional offsets and learn the offsets from the target tasks without extra supervision. The original counterparts in existing Faster R-CNN can be replaced by the new modules, and the new network deformable convolutional networks can be trained end-to-end by back-propagation. Extensive experiments validate the performance boost of this approach including semantic segmentation and object detection.

Based on the philosophy of "divide and conquer", CRAFT [94] divides both the region proposal and object classification into two sub-tasks (illustrated in Fig. 6c). An additional Fast R-CNN is attached to the original region proposal network (RPN) to provide more compact and better-localized object proposals in the proposal generation stage. In object classification, two Fast R-CNNs are employed in a cascade structure to reduce false positives by capturing both inter- and intra-category variances. Similarly, [95] exploits the features in all layers to reject easy negatives via cascade rejection classifiers and evaluates left proposals using a scale-dependent pooling method. Group recursive learning is utilized in [50] with multi-stage detection. The proposed architecture consists of three cascaded networks which respectively learn to perform weakly-supervised object segmentation, object proposal generation, and recursive detection refinement. A multi-stage object detection architecture, Cascade R-CNN [6], is proposed to address problems that detection

performance tends to degrade with increasing IoU thresholds (Fig. 6d). It is composed of a sequence of detectors trained with increasing IoU thresholds, to be sequentially more selective against close false positives. These detectors are trained stage by stage, leveraging the observation that the output of a detector is a good distribution for the training of the next higher quality detector. Taking Fast R-CNN [25] as a baseline with a backbone VGG-16 [79] or ResNet-101 [30], ME R-CNN [47], representing for Multi-Expert Region-based CNN, consists of multiple experts and constructed on top of the R-CNN framework. Focusing on better capturing the appearance variations brought by various shapes, poses, and viewing angles, ME R-CNN is equipped with three experts each responsible for objects with particular shapes: horizontally elongated, square-like, and vertically elongated. Besides, for better data augmentation, the exhaustive search is applied in the training stage for providing a compact but effective set of regions of interest (RoIs) for object detection. To solve the scale problem in object detection, a one-stage object detector named Scale-Transferrable Detection Network (STDN) [105] using scale-transfer module (STM) and DenseNet [38] is proposed. DenseNet is altered to integrate high-level semantic information and low-level details to achieve more powerful features. STM consists of pooling and super-resolution layers with no additional parameters and computation.

We have reviewed some of the representative models which design new architecture diagram and highlights of these models are illustrated in Table 2. Not only the networks talked above can be categorized into this subsection, but also many other methods like DSSD [20], MR-CNN [22], GBD-Net [98], which are classified into other parts.

3.2 Multi-layer exploiting

Detectors like SPPnet [29], Fast R-CNN [25], Faster R-CNN [71] and YOLO [68] are based on the top-most feature of deep CNN and do not make full use of the bottom details. SSD [56], DSOD [74] and MS-CNN [5] make predictions on multi-scale features. SSD [56] adopts default boxes, matching strategy introduced in MultiBox [17] and several extra convolutional layers in which multiple feature maps with the different resolution are combined to naturally handle objects of various size (architecture can be seen in Fig. 7a). Nevertheless, the layers from the bottom of a CNN have weak semantic information, which will harm their representational capacity for small object recognition. In this part, we focus on methods *concatenating multi-scale features* of CNN [2, 7, 41, 43, 49, 104] or *using top-down pathway and conduct skip connections* [9, 53, 77] or combining both [9, 44, 99] (see Fig. 7). These operations will not only help enhance the power of features but also help introduce contextual information (aftermentioned).

Several recent works [7, 41, 49, 104] investigate the architecture of SSD and propose some useful methods to merge features of the original SSD architecture. R-SSD [41] analyzes how to use features effectively to improve the performance of SSD. Rainbow concatenation, pooling, and deconvolution are performed simultaneously to create feature maps with an explicit relationship between different layers rather than growing layers close to the input data, *e.g.*, by replacing VGG-16 [79] with ResNet-101 [30]. Multi-level feature fusion method is proposed to add contextual information for small object detection in SSD pipeline in Feature-Fused SSD [7] architecture. Two modules, concatenation module, and element-sum module, are carefully investigated and features from different layers of VGG-16 (*e.g.*, `conv3_3`, `conv4_3`, `conv5_3`) are also explored. FSSD [49] uses a lightweight feature fusion module to make the network run faster. Features from different layers with various scales are concatenated first and then applied to generate a set of pyramid features and finally predicted independently. ESSD [104] extends the shallow part of SSD through

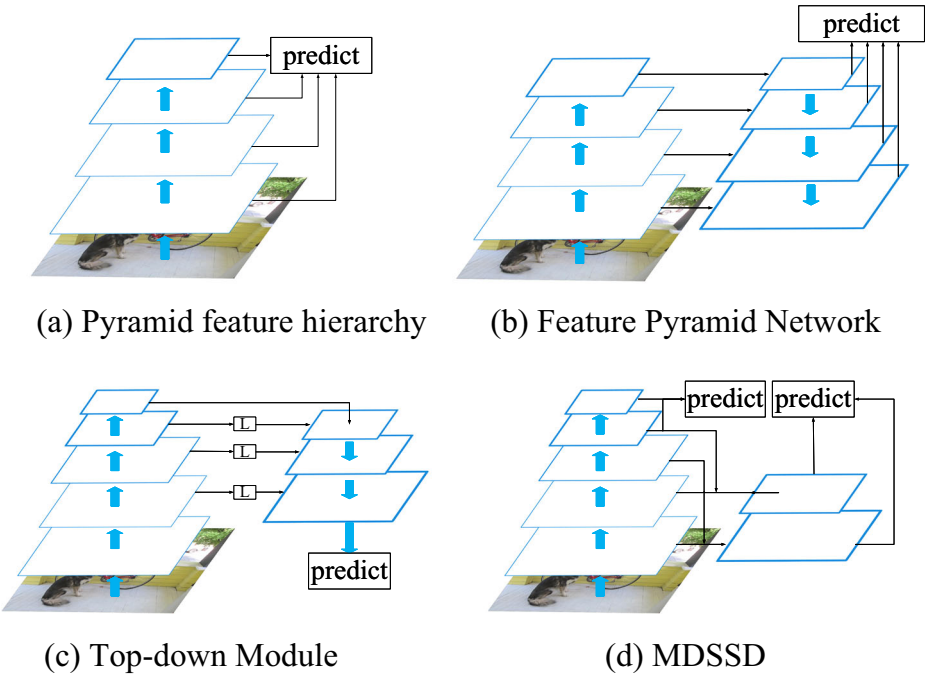


Fig. 7 Architecture of multi-scale fusion and top-down pathway. **a** Multi-scale detection used in SSD [56]. **b** Feature Pyramid Network (FPN) [53] is a classic top-down structure. **c** Top-down Module (TDM) [77] where "L" represents lateral connection module and the prediction is implemented at the top of the top-dwon module. **d** A combination of multi-layer concatenation and top-down in MDSSD [9]

Extension Module which consists of several convolutional layers and deconvolutional layers. This kind of extension also focuses on the contextual information which is similar to the two feature fusion modules in Feature-Fused SSD.

Meanwhile, there are also several proposed methods trying to improve the performance of the two-stage modules. ION [2] uses skip pooling and spatial recurrent neural networks to integrate information at multiple layers, and detects objects using these features. To solve the coarseness of the feature maps generated by Region Proposal Network (RPN) [71] which will harm small object detection and precise localization, a deep hierarchical network named HyperNet [43] is proposed to handle region proposal generation and object detection jointly. HyperNet is mainly based on Hyper Feature which aggregates hierarchical feature maps first and then compresses them into a uniform space. The Hyper Feature well incorporates deep but highly semantic, intermediate but really complementary, and shallow but naturally high-resolution features of the image, thus enabling us to construct HyperNet by sharing them both in generating proposals and detecting objects via an end-to-end joint training strategy.

Instead of using multi-layer feature aggregation, a top-down pathway and lateral connection are introduced to merge multi-layer features. Instead of combining high-level and low-level features with skip connections, top-down contextual information is required when selecting the right features from low-level. Top-down modulations are proposed to incorporate fine details into the detection framework. The TDM [77] is applied as a supplement to the standard bottom-up, feedforward ConvNet, connected using lateral connections (shown in Fig. 7c). These connections modulate the low layer filters and the top-down network

handles the selection and integration of contextual information and low-level features. The TDM can be easily integrated into the state-of-the-art two-stage detection framework and gets a significant improvement on detection including small objects. After investigating how to construct feature pyramids with marginal extra cost, a top-down architecture with lateral connections is proposed for building high-level semantic feature maps at all scales. This architecture named Feature Pyramid Network (FPN) [53] (Fig. 7b) can be easily embedded into the Faster R-CNN system, and the joint model achieves state-of-the-art results on the COCO detection benchmark with a speed of 6 FPS on a single GPU. MDSSD [9] (Fig. 7d) designs several multi-scale Deconvolution Fusion Modules which is a modified top-down architecture with skip connection to provide a significant boost on detection of small objects. The high-level semantic features from different depth are fed into several deconvolution layers to produce higher resolution features and then merged with the low-level features to achieve skip connections.

Inheriting both the multi-scale concatenation and top-down structure, RefineDet [99] and RON [44] introduce some new techniques to improve performance. Based on one-stage approach, RON [44] associates the best of two-stage approach and one-stage approach. RON is an efficient and practical framework for object detection. Reverse connection is applied to address multi-scale object localization in a top-down model and the objectness prior is employed to solve the negative sample mining and reduce the searching space of objects. Optimizing reverse connection, objectness prior and object detector jointly by a multi-task loss function, RON can predict detection results from different scales of feature maps. RefineDet [99] inherits the merits of both one-stage approach and two-stage method while discarding their shortcomings. It is composed of two inter-connected modules: the anchor refinement module (ARM) and the object detection module (ODM). ARM tries to identify and remove negative anchors to reduce search space for the classifier and coarsely adjust the locations and sizes of anchors for better input of the regressor. ODM predicts the bounding boxes and classes from the output of ARM. There is a transfer connection block (TCB) to transfer the features in the ARM to predict locations, sizes, and class labels of objects in the ODM. The connections between TCBs are achieved by a top-down pathway with lateral connections with ARM and ODM. The whole network is trained end-to-end with the multi-task loss function.

In this subsection, these methods try to combine features from multiple layers to achieve better feature representation. Intuitively, combining high-level features with high semantic information and low-level with high resolution can bring performance boost. Different combining mechanisms are proposed and we highlight them in Table 2.

3.3 Contextual reasoning

Context [86] has been proven to play an important role in visual recognition. In the task of object detection, it's always to locate some objects with low resolution which is known as *small objects* without considering its background information. Contextual or semantic information can be introduced into the detector through several ways, such as *concatenating features* [7, 9, 20, 102, 104], using a *semantic segmentation subnet* [22, 75, 102], and *other techniques* [57, 98].

As mentioned before, SSD is a simple architecture that detects objects in a dense manner with several different convolution layers of various resolution. While the biggest problem in SSD is that the low layer with high resolution but has less semantic information. Deconvolution module is introduced in DSSD [20] to add contextual information for object detection, especially for small objects. DSSD replaces the original VGG-16 [79] in SSD

with ResNet-101 [30] and adding five extra deconvolutional modules with one convolution layer representing for different resolutions, named DSSD Layers which is similar to the original SSD Layers. The change of backbone and the added extra DSSD Layers will not only boost the accuracy but also increase the inference time. In a similar way, contextual information is introduced by multi-level feature fusion, extension module and Deconvolution Fusion Modules in Feature-Fused SSD [7] , ESSD [104] and MDSSD [9] respectively.

Also focusing on improving SSD, a novel single-stage object detector named Detection with Enriched Semantics (DES) [102] which consists of two main branches is proposed. A segmentation branch which uses the idea of weakly supervised semantic segmentation is introduced to solve the problem that low-level feature map doesn't have high-level semantic information. A global activation module is utilized to provide global context information and pure channel-wise feature map learning in high-level layers. Experiments show that this method excels in both accuracy and speed. For its flexibility and simplicity, it can be applied to other two-stage or one-stage object detectors.

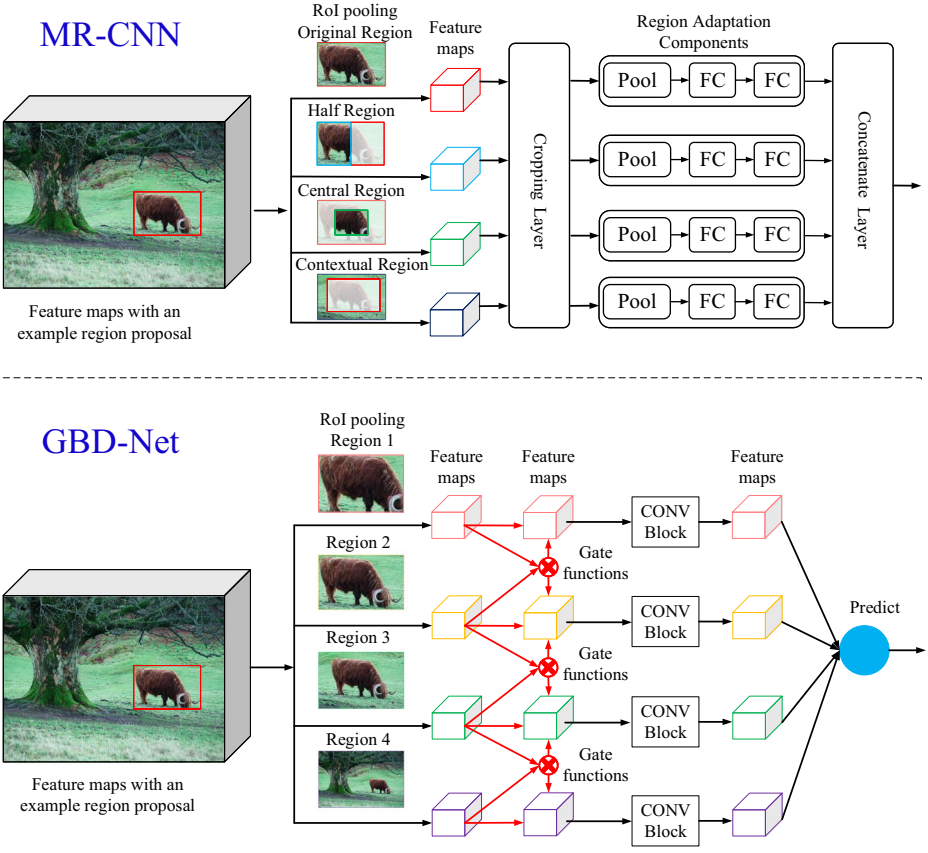


Fig. 8 Architecture of MR-CNN [22] and GBD-Net [98]. MR-CNN with region adaptation components of ("pool" operation in the figure stands for "adaption max pooling") can be extended to also learn semantic segmentation-aware CNN features. Gate functions in GBD-Net are defined for controlling the message passing rate

Based on multi-region deep CNN, semantic segmentation-aware features are encoded in MR-CNN [22] (see the above of Fig. 8). Contextual information is introduced through the contextual region and semantic segmentation module. Experiments show that MR-CNN surpasses any other previous work by a significant margin. Based on Fast R-CNN, Inside-Outside Net (ION) [2] is introduced to leverage contextual information and multi-scale knowledge for object detection. This architecture consists of a 2× stacked 4-directional IRNN for contextual information and multi-layer ROI pooling with skip connection and normalization for improving localization accuracy. It achieves state-of-the-art results on both PASCAL VOC [19] and MS COCO [52], and it's particularly effective at improving the performance of small objects detection. Based on Faster R-CNN [71] and instead of inheriting a bottom-up, feedforward structure of CNNs, [75] adopts the idea that humans, top-down information, context, and feedback play an important role in object detection. It augments Faster R-CNN with a semantic segmentation network from ParseNet and uses it for top-down contextual priming and top-down iterative feedback using two-stage training. Results indicate that it improves the performance on object detection, semantic segmentation, and region proposal generation.

Instead of simply concatenating features, a novel gated bi-directional CNN (GBD-Net) [98] (shown in the below of Fig. 8) is proposed to pass messages between features from different regions during both feature learning and feature extraction, which can be exploited through convolution in two directions and can be conducted in different layers. Therefore, local and contextual visual patterns can validate the existence of each other by learning their nonlinear relationships and their close iterations are modeled in a much more complex way. SIN [57] makes use of two kinds of context including scene contextual information and object relationships within a single image. It can be incorporated into a typical detection framework (*e.g.* Faster R-CNN) with a graphical model, formulating object detection as a problem of graph structure inference. When given an image, the objects are considered as nodes in a graph and relationships between the objects are modeled as edges in such a graph.

Except for multi-layer feature combination like methods in the above part, contextual information can also be introduced by the semantic segmentation branch and other techniques. Highlights of these methods are shown in Table 2 and experimental results are compared in Section 5.

3.4 Training strategy

With the increase of the depth of the backbone, the arising of more challenging detection datasets, and the special goal of object detection, the training of a detector is time-consuming and hard. Several training strategies are introduced in this part to deal with above problems, such as *class imbalance* [54, 76, 87], *non-maximum suppression* [3, 34] and *large mini-batch* [65].

Class imbalance The task of object detection is to detect a various number of objects with different resolutions and we use a reduced CNN (backbone) applied for classification and convert object detection into a classification problem. This introduces a significant imbalance between the number of annotated objects (foreground) and the number of backgrounds during the training stage. Two-stage approaches set the ratio between foreground and background as a fixed number (*e.g.* 1:3). To solve the imbalance problem between background and foreground, a training strategy named online hard example mining (OHEM) [76] is proposed based on bootstrap or hard example mining which is utilized when training SVMs. Hard examples are selected during training by loss instead of using hyperparameters to

determine the ratio between positives and negatives, thus simplifying training. Experiments show that this training algorithm can lead to better training convergence and comparable accuracy improvements in detection on standard benchmarks. What's more, it is orthogonal with the region-based object detectors like Fast R-CNN [25], SPPnet [29]. Instead of searching for hard examples based on the original dataset, A-Fast-RCNN [87] adopts a simple but efficient way that generates hard positives through adversarial networks. It just focuses on examples with occlusions and deformations and proposes two subnets: Adversarial Spatial Dropout Network (ASDN) which learns how to occlude a given object and Adversarial Spatial Transformer Network (ASTN) which creates deformations on the object features. The original detector Fast R-CNN and the two adversarial networks are trained jointly. Similarly, to solve the imbalance problem in the one-stage method which is blamed for the lower performance compared to the two-stage approach, a dynamically scaled cross entropy loss function, Focal Loss [54] is introduced to improve the accuracy by reshaping the standard cross entropy loss such that it down-weights the loss assigned to well-classified examples. Focal Loss focuses on training on a sparse set of hard examples and prevents the vast number of easy negatives from overwhelming the detector during training. A single, unified network, RetinaNet is designed to validate the effectiveness of Focal Loss. This network is composed of a backbone-ResNet-based Feature Pyramid Network (FPN) and two task-specific subnetworks-one is class subnet for object classification and the other is box subnet for bounding box regression.

Non-maximum suppression Non-maximum suppression (NMS) is an independent part which is attached to the object detection pipeline as the last step in both one-stage and two-stage approach. Its function is to reduce the number of detections and select the exact bounding box with the highest score by setting the score of neighboring detections to zero directly. An algorithm, named Soft-NMS [3], is proposed to deal with this problem. It decays the detection scores of all other objects as a continuous function of their overlap with the bounding box with the maximum score. Soft-NMS obtains comparable improvements for the coco-style mAP metric on standard datasets like PASCAL VOC 2007 and MS-COCO by just replacing the NMS with Soft-NMS without any extra hyper-parameters. Since Soft-NMS does not require any additional training and is simple to implement, it can be easily integrated into any object detection pipeline. Instead of modifying the post-processing algorithm–NMS, a new network architecture named Gnet [34] is designed to perform NMS, using only boxes and their score. A loss that penalizes double detections and joint processing of detections are used in the building of this network. Gnet makes the entire detection pipeline become a real end-to-end trainable network without any post-processing part. Experiments demonstrate that with enough training data, the proposed Gnet is a suitable replacement for traditional NMS both for one-stage and two-stage method. It's a good idea to perform NMS with a neural network while it does cost time. A relation module [36], focusing on finding the relation between the region proposals instead of processing them individually, is proposed to perform duplicate removal instead of the original greedy NMS and Gnet. Experimental results show that this module is effective in improving duplicate removal step with comparable process speed compared to the greedy NMS and much faster than Gnet.

Large mini-batch There is a trend to use a very large mini-batch size to speed up the training of CNN-base image classification [27, 96] while the mini-batch size in object detection remains very small (*e.g.* 2-16) due to the GPU memory constraint. To deal with the potential problems brought by small mini-batch size in object detection, such as long training time,

failure for batch normalization and imbalance between positives and negatives, a large mini-batch detector, MegDet [65] is proposed. Variance equivalence assumption and new warmup strategy are introduced to help the convergence and higher performance. More importantly, Cross-GPU Batch Normalization is advised to make the training converge quickly and it's the first time to use BN in object detection training. Experimental results show that these useful training strategies make the training faster (from 33 hours to 4 hours), and achieve even better accuracy.

Some training problems like class imbalance, NMS, and large mini-batch are talked in this part. Strategies tried to solve these are highlighted in Table 2. For tricks to train object detection networks, you can refer [103].

3.5 Others

Apart from techniques talked above, some object detectors specially designed for *real-time object detection* [33, 48, 62, 80, 88, 90, 91], ideas borrowed from other kind of tasks like *recurrent neural networks (RNN)* [2, 66], *generative adversarial networks (GANs)* [60, 87, 92] and *weakly supervised learning* [84, 101], and new backbone specially designed for object detection [51].

Real-time object detection Intuitively, the two-stage approach focuses on the accuracy perspective in object detection while the one-stage approach performs better in speed perspective. It's true that most methods of the one-stage approach can run towards real-time like SSD [56], YOLO [68–70]. Actually, there are also some other methods designed to be real-time object detectors [33, 48, 62, 80, 90, 91]. The main technique to design a real-time object is to use a light-weight backbone like Xception [8], SqueezeNet [40] or borrow an idea from them to design a similar lightweight network. PVANet [33] proposes a novel network structure, which is an order of magnitude lighter than other state-of-the-art networks while maintaining accuracy. Based on the basic principle of more layers with fewer channels, this new deep neural network minimizes its redundancy by adopting recent innovations including C.ReLU and Inception structure. This network can be trained efficiently to achieve reliable results while the required compute is less than 10% of the recent ResNet-101. Focusing on the application on autonomous driving with model size, inference speed, and comparable efficiency constraint, SqueezeDet [91] adopts several convolutional layers to extract features and compute bounding boxes and class probabilities simultaneously. This model is fully convolutional, which leads to a small model size and better energy efficiency. Light-Head R-CNN [48] uses a thin feature map and a light R-CNN subnet which consists of pooling and single fully-connected layer to make the head of the network as light as possible. With a tiny network (*e.g.*, an Xception like network), Light-Head R-CNN gets 30.7 mmAP at 102 FPS on COCO [52]. Tiny SSD [90], a single-shot detector for real-time embedded object detection, is composed of a highly optimized, non-uniform Fire sub-network stack borrowing from SqueezeNet and a non-uniform sub-network stack of highly optimized SSD-based auxiliary convolutional feature layers explicitly designed to minimize model size while maintaining object detection performance. The results show that Tiny SSD achieves a model size of 2.3MB (about 26× smaller than Tiny YOLO [69]) while still maintaining an mAP of 4.2% higher than Tiny YOLO on VOC 2007 [18]. To design a fast and efficient object detection system, F-YOLO [62] investigates three aspects: (1) Network architecture, (2) Loss function and (3) Training data. Inspired by DenseNet [38], Yolo-v2 [69] and Single Shot Detector (SSD), dense map with stacking and deep but narrow network architecture is altered in F-YOLO, which contains only 15M parameters

compared to 138M in VGG-16 model [79]. Feature Map-NMS (FM-NMS) is introduced to finish the network distillation, which is the first one to apply distillation on a single pass detector (Yolo). What's more, distillation loss employed for both labeled and unlabeled data is designed to fit the network distillation. As a modification of R-FCN [10], R-FCN-3000 [80] focuses on large scale object detection. The main idea of R-FCN-3000 architecture is to decouple localization and classification by predicting objectness and classification scores independently. Same as R-FCN, RPN is used for generating proposals which are fed into a *super-class* detection branch (like R-FCN) which jointly predicts scores for each super-class. Extra fully convolutional layers are applied to generate per class scores and then these scores are averaged inside the RoI to get the classification probability which is multiplied with the super-class detection probability for detecting 3000 classes.

RNN Recurrent Neural Networks (RNN) [89] is a kind of neural network where connections between neurons form a directed graph along a sequence which has been applied to process sequences of inputs. Recurrent Rolling Convolution (RRC) architecture over multi-scale feature maps is introduced in [66] to construct object classifiers and bounding box regressors which are "deep in context". Its goal is to achieve accurate detection with a high IoU threshold. Results on KITTI [21] dataset show the effectiveness of the RRC with a reduced VGG-16 [79]. A 2x stacked 4-directional IRNN module is proposed in ION [2] to introduce context information to improve localization accuracy.

GANs Generative Adversarial Networks (GANs) [26] consists of two networks: one generates candidates (generative) and the other evaluates them (discriminative) (thus the "adversarial"). Inspired by this, A-Fast-RCNN [87] uses an adversarial network to generate hard examples for training instead of selecting them based on the original dataset. Similarly, [92] and [60] also adopt adversarial networks to generate examples to boost the accuracy of object detection.

Weakly supervised Object detectors find and classify objects through datasets with annotated ground-truth bounding boxes and class labels which are marked by hand. This kind of annotation costs money and time, so there are not so many object detection datasets as ImageNet [14] which has millions of labeled images. Weakly supervised training can help boost performance with unannotated images through "turn" classifier to a detector. [101] proposes weakly-supervised to a fully-supervised framework for object detection (W2F). Given an image collection with only image-level labels, Multiple Instance Learning (MIL) is employed to train a weakly-supervised detector, and then a pseudo-ground-truth excavation algorithm is exploited to seek pseudo-ground-truth boxes, which are in turn refined using pseudo-ground-truth adaptation algorithm and applied to train a supervised detector. [84] incorporates external knowledge about object similarities from visual and semantic domains in modeling the category-specific differences and then transferring this knowledge for adapting an image classifier to an object detector for a "weak" category.

New backbone The backbones for object detection are borrowed from the classification task. However, there is a gap between the classification and the object detection problem, which not only needs to recognize the category of the object classes but also spatially localize the bounding boxes of different sizes. First focusing on the gap between image classification and object localization, DetNet [51] is a novel backbone network specifically for the object detection task. Taking ResNet-50 as the baseline, extra stages are introduced to maintain the spatial resolution of the features for object detection and a low complexity

dilated bottleneck structure is employed to keep the efficiency of the network. State-of-the-art results have been obtained for both object detection and instance segmentation on the MS COCO [52] benchmark based on DetNet (4.8G FLOPs) backbone.

4 Evaluation

4.1 Benchmark

PASCAL VOC 2007 [18] and 2012 [19] are the two main standard benchmarks that have been widely applied in object detection. There are 20 categories of objects in PASCAL VOC dataset. PASCAL VOC 2007 consists of 9,963 images totally in which 5,001 images for training and validation and 4,952 images for testing. All of these images are annotated with the class label and ground-truth bounding boxes. The PASCAL VOC 2012 is an extended version of VOC 2007 which contains a total of 22,531 images. The `trainval` set contains 11,540 images and the `test` set has 10,991 with no public ground-truth bounding boxes available. All of the methods should submit their test results to the evaluation server of PASCAL VOC.

MS COCO [52] is a new and more complex object detection benchmark starting from 2014. Its goal is to improve the state-of-the-art in object recognition by placing the question of object recognition in the context of the broader question of scene understanding. There are more than 200,000 images and 80 object categories in MS COCO. Specificly, the training set contains 80,000 images, the validation set consists of 40,000 images, and the test set contains 80,000 images. The object detection challenge held by MS COCO every year has generated many fantastic works. Similar to PASCAL VOC 2012, the challengers should submit their results to the evaluation server for the test evaluation. More specifically, the original `train/val` set which contains 83K/41K images is split into 118K `trainval35k` for training and 5K `minival` for testing.

Apart from the two standard benchmarks for general object detection, ImageNet [14] classification dataset is employed for the pre-training of the backbone and object detection dataset is also applied for training of the detectors like R-CNN [24], OverFeat [73] and SPPnet [29]. There are also some datasets for special object detection like KITTI [21] for autonomous driving, Caltech [15, 16] for pedestrian detection, and CelebA [55] for face detection.

4.2 Metrics

The metrics for evaluating object proposals are functions of intersection over union (IOU) or Jaccard index that measures the difference between the predicted bounding box and the corresponding ground-truth. the area of overlap IOU between the predicted bounding box B_p and ground-truth B_{gt} is formulated as:

$$IOU(B_p, B_{gt}) = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \tag{1}$$

Based on IOU, recall can be computed as the ratio between ground-truth and predicted bounding boxes above a certain IOU overlap threshold. Therefore, three metrics are proposed to evaluate objectness detection methods as follows:

– *recall-proposal curve*: which illustrates recall for different number of proposals.

- *recall-overlap curve*: which demonstrates the variation of recall under different IOU overlap threshold.
- *average recall (AR)*: which computes the area under "recall-overlap" curve in a range of overlap values (*e.g.*, 0.5-1).

Average precision (AP) and mAP overall object classes are two standard metrics for evaluating object detection methods. Each bounding box will be assigned a score (likelihood of the box containing an object). Based on the predictions a precision-recall curve (PR curve) is computed for each class by varying the score threshold. The average precision (AP) is the area under the PR curve. While precision stands for the fraction of detections that are true positives, recall measures the fraction of positives that are correctly detected. First, the AP is computed for each class and then averaged over the different classes. The end result is the mAP.

The detection output is assigned to true positive if the IOU between the predicted bounding box and ground-truth exceeds a predefined threshold (*e.g.*, 0.5). Otherwise, the detection is considered as a false positive. In addition, if multiple detection outputs overlap with the same ground-truth object, only one will be set as true positive and the others are considered as false positives (this process is called non-maximum suppression).

In PASCAL VOC [19] datasets, the area overlap threshold is set to be 0.5 (written as mAP@0.5). A new evaluation metric is proposed for MS COCO [52]. The mAP is averaged over ten different IOU thresholds, from 0.5 to 0.95 with a step of 0.05 (written as mAP@0.5:0.95). Actually, both mAP@0.5 and mAP@0.5:0.95 are utilized to evaluate methods evaluated on MS COCO while mAP stands for mAP@0.5:0.95 which is the primary metric. Additionally, mAP_S is used for evaluation for small objects whose resolution is under 32×32.

Despite the wide acceptance of average precision (AP), a recent work [63] points out its numerous shortcomings: (i) the inability to distinguish very different RP curves, and (ii) the lack of directly measuring bounding box localization accuracy. Localization Recall Precision (LRP) Error is proposed to deal with these shortcomings. LRP Error consists of three components related to localization, false negative (FN) rate, and false positive (FP) rate. Representing the minimum achievable LRP error, Optimal LRP determines the "best" confidence score threshold for a class, which balances the trade-off between localization and recall-precision. Experiments show that it provides richer and more discriminative information than AP though it hasn't been widely employed.

5 Experimental comparison

[39] presents a comprehensive comparison over SSD [56], Faster R-CNN [71], R-FCN [10] and [28] compares some representative models like YOLO [68, 69], ION [2] and Fast R-CNN [25]. In this part, we give a comprehensive but straightforward comparison of different methods talked in this paper from both accuracy and speed aspect.

First, we give an overall review of the results of classic methods in Table 1, improvements in Table 2 and real-time object detection in Table 3. With the same idea of the improvements discussed in Section 3, the experimental results are listed in the same order and topology. Most improvements of the two-stage approach are based on classic methods like R-CNN [24], Fast R-CNN [25] and Faster R-CNN [71] and most of improved one-stage methods are supported by SSD [56]. Besides, some techniques like HyperNet [43] and RefineDet [99] inherit advantages from both one-stage and two-stage approach. It's no doubt that two-stage

Table 1 Classic methods

Method	Backbone	Input size	FPS	mAP (%)			Highlights
				VOC 2007	VOC 2012	COCO	
R-CNN [24]	AlexNet	227 × 227	<0.1	58.5	53.3	-	**Highlights**: First to apply CNN on object detection with siginificant improvement over previous detectors; **Disadvatages**: the pipeline cannot be trained end-to-end (CNN, SVM); Training is expansive and testing is slow.
SPPnet [29]	ZF-5	∼ 1000 × 600	<1	59.2	-	-	**Highlights**: Sharing features over the full image, First to integrate CNN with SPP; **Disadvantages**: better than RCNN on only speed; Training is also slow.
Fast R-CNN [25]	VGG-16	∼ 1000 × 600	0.5	70.0	68.4	19.7	**Highlights**:First end-to-end trained detecor (ingoring selective search); a RoI pooling layer is proposed; Much faster than SPPnet; **Disadvantages**: Additional selective search becomes the new problem; Too slow for real-time application.
Faster R-CNN [71]	VGG-16 (ResNet-101)	∼ 1000 × 600	7	78.8	70.4	21.9 (27.2)	**Highlights**: Propose a Region Proposal Network(RPN) which is better than SS in both speed and quality; **Disadvantages**: Training is complex due to RPN; Still runs slow.

Table 1 (continued)

Method	Backbone	Input size	FPS	mAP (%)			Highlights
				VOC 2007	VOC 2012	COCO	
YOLO [68]	GoogLeNet*	448 × 448	45	63.4	57.9	-	**Highlights**: First unified object detector, detecting objects in S*S grids without region proposal process; Significantly faster than previous detectors; **Disadvantages**: Accuracy falls far behind than many detectors; Not good at localizing small objects.
SSD [56]	VGG-16	300 × 300	46	77.2	75.8	23.2	**Highlights**: First accurate and efficient unified object detector, detecting objects in a multi-scale layers; Significantly more accurate than YOLO; **Disadvantages**: Struggle to detect small objects.
YOLOv2 [69]	Darknet-19	544 × 544	67	76.8	73.4	21.6	**Highlights**: Propose an efficient backbone Darknet-19; Use strategies to improve both speed and accuracy; Can detect over 9000 object categories; Achieve high accuracy and speed; **Disadvantages**: Not good at detecting small objects.
YOLOv3 [70]	Darknet-53	320 × 320	45	-	-	28.2	**Highlights**: Borrow multi-scale prediction from SSD and RPN; More powerful backbone Dark-net 53; Achieve high improvements over YOLOv2 in both speed and accuracy; **Disadvantages**: Still has problem when detecting small objects.

Table 2 Experimental comparison

	Method	Backbone	Input size	FPS	mAP (%)			Highlights
					VOC 2007	VOC 2012	COCO	
Architecture Diagram	R-FCN [10]	ResNet-101	∼ 1000 × 600	9	80.5	77.6	29.9	**Highlights**: Fully convolutional detection framework; Design a set of position sensitive score maps using a RoI-wise subnet; Faster than Faster RCNN while keeping comparable accuracy; **Disadvantages**: Training is complicate.
	Mask R-CNN [31]	FPN	∼ 1000 × 600	5	-	-	35.7	**Highlights**: An extension of Faster RCNN with instance segmentation; Design RoI Align to replace the RoI pooling for pixel-to-pixel alignment; Can run at 5FPS.
	CoupleNet [106]	ResNet-101	∼ 1000 × 600	8	82.7	80.4	34.4	**Highlights**: Add another branch to capture context information by RoI pooling based on R-FCN; Investigate different coupling strategies and normalization.
	Light-Head [48]	ResNet-101	1200 × 800	-	-	-	40.8	**Highlights**: Use a large-kernel separable convolution (ROI warping) to produce "thin" feature maps with smaller channel number; Design a cheap R-CNN subnet; Highly efficient while keeping accuracy;
	DCN [11]	FRCNN	∼ 1000 × 600	-	+0.6	-	+2.7	**Highlights**: Design deformable convolutin and deformable RoI pooling to enhance transformation modeling capability of CNNs; More accurate than plain counterpart Faster RCNN with ResNet-101 (FRCNN).

Table 2 (continued)

	Method	Backbone	Input size	FPS	mAP (%)			Highlights
					VOC 2007	VOC 2012	COCO	
	CRAFT [94]	VGG-16	~ 1000 × 600	-	75.7	71.3	-	**Highlights**: Attach the RPN with an additional Fast RCNN to provide more compact and better proposals; Use two Fast RCNN when classification.
	ME-RCNN [47]	ResNet-101	~ 1000 × 600	-	78.7	76.1	22.8	**Highlights**: Adopt "multi-expert" to allow different streamlines for processing different RoIs for objects with particular shapes: horizontally elongated, square-like, and vertically elongated.
	Cascade R-CNN [6]	ResNet-101	~ 1000 × 600	-	-	-	42.8	**Highlights**: An extension of the RCNN, where detector stages deeper into the cascade are sequentially more selective against close positives.
	STDN [105]	DenseNet-169	513 × 513	28.6	80.9	-	31.8	**Highlights**: Propose a scale-transfer module embeded into DenseNet; The module consists of pooling and super-resolution layers with no additional parameters and computation.
Multi-layer Exploiting	HyperNet [43]	VGG-16	~ 1000 × 600	5^2	76.3	71.4	-	**Highlights**: Compress highly semantic, intermediate but really complementary, and shallow but naturally high-resolution features into a unifrom space, name Hyper Feture. Hyper Feature can be used in both region proposal and classification.

Table 2 (continued)

Method	Backbone	Input size	FPS	mAP (%)			Highlights
				VOC 2007	VOC 2012	COCO	
ION [2]	VGG-16	$\sim 1000 \times 600$	1.25	76.5	76.4	23.6	**Highligths**: Expand the Fast R-CNN to include multi-scale representation and contextual information. The contextual information is integrated using spatial recurrent neural networks.
R-SSD [41]	VGG-16	300×300	35	78.5	76.4	-	**Highlights**: Rainbow concatenation, pooling and deconvolution are performed simultaneously to create feature maps with an explicit relationship between different layers instead of increasing layers directly.
FF-SSD [7]	VGG-16	300×300	43	78.9	-	-	**Highlights**: Combine features from high layer to introduce semantic information for small objects detection.
FSSD [49]	VGG-16	300×300	65.8	78.8	82	27.1	**Highlights**: Propose a novel and lightweight way of combining feature maps from different levels and generating feature pyramid to fully utilize the features.
ESSD [104]	VGG-16	300×300	25	79.4	-	-	**Highlights**: Extend the shallow part of SSD with high-level semantic information in a multi-scale manner.

Table 2 (continued)

Method	Backbone	Input size	FPS	mAP (%)			Highlights
				VOC 2007	VOC 2012	COCO	
FPN [53]	FRCN	$\sim 1000 \times 600$	-	-	-	36.2	**Highlights**: Propose a top-down architecture with lateral connections is proposed for building high-level semantic feature maps at all scales. FPN can be easily embeded into Faster R-CNN and achieves a significant boost in accuracy without much cost in computation.
TDM [77]	ResNet-101	$\sim 1000 \times 600$	-	-	-	35.2	**Highligths**: Top-down modulations are proposed to incorporate fine details into the detection framework as a supplement to the the standard bottom-up, feedforward ConvNet, connected using lateral connections.
MDSSD [9]	VGG-16	300×300	38.5	78.6	-	26.8	**Highlights**: Propose a novel feature fusion module which consists several multi-scale deconvolution Fusion Modules to introduce semantic information for small object detection based on SSD
RON [44]	VGG-16	384×384	15	77.6	75.4	27.4	**Highlights**: Combine Faster RCNN and SSD effectively. The reverse connection assists the former layers of CNNs with more semantic information; The objectness prior gives an explicit guide to the searching of objects; The multi-task loss function enables us to optimize the whole network end-to-end on detection performance.

Table 2 (continued)

	Method	Backbone	Input size	FPS	mAP (%)			Highlights
					VOC 2007	VOC 2012	COCO	
	RefineDet [99]	VGG-16	320 × 320	40.3	80	78.1	29.4	**Highlights**: Propose a novel refinement module to get better and less anchors; Design a transfer connection block named TCB to to transfer the features.
Contextual Reasoning	MR-CNN [22]	VGG-16	∼ 1000 × 600	-	78.2	73.9	-	**Highlights**: Use features from multiple regions surrounding or inside the proposals; Encode semantic segmentation-aware features.
	Contextual [75]	VGG-16	∼ 1000 × 600	-	76.4	72.6	27.5	**Highlights**: Augment Faster R-CNN with a semantic segmentation network from ParseNet and uses it for top-down contextual priming and top-down iterative feedback using two stage training.
	DSSD [20]	ResNet-101	321 × 321	9.5	78.6	76.3	28	**Highlights**: Adopt a asymmetric hourglass network structure with deconvolutional layers and skip connection to stregthen features on the top of bottom-up structure.
	DES [102]	VGG-16	300 × 300	76.8	79.7	77.1	28.3	**Highlights**: Propose an additional segmentation branch to provide semantic information for detection.

Table 2 (continued)

	Method	Backbone	Input size	FPS	mAP (%)			Highlights
					VOC 2007	VOC 2012	COCO	
	GBD-Net [98]	BN	~ 1000 × 600	-	77.2	-	24.4	**Highlights**: Propose a Gated BiDirectional module to module therelations of multi-scale contextual regions; GBD-Net pass information among features from different context regions through convolution between neighboring support regions in two directions; Gated functions are utilized to control information transmission.
	SIN [57]	VGG-16	~ 1000 × 600	-	76	73.1	23.2	**Highlights**: Formulate object detection as a problem of graph structure inference, where given an image the objects are treated as nodes in a graph and relationships between the objects are modeled as edges in such graph.
Training Strategy	OHEM [76]	VGG-16	~ 1000 × 600	7	74.6	71.9	22.6	**Highlights**: A simple but effective online hard example mining (OHEM) algorithm to improve training of region-based detectors.
	A-Fast-RCNN [87]	Fast R-CNN	~ 1000 × 600	0.5	+2.3	+2.6	+1.4	**Highlights**: Generate hard positives through adversarial networks. It just focuses on examples with occlusions and defromations and proposes two subnets: Adversarial Spatial Dropout Network (ASDN) which learns how to occlude a given object and Adversarial Spatial Transformer Network (ASTN) which creates deformations on the object features.

Table 2 (continued)

Method	Backbone	Input size	FPS	mAP (%)			Highlights
				VOC 2007	VOC 2012	COCO	
RetinaNet [54]	ResNet-101	500 × 500	11	-	-	34.3	**Highlights**: Propose a Focal Loss which focuses training on a sparse set of hard examples.
Soft-NMS [3]	Faster R-CNN	∼ 1000 × 600	7	+1.7	-	+1.1	**Highlights**: It decays the detection scores of all other objects as a continuous function of their overlap with the bounding box with the maximum score.
Relation [36]	FRCN	∼ 1000 × 600	7	-	-	+2.7	**Highlights**: Model the relationships between objects instead of recognized individually. It can be utilized as NMS process.
MegDet [65]	ResNet-50+FPN	∼ 1000 × 600	-	-	-	52.5	**Highlights**: Allow large mini-batch training (like 256) by using cross GPU batch normalization; Train COCO within 4 hours on 128 GPUs and achieved improved accuracy; Winner of COCO2018 detection challenge.

FRCN represents for Faster R-CNN with ResNet-101 as backbone. FSSD and MDSSD are tested on Nvidia 1080 Ti, while others are Titan X

Table 3 Real-time object detection & new backbone

	Method	Backbone	Input size	FPS	mAP (%)			Highlights
					VOC 2007	VOC 2012	COCO	
Backbone	Mask R-CNN	DetNet-59 [51]	$\sim 1000 \times 600$	-	-	-	37.1	**Highlights**: First to analyze the inherent drawbacks of traditional ImageNet pre-trained model for fine-tunning recent object detectors. Propose DetNet,which is specifically designed for object detection task by maintaining the spatial resolution and enlarging the receptive field.
	FPN		$\sim 1000 \times 600$	-	-	-	40.3	
Real-time Detectors	PVANet [33]	PVANet	224×224	21.7	84.9	84.2	-	**Highlights**: Propose a novel network structure, which is an order of magnitude lighter than other state-of-the-art networks while maintaining the accuracy; Minimize its redundancy by adopting recent innovations including C.ReLU and Inception structure.
	Light-Head [48]	Xception*	1100×700	102	-	-	30.7	**Highlights**: Combine an Xception like lightweight backbone and Light-Head RCNN detection part.

Table 3 (continued)

Method	Backbone	Input size	FPS	mAP (%)			Highlights
				VOC 2007	VOC 2012	COCO	
Tiny SSD [90]	SqueezeNet	300 × 300	2.3MB	61.3	-	-	**Highlights**: Optimize SqueezeNet by reducing the number of filters, input channels and performing downsampling at a later stage in the network; Optimize sub-network stack of SSD-based convolutional feature layers.
Pelee [88]	PeleeNet	224 × 224	23.6	70.9	-	22.4	**Highlights**: Propose two-way dense layer and stem block; Use dynamic number of channels in bottleneck layer; Adopt Transition layer without compression; Perform BN before ReLU.
F-YOLO [62]	Tiny-Darknet [67]	416 × 416	200	59.4	-	-	**Highlights**: Use Tiny-darknet as backbone; First one to apply distillation on single pass detector; Transfer the soft labels from the convolutional feature maps of teacher network, more efficient. Extremely fast and fewer pareameters than VGG-based detectors.

Pelee is tested on iPhone 8, while others are Titan X (Maxwell)

methods [6, 65] still hold the state-of-the-art position of object detection on both PASCAL VOC [18, 19] and MS COCO [52] dataset while some one-stage methods [54, 70, 99, 105] catch up with a lot of two-stage approaches with almost real-time inference speed. Next, we discuss these improvements from four main perspectives mentioned above.

Architecture diagram In virtue of the powerful feature representation ability of ResNet-101 [30] or DenseNet [38], R-FCN [10], Mask R-CNN [31], CoupleNet [106], Light-Head R-CNN [48] and STDN achieve great success both in accuracy and speed view. As analyzed in [30], ResNet-101 can bring a boost of 6% compared to VGG-16 on object detection for Faster RCNN. Most importantly, the employment of position-sensitive RoI pooling or RoI Align in R-FCN and Mask R-CNN can also bring significant improvement. Cascade structure achieve almost 2× boost compared to original Faster RCNN (42.8% *vs.* 21.9%).

Multi-layer exploiting In Table 2, many methods try to exploit multi-layer features based on SSD or Faster RCNN. Obviously, multi-layer features will not only bring improvements on accuracy but also result in lost is inference speed. As modifications of SSD [56], R-SSD [41], Feature-Fused SSD [7], FSSD [49], ESSD [104], and MDSSD [9] are often compared with SSD and its another descendant DSSD [20]. These four methods gain almost the same improvement compared to conventional SSD and DSSD. Actually, these simple feature fusion methods can only achieve a slight accuracy boost. The best way to combine multi-layer features may be adopting top-down path [77] or using top-down path and feature pyramid network simultaneously [53]. Especially, FPN achieves significant improvement compared to many other methods and can be easily embedded into one-stage or two-stage approaches.

Contextual reasoning MR-CNN [22] and [75] introduce information to the detection pipeline by using a semantic segmentation subnet or information, demonstrating the effectiveness compared their baseline Faster RCNN (+8.2% for VOC2007 and +5.6% for COCO separately). Similarly, DES improves SSD with a segmentation branch, achieving +5.1% mAP on COCO. DSSD [20] applies Deconvolution modules to introduce context in the pipeline of SSD after the detection layers, getting large progress on COCO (4.8 points). Undoubtedly, DSSD drops severely on the speed running at 9.5 FPS for its ResNet-101 [30] backbone and extra deconvolutional modules. GBD-Net [98] and SIN [57] investigate the structure of CNN and add information to network from different ways though they do not perform very well on PASCAL VOC while they surpass Faster R-CNN on COCO (2.5 points and 1.3 points). From the results above, the segmentation branch helps a lot for object detection for semantic information are offered and introducing semantic information from higher layer by deconvolution operation brings both boost accuracy and lost speed.

Training strategy OHEM [76], A-Fast-RCNN [87], and Focal Loss [54] put their attention on the class imbalance problem of object detection. Both of these methods achieve progressive improvement. Soft-NMS [3] and Relation module [36] solve the problem of duplicate removal named non-maximum suppression (NMS) which is a post-processing algorithm of object detection pipeline. MegDet [65] adopts a large mini-batch and won first place in the 2017 MS COCO object detection challenge.

New backbone DetNet is designed especially for object detection as a backbone. With a DetNet-59, Mask R-CNN obtains a gain of 1.4 points (37.1% *vs.* 35.7) and FPN gets 4.1

points (40.3% *vs.* 36.2). The results demonstrate the effectiveness of the specially designed backbone for object detection, considering the maintaining resolution and enlarging the receptive field in the high layer of ResNet-50 [30].

Real-time object detection. Though a lot of object detectors mentioned above achieve a real-time speed like YOLO [68–70], SSD [56] and its descendants [7, 9, 41, 49, 104], STDN [105], RefineDet [99] and RetinaNet [54]. There are still many others models trying to achieve real-time detection by designing a lightweight backbone. These models get fast speed and comparable performance on accuracy.

6 Conclusion

In this paper, we have reviewed the most recent improvements on object detection. Specificly, backbone and famous detection networks are discussed first. Then we investigated these progresses mainly from four directions: architecture diagram, multi-layer exploiting, contextual reasoning and training strategy. Real-time object detection and ideas borrowed from other tasks like RNN and GAN are discussed included as others. Benchmarks and metrics are talked after the analysis. Finally, we simply but comprehensively analyzed the experimental results of these improvements.

Despite the success achieved in the past several years, there remain problems to be dealt with, which we see focused on the following aspects:

(1) Powerful and Efficient Features for Object Detection: One of the factors for the significant success in object detection is the powerful feature extraction capability deep CNNs, named backbone. These CNNs are pretrained on large scale image classification datasets ImageNet [14], recent literature [32] indicates that pretraining can speed up the training for object detection but it is unnecessary. Most importantly, the gap between the classification task and object detection should be further investigated for not only high-level semantic information but also more detail information that is needed for localizing objects. DetNet [51] offers a new idea by maintaining spatial resolution and enlarging the receptive field.

(2) Better and Less Anchors: Object detectors are strongly based on anchors, where RPN is utilized to generate proposals in region-based detectors and pre-defined anchors or default boxes in one-stage methods. Region proposal process can provide high-quality anchors for classification while it is very time-consuming, one the other hand, most of the predefined anchors or default boxes are negative. RefineDet [99] tries to select less default boxes by refinement neural network.

(3) Weakly Supervised or Unsupervised Learning: Current state-of-the-art object detectors adopt fully-supervised models learned from labeled data with object bounding boxes or segmentation masks [24, 31, 56, 102], however large scale data with bounding boxes or segmentation masks are very limited. Therefore, how to detect objects with less labeled data using weakly supervised or Unsupervised learning should be further studied [84, 101].

(4) Zero-shot Object Detection: Zero-shot learning method aims to slove a task without receiving any example of that task at training phase. In conventional object detection process, it is necessary to determine a certain number of object classes in order to be able to do object detection wiht high success rate. It is also necessary to ccollect as many as sample images as possible for selected object classes. Zero-short object detection [1, 13] aims to detect unknown objects which are not observed during training.

Till now, more and more methods are emerging to make object detection more accurate or faster or both on accuracy and speed. We hope this review on recent progress of object detection can make some help to researchers related to this area.

References

1. Bansal A, Sikka K, Sharma G, Chellappa R, Divakaran A (2018) Zero-shot object detection. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 384–400
2. 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, pp 2874–2883
3. Bodla N, Singh B, Chellappa R, Davis LS (2017) Soft-nms—improving object detection with one line of code. In: 2017 IEEE international conference on Computer vision (ICCV). IEEE, pp 5562–5570
4. Byeon YH, Pan SB, Moh SM, Kwak KC (2016) A surveillance system using cnn for face recognition with object, human and face detection. In: Information science and applications (ICISA) 2016. Springer, pp 975–984
5. 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. Springer, pp 354–370
6. Cai Z, Vasconcelos N (2017) Cascade r-cnn: Delving into high quality object detection. arXiv:1712.00726
7. 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 2017). International Society for Optics and Photonics, vol 10615, pp 106151e
8. Chollet F (2017) Xception: Deep learning with depthwise separable convolutions. Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, pp 1800–1807. https://doi.org/10.1109/CVPR.2017.195
9. Cui L (2018) Mdssd: Multi-scale deconvolutional single shot detector for small objects. arXiv:1805.07009
10. 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, pp 379–387
11. Dai J, Qi H, Xiong Y, Li Y, Zhang G, Hu H, Wei Y (2017) Deformable Convolutional Networks, pp 764–773
12. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: 2005. CVPR 2005. IEEE computer society conference on Computer vision and pattern recognition, vol 1. IEEE, pp 886–893
13. Demirel B, Cinbis RG, Ikizler-Cinbis N (2018) Zero-shot object detection by hybrid region embedding. arXiv:1805.06157
14. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-fei L (2009) Imagenet: A large-scale hierarchical image database. In: 2009. CVPR 2009. IEEE conference on Computer vision and pattern recognition. IEEE, pp 248–255
15. Dollár P, Wojek C, Schiele B, Perona P (2009) Pedestrian detection: a benchmark. In: 2009. CVPR 2009. IEEE conference on Computer vision and pattern recognition. IEEE, pp 304–311
16. Dollar P, Wojek C, Schiele B, Perona P (2012) Pedestrian detection: an evaluation of the state of the art. IEEE Trans Pattern Anal Mach Intell 34(4):743–761
17. 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, pp 2147–2154
18. Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A (2008) The pascal visual object classes challenge 2007 (voc 2007) results (2007). http://host.robots.ox.ac.uk/pascal/VOC/voc2007/workshop/index.html
19. Everingham M, Van gool L, Williams CKI, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. Int J Comput Vis 88(2):303–338
20. Fu C, Liu W, Ranga A, Tyagi A, Berg AC (2017) Dssd: Deconvolutional single shot detector. arXiv:1701.06659
21. Geiger A, Lenz P, Urtasun R (2012) Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on computer vision and pattern recognition (CVPR)

22. Gidaris S, Komodakis N (2015) Object detection via a multi-region and semantic segmentation-aware cnn model. In: Proceedings of the IEEE International Conference on Computer Vision, pp 1134–1142
23. Girshick R, Felzenszwalb PF, Mcallester D (2012) Discriminatively trained deformable part models release 5
24. 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, pp 580–587
25. Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448
26. Goodfellow I, 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, pp 2672–2680
27. Goyal P, Dollár P, Girshick R, Noordhuis P, Wesolowski L, Kyrola A, Tulloch A, Jia Y, He K (2017) Accurate, large minibatch sgd: training imagenet in 1 hour. arXiv:1706.02677
28. Han J, Zhang D, Cheng G, Liu N, Xu D (2018) Advanced deep-learning techniques for salient and category-specific object detection: a survey. IEEE Signal Proc Mag 35(1):84–100
29. He K, Zhang X, Ren S, Sun J (2014) Spatial pyramid pooling in deep convolutional networks for visual recognition. In: European conference on computer vision. Springer, pp 346–361
30. He K, Zhang X, Ren S, Sun J (2016) Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp 770–778. https://doi.org/10.1109/CVPR.2016.90
31. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In: 2017 IEEE international conference on Computer vision (ICCV). IEEE, pp 2980–2988
32. He K, Girshick R, Dollár P (2018) Rethinking imagenet pre-training. arXiv:1811.08883
33. Hong S, Roh B, Kim KH, Cheon Y, Park M (2016) Pvanet: lightweight deep neural networks for real-time object detection. arXiv:1611.08588
34. Hosang J, Benenson R, Schiele B (2017) Learning non-maximum suppression. In: The IEEE conference on computer vision and pattern recognition (CVPR), vol 2
35. Howard A, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv:1704.04861
36. Hu H, Gu J, Zhang Z, Dai J, Wei Y (2017) Relation networks for object detection. arXiv:1711.11575.8
37. Hu J, Shen L, Sun G (2017) Squeeze-and-Excitation Networks. arXiv:1709.01507, pp 1–11. https://doi.org/10.1109/CVPR.2018.00745
38. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, pp 2261–2269. https://doi.org/10.1109/CVPR.2017.243
39. Huang J, Rathod V, Sun C, Zhu M, Korattikara A, Fathi A, Fischer I, Wojna Z, Song Y, Guadarrama S et al (2017) Speed/accuracy trade-offs for modern convolutional object detectors. In: IEEE CVPR, vol 4
40. Iandola F, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K (2016) SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <,0.5MB model size. arXiv:1602.07360, pp 1–13. https://doi.org/10.1007/978-3-319-24553-9
41. Jeong J, Park H, Kwak N (2017) Enhancement of SSD by concatenating feature maps for object detection. arXiv:1705.09587
42. Jian M, Qi Q, Dong J, Sun X, Sun Y, Lam KM (2018) Saliency detection using quaternionic distance based weber local descriptor and level priors. Multimedia tools and applications, pp 1–18
43. 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, pp 845–853
44. Kong T, Sun F, Yao A, Liu H, Lu M, Chen Y (2017) Ron: Reverse connection with objectness prior networks for object detection. In: IEEE Conference on computer vision and pattern recognition, vol 1, pp 2
45. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet Classification with Deep Convolutional Neural Networks. Advances In Neural Information Processing Systems, pp 1–9. https://doi.org/10.1016/j.protcy.2014.09.007
46. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
47. Lee H, Eum S, Kwon H (2017) Me r-cnn: Multi-expert r-cnn for object detection. arXiv:1704.01069

48. Li Z, Peng C, Yu G, Zhang X, Deng Y, Sun J (2017) Light-head r-cnn: In defense of two-stage object detector. arXiv:1711.07264
49. Li Z, Zhou F (2017) Fssd: Feature fusion single shot multibox detector. arXiv:1712.00960
50. Li J, Liang X, Li J, Wei Y, Xu T, Feng J, Yan S (2018) Multistage Object Detection With Group Recursive Learning. IEEE Trans Multimed 20(7):1645–1655
51. Li Z, Peng C, Yu G, Zhang X, Deng Y, Sun J (2018) DetNet: A Backbone network for Object Detection. arXiv:1804.06215, 1(2), 3
52. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: Common objects in context. In: European conference on computer vision. Springer, pp 740–755
53. Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: CVPR, vol 1, pp 4
54. Lin TY, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. arXiv:1708.02002
55. Liu Z, Luo P, Wang X, Tang X (2015) Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (ICCV)
56. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C, Berg AC (2016) Ssd: Single shot multibox detector. In: European conference on computer vision. Springer, pp 21–37
57. 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, pp 6985–6994
58. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3431–3440
59. Lowe DG (2004) Distinctive image features from scale invariant keypoints. Int J Comput Vis 60:91–11020,042. https://doi.org/10.1023/B:VISI.0000029664.99615.94
60. Lu J, Sibai H, Fabry E (2017) Adversarial examples that fool detectors. arXiv:1712.02494
61. Ma N, Zhang X, Zheng HT, Sun J (2018) ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. arXiv:1807.11164, pp 1–19
62. Mehta R, Ozturk C (2018) Object detection at 200 frames per second. arXiv:1805.06361
63. Oksuz K, Cam BC, Akbas E, Kalkan S (2018) Localization recall precision (lrp): A new performance metric for object detection. arXiv:1807.01696
64. Ouyang W, Wang X, Zeng X, Qiu S, Luo P, Tian Y, Li H, Yang S, Wang Z, Loy CC et al (2015) Deepid-net: Deformable deep convolutional neural networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 2403–2412
65. 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, pp 6181–6189
66. QiongYan J, LiXu Y (2017) Accurate single stage detector using recurrent rolling convolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition
67. Redmon J (2013) Darknet: Open source neural networks in c. http://pjreddie.com/darknet 2016
68. 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, pp 779–788
69. Redmon J, Farhadi A (2017) Yolo9000: better, faster, stronger arXiv preprint
70. Redmon J, Farhadi A (2018) Yolov3: An incremental improvement. arXiv:1804.02767
71. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp 91–99
72. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L (2018) Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. arXiv:1801.04381
73. Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, LeCun Y (2013) Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv:1312.6229
74. Shen Z, Liu Z, Li J, Jiang YG, Chen Y, Xue X (2017) Dsod: Learning deeply supervised object detectors from scratch. In: The IEEE international conference on computer vision (ICCV), vol 3, pp 7
75. Shrivastava A, Gupta A (2016) Contextual priming and feedback for faster r-cnn. In: European conference on computer vision. Springer, pp 330–348
76. 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, pp 761–769
77. Shrivastava A, Sukthankar R, Malik J, Gupta A (2016) Beyond skip connections: Top-down modulation for object detection. arXiv:1612.06851

78. Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, Hubert T, Baker L, Lai M, Bolton A et al (2017) Mastering the game of go without human knowledge. Nature 550(7676):354
79. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
80. Singh B, Li H, Sharma A, Davis LS (2018) R-fcn-3000 at 30fps: Decoupling detection and classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1081–1090
81. Srivastava RK, Greff K, Schmidhuber J (2015) Highway networks. arXiv:1505.00387
82. Szegedy C, Liu W, Jia Y, 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, pp 1–9
83. 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, pp 2818–2826
84. Tang Y, Wang J, Wang X, Gao B, Dellandréa E, Gaizauskas R, Chen L (2017) Visual and semantic knowledge transfer for large scale semi-supervised object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence
85. Uijlings JR, Van De Sande KE, Gevers T, Smeulders AW (2013) Selective search for object recognition. Int J Comput Vis 104(2):154–171. https://doi.org/10.1007/s11263-013-0620-5
86. Wang J, Fu W, Liu J, Lu H et al (2014) Spatiotemporal group context for pedestrian counting. IEEE Trans Circ Syst Video Technol 24(9):1620–1630
87. Wang X, Shrivastava A, Gupta A (2017) A-fast-rcnn: Hard positive generation via adversary for object detection. In: IEEE Conference on computer vision and pattern recognition
88. Wang RJ, Li X, Ao S, Ling CX (2018) Pelee: A real-time object detection system on mobile devices. arXiv:1804.06882
89. Williams RJ, Zipser D (1989) A learning algorithm for continually running fully recurrent neural networks. Neural Comput 1(2):270–280
90. Wong A, Shafiee MJ, Li F, Chwyl B (2018) Tiny ssd: A tiny single-shot detection deep convolutional neural network for real-time embedded object detection. arXiv:1802.06488
91. Wu B, Iandola F, Jin PH, Keutzer K (2016) Squeezedet: unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In: IEEE Conference on computer vision and pattern recognition workshops, pp 446–454
92. Xie C, Wang J, Zhang Z, Zhou Y, Xie L, Yuille A (2017) Adversarial examples for semantic segmentation and object detection. In: Proceedings of International Conference on Computer Vision (ICCV), pp 1378–1387
93. Xie S, Girshick R, Dollár P, Tu Z, He K (2017) Aggregated residual transformations for deep neural networks. Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, pp 5987–5995. https://doi.org/10.1109/CVPR.2017.634
94. Yang B, Yan J, Lei Z, Li SZ (2016) Craft objects from images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 6043–6051
95. Yang F, Choi W, Lin Y (2016) Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2129–2137
96. You Y, Zhang Z, Hsieh C, Demmel J, Keutzer K Imagenet training in minutes
97. Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: European conference on computer vision. Springer, pp 818–833
98. Zeng X, Ouyang W, Yang B, Yan J, Wang X (2016) Gated bi-directional cnn for object detection. In: European conference on computer vision. Springer, pp 354–369
99. Zhang S, Wen L, Bian X, Lei Z, Li SZ (2017) Single-shot refinement neural network for object detection. arXiv preprint
100. Zhang X, Zhou X, Lin M, Sun J (2017) ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. arXiv:1707.01083. https://doi.org/10.1109/CVPR.2018.00716
101. Zhang Y, Bai Y, Ding M, Li Y, Ghanem B (2018) W2f: a weakly-supervised to fully-supervised framework for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 928–936
102. Zhang Z, Qiao S, Xie C, Shen W, Wang B, Yuille AL (2018) Single-shot object detection with enriched semantics. Technical report, Center for Brains, Minds and Machines (CBMM)
103. Zhang Z, He T, Zhang H, Zhang Z, Xie J, Li M (2019) Bag of freebies for training object detection neural networks. arXiv:1902.04103

104. Zheng L, Fu C, Zhao Y (2018) Extend the shallow part of single shot multibox detector via convolutional neural network. arXiv:1801.05918
105. 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, pp 528–537
106. Zhu Y, Zhao C, Wang J, Zhao X, Wu Y, Lu H et al (2017) Couplenet: Coupling global structure with local parts for object detection. In: Proceedings of international conference on computer vision (ICCV), vol 2
107. Zitnick CL, Dollàr P (2014) Edge boxes: Locating object proposals from edges. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 8693 LNCS(PART 5), 391–405. https://doi.org/10.1007/978-3-319-10602-1_26

Hao Zhang received the B.S. degree from Nanchang University, Nanchang, China, in 2017. He is currently pursuing the M.S. degree with the School of Information Engineering, Nanchang University. His research interests focus on deep learning/machine learning, and its application in computer vision.

Xianggong Hong received the M.S. degree in Electronic and Communication Engineering from Nanchang University, Nanchang, China, in 2009, where he is currently a Professor with the Cognition Sensor Network Laboratory, School of Information Engineering. His research interests focus on digital image processing and digital voice exchanging.