IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Mathematical Modelling of Engineering Problems
Vol. 10, No. 5, October, 2023, pp. 1587-1610
Journal homepage: http://iieta.org/journals/mmep

# An Examination of Advances in Multistage Object Detection Techniques Utilizing Deep Learning

Thanh Quyen Ngo, Nguyen Duc Toan, Long Ho Le, Trung Dung Nguyen, Hoanh Nguyen*

Faculty of Electrical Engineering Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh City 700000, Vietnam

Corresponding Author Email: nguyenhoanh@iuh.edu.vn

https://doi.org/10.18280/mmep.100510

**Received:** 23 June 2023
**Revised:** 16 August 2023
**Accepted:** 10 September 2023
**Available online:** 27 October 2023

***Keywords:***
*anchor box-based object detection methods, deep learning, multi-stage object detection methods, object detection, point-based object detection methods*

**ABSTRACT**

Techniques for object detection rooted in deep learning can be broadly segregated into two major categories: single-stage and multi-stage architectures. Notably, multi-stage object detection methods often deliver superior performance due to their intricate structure. However, they demand careful scrutiny during both their design and training phases. This manuscript offers a thorough review of the latest progress in the realm of multi-stage object detection, with the objective of fostering a comprehensive understanding of contemporary designs from a network architecture viewpoint. To facilitate this, the structure of the multi-stage object detection framework is divided into distinct modules, each reflective of a specific learning process stage. Each module is addressed in a systematic manner, beginning with an in-depth exploration of initial structural designs and proceeding to discuss optimization solutions drawn from recent scholarly contributions. A summarization of the performance of reviewed strategies within each module is provided, thereby offering a clear overview of current methodologies. Additionally, significant unresolved challenges in each module are identified, highlighting potential areas of investigation for future research.

## 1. INTRODUCTION

### 1.1 Background

Object detection methodologies, fundamental in a myriad of practical applications such as autonomous driving, video surveillance, and medical image analysis, rely heavily on precise object detection. The intricate nature of these methods facilitates higher accuracy, enabling objects to be identified and localized even within demanding and fluctuating environments. The progress within the field of object detection, therefore, holds transformative potential for the broader computer vision community, allowing for the development of more robust, precise, and efficient systems. This has far-reaching implications, enhancing the dependability and applicability of computer vision in everyday life.

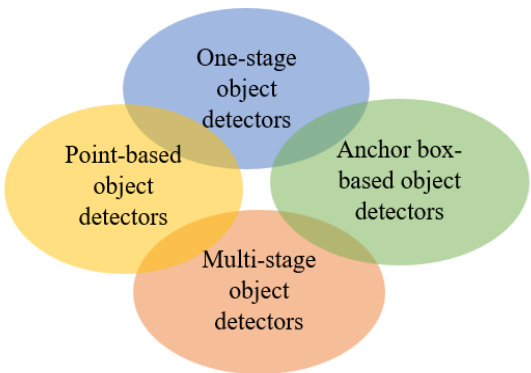



**Figure 1.** Different groups of deep learning object detectors

Typically, an object detection framework encompasses two primary tasks: the classification task, which predicts the class labels of objects, and the localization task, which predicts the objects' locations within an image. The latter is often more challenging due to factors such as scale variation, occlusion, and viewpoint. Initial object detection methodologies were predominantly based on hand-crafted features and a linear classifier for object prediction within an image [1]. However, with the advent of deep convolutional neural networks (CNN) and its widespread applications in many fields [2, 3], object detection techniques rooted in deep learning have garnered substantial attention from the academic community.

As illustrated in Figure 1, deep learning-based object detection methodologies can be chiefly divided into four groups. Depending on the approach to refining object locations within an image, deep learning object detectors can be bifurcated into multi-stage and one-stage object detection techniques. The former first generates a series of proposal boxes indicative of object instances using a proposal generation network in the initial stage. These proposal features are then extracted and fed into subsequent networks for repeated object location refinement before final predictions are rendered. Conversely, one-stage object detection frameworks directly produce final predictions, bypassing object proposal generation. Given that multi-stage object detectors generate a sparse set of proposal boxes in the initial stage, they do not encounter the problem of class imbalance as one-stage object detectors do. Moreover, by refining object locations multiple times, the final bounding boxes generated by multi-stage object detectors are significantly improved compared to the predicted boxes generated by one-stage object detectors. However, multi-stage object detection frameworks typically

possess a more complex structure than their one-stage counterparts, necessitating careful design to achieve optimal detection performance. Various benchmarks suggest that while multi-stage detection methods deliver superior detection accuracy, one-stage detection frameworks are characterized by simpler architecture and faster processing speed [4, 5]. It is noteworthy that two-stage object detection methods, due to their efficiency, have received most attention among multi-stage object detection techniques. This paper predominantly follows the design of two-stage object detection.

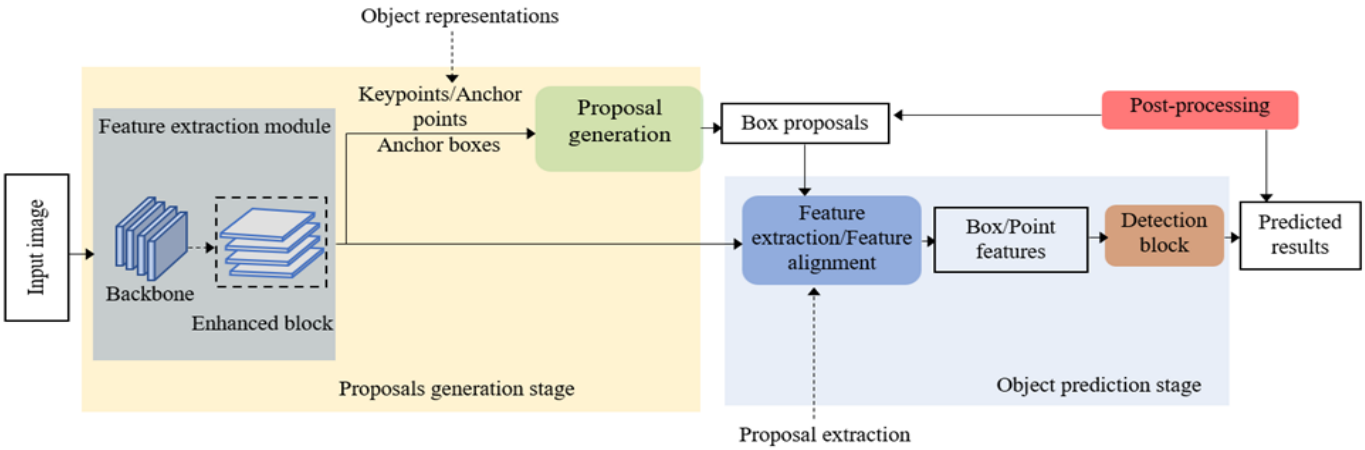



**Figure 2.** The structure of two-stage object detection framework

Deep learning object detection methodologies can be further segregated into two primary categories based on how they represent objects within an image: anchor box-based object detection methods and point-based object detection methods. Anchor box-based methods depict objects as rectangular bounding boxes on a feature map, while point-based methods utilize keypoints or anchor points on a feature map to signify objects. Anchor box-based object detection methods, owing to the convenience of processing rectangular boxes, are more straightforward to implement when compared to point-based methods. However, they yield merely coarse localization of objects, potentially influenced by background details and foreground facets with minimal semantic information. In the early stages of deep learning object detectors, rectangular bounding box representation schemes were predominantly employed to produce object proposals. With the recent introduction of the FPN architecture and Focal Loss, point-based object detection methods have garnered substantial attention, resulting in significant advancements [6]. It is crucial to note that while point-based schemes are generally employed in one-stage object detection pipelines, they can be modified to suit multi-stage object detection pipelines.

Numerous techniques have been proposed in recent years to augment the detection performance for specific types of object detection frameworks, such as one-stage framework, two-stage framework, anchor box-based framework, or point-based framework. For instance, RetinaNet proposed a novel Focal Loss function to address the class imbalance issue prevalent in one-stage object detectors [6]. Concurrently, ThunderNet introduced an innovative lightweight structure to enhance the inference speed of two-stage object detectors [7]. While some original techniques were designed to be utilized in diverse types of object detection pipelines, many others were introduced based on a particular type of object detection pipeline and required modification to be integrated into other types of object detection pipelines. Consequently, a technique proposed in a one-stage object detector could be employed in a two-stage object detector to improve its detection performance. Given the more complex structure of multi-stage object detection frameworks compared to one-stage frameworks, meticulous design is imperative to achieve optimal detection accuracy. It is hypothesized that a multi-stage object detector, with well-designed components, could achieve state-of-the-art detection results.

This paper aims to systematically review recent techniques that can be incorporated into a deep learning multi-stage object detection framework, particularly a two-stage object detection framework. The techniques reviewed in this paper are not limited to those originally introduced in multi-stage object detectors but also include those that can be modified to be applied in multi-stage object detectors. The structure of a multi-stage object detection framework is divided into different modules, including feature extraction, object representations, proposal generation, proposal extraction, detection block, post-processing, feature selection, and sampling strategy, as illustrated in Figure 2. The initial structural designs are discussed in depth, followed by a review of recent optimization techniques proposed to boost the performance of each module. Furthermore, a performance summary of the reviewed techniques is provided to examine their strengths and weaknesses. Major unresolved issues in each module are also discussed, suggesting potential avenues for future research.

### 1.2 Scope of the survey and contributions

Covering all strategies proposed for the design of an effective object detection framework is beyond the scope of this paper, given the expansive nature of deep learning object detection within the domains of computer science and machine learning. Acknowledging the dominance of multi-stage object detection methods, as evidenced by their superior detection accuracy on standard benchmarks compared to one-stage methods, the focus of this review is primarily on recent techniques that can be employed to enhance a multi-stage object detection framework. The review predominantly considers techniques from the past five years, but also includes

some earlier works to facilitate a comprehensive understanding of the subject matter. Regrettably, due to these constraints, not all works could be encompassed within this review, and apologies are extended to authors whose works are not included. The hope is that this survey will shed light on potential future research directions in the design of object detection frameworks.

This paper contributes to the field in several ways:

It provides a comprehensive overview of the architecture of deep learning multi-stage object detection, dissecting the architecture into distinct modules according to the learning process, and discussing the early structural designs of each module.

It systematically reviews recent optimization techniques proposed to refine the architecture of deep learning object detectors, aiming for enhanced detection performance. The techniques explored in this paper can be leveraged in the design of a multi-stage object detection framework.

Each technique is thoroughly examined, discussing its structure in detail to help readers gain a profound understanding of the key features of existing strategies.

A performance summary is provided for the reviewed techniques in each module. By comparing the performance of various strategies, researchers can discern the strengths and weaknesses of current methods.

Lastly, significant open issues and challenges in each module are discussed, offering insights into potential research directions.

### 1.3 Comparison with previous reviews and surveys

Some deep learning object detection surveys and reviews have been published in past years. These include surveys on the problem of generic object detection [8-10] and category-specific object detection, such as pedestrian detection [11], face detection [12], and text detection [13]. While these previous surveys provide a general overview of each model, this paper gives specific attention to recent techniques that have been proposed to improve the architecture of each module in a deep learning generic object detection model. In particular, we comprehensively present and discuss existing techniques from a network architecture point-of-view. The reviewed techniques are grouped into different sections based on the structure of a multi-stage object detection framework. All methods reviewed in this paper can be exploited to design an efficient multi-stage object detector. We hope that this survey will provide novel insights and inspirations that guide future research directions, especially for designing multi-stage object detection frameworks.

## 2. DEEP CNN MULTI-STAGE OBJECT DETECTION

State-of-the-art object detectors mainly follow a multi-stage object detection pipeline. In a multi-stage object detection pipeline, proposal generation stage and object prediction stage are iterated multiple times to produce high-quality predicted results. Among multi-stage object detection methods, two-stage object detection methods, which consist of one proposal generation stage and one object prediction stage, have attracted most attention because of the speed-accuracy trade-off. In this paper, we mainly follow the two-stage object detection design. Figure 2 illustrates the structure of a two-stage object detection framework. Based on input images, the proposal generation stage generates a set of proposal boxes which represent object scales and locations in an image. The object prediction stage adopts feature maps and proposal boxes generated by the proposal generation stage as inputs and produces final predictions. A multi-stage object detection method with more than two stages usually iterates the proposal generation stage [14, 15] or the object prediction stage [16, 17] to produce better predicted results. Based on the structure of the two-stage object detection framework, we divide each stage into different modules as shown in Table 1 and review recent optimization techniques that proposed to improve the performance of each module. In addition, other problems that occur in both stages are also discussed.

**Table 1.** Categorization of the reviewed strategies for each module

| Module | Techniques | Publications |
|---|---|---|
| Feature Extraction | Detection based on multi-layer backbone | MS-CNN, Scale-aware Fast R-CNN, Exploit-All-the-Layers |
| | Detection based on feature pyramid | FPN, PANet, Libra R-CNN, AugFPN, BiFPN, A2-FPN, Recursive Feature Pyramid |
| Object Representations | Rectangular bounding box representations | ThunderNet, FPN, Faster R-CNN, Metaanchor, Anchor Box Optimization, Sparse R-CNN |
| | Point representations | DenseBox, FSAF, Guided Anchoring, FCOS, FoveaBox, SPAD, CornerNet, CenterNet, ExtremeNet, Objects as Points, RepPoints, VarifocalNet |
| Proposal Generation | Region proposal networks | RPN |
| | Cascade RPN | Cascade RPN, Iterative RPN |
| | Extended RPN | BorderRPN |
| | Based on point representations | GA-RPN, CPN, SC-RPN |
| Proposal Extraction | RoI Pooling | Faster R-CNN |
| | RoI Align | RoI Align, Grid R-CNN |
| | Deformable RoI Pooling | Deformable RoI Pooling |
| | Discriminative RoI Pooling | D2Det |
| Detection Block | Based on localization sensitive scores | Cascade R-CNN, IoU-Net, Learning-to-Rank |
| | Based on task-specific structure | D2Det, Grid R-CNN, TSD, Double-Head |
| Post-processing | NMS-based | IoU-guided NMS, Soft-NMS, Adaptive-NMS |
| | Learning-based | Learning NMS, Relation Networks |
| | Pooling-based | MaxpoolNMS |
| Sampling strategy | Hard sampling | Libra R-CNN, SC-RPN |
| | Soft sampling | Prime Sample Attention, Adaptive Training Sample Selection |

Note: A work may appear at multiple locations if it proposes optimization techniques in different modules.

### 2.1 Overview

In this section, we aim to present a comprehensive analysis of the core components integral to the design of multi-stage object detection methods. We will delve into the specifics of each component, exploring both traditional and contemporary practices and their impacts. Below, we provide a brief outline of the major subsections that will be covered:

Feature extraction: We'll commence with a review of how features are extracted from raw data, elaborating on the importance of these extracted features in enhancing the distinguishing characteristics of various objects.

Object representations: Next, we'll discuss the different techniques utilized to internally represent objects, setting the foundation for how objects are differentiated within the system.

Proposal generation: This subsection will focus on the methodologies used to generate a series of proposals or regions where potential objects of interest might exist.

Proposal extraction: Here, we will explore how relevant proposals are extracted from the generated regions, focusing on those with a higher likelihood of containing the objects of interest.

Detection block: This subsection will dissect the structure of the detection block that refines the spatial locations of the objects and classifies the proposed regions into distinct object classes.

Post-processing: We will review the strategies employed in the final stages of object detection to remove redundant detections and provide a more accurate and clean output.

Sampling strategy: Lastly, we'll discuss the sampling strategies used to maintain a balance between positive and negative samples in the training data, a critical factor in ensuring the efficacy of the object detection method.

In the following, we briefly elaborate on the architecture of each module and introduce optimization strategies that have been proposed in recent years.

### 2.2 Feature extraction

Feature extraction module extracts features from input images. Early deep CNN object detectors usually adopt a backbone CNN model (e.g., VGG [18], ResNets [19]) to generate feature maps and use a single feature layer for proposal generation and object prediction tasks. To address the problem of object scale variation, various methods exploit different feature maps at different layers of the backbone for generating proposals and predicting objects [20-22]. With the advent of FPN (Feature Pyramid Network) [23], recent techniques in feature extraction focus on generating a high-level semantic feature pyramid [24-29]. These techniques combine feature maps at different layers of the backbone by an enhanced or extended block. By propagating the strong semantic features from deeper layers at lower layers, the detection performance of proposal generation and object prediction have been substantially improved. In this paper, feature extraction module includes the backbone network and enhanced/extended modules.

### 2.3 Object representations

Since an object can be located at any shape, scale, and position in an image, and the appearance of objects of the same class can be very different in different images, an object representation strategy is required for generating initial guesses of objects in image. Classical object detection methods are usually based on the sliding-window strategy, in which a classic classifier is applied on a dense image grid. Early deep learning-based object detectors mostly adopt rectangular bounding box representation, which places a number of rectangular bounding boxes on feature maps, to enumerate possible shapes, scales, and positions for target objects [7, 23, 30-33]. Recently, state-of-the-art object detectors have relied on point representations to represent objects. Point representations-based methods directly use points on feature maps to depict objects and form bounding boxes based on proposal points [34-45].

### 2.4 Proposal generation

In multi-stage object detection frameworks, proposal generation network produces a set of high-quality object proposals based on input feature maps and object representations. These sparse object proposals are then used by the second stage for producing final predictions. Classical object detection methods are based on merging super-pixels [46] or sliding windows [47] to extract high-quality proposal boxes. With the success of CNN, Ren et al. [30] introduced novel region proposal network (RPN) to produce high-quality proposals in a fully convolutional way. RPN has been used by many modern object detectors as a proposal generation network. Recently, various methods have been proposed to improve the performance of RPN. Based on RPN, Cascade RPN [15] and Iterative RPN [48] proposed a cascade structure which performs proposal refinement several times while BorderRPN [49] employs point features as an extended module to enhance proposal generation network. Another approach is based on point representations to generate proposals [36, 50, 51].

### 2.5 Proposal extraction

In multi-stage object detection frameworks, proposal extraction aims to extract fixed-sized proposal feature maps based on proposals generated by the proposal generation network. Most early two-stage object detectors employ RoI (Region of Interest) pooling scheme [52] to extract high-quality proposal features. To alleviate the misalignment problem caused by quantization process in RoI pooling scheme, RoI Align [53] proposed to use bilinear interpolation in each sub-region of input feature map to generate proposal features. Recently, Deformable RoI Pooling [54] and Discriminative RoI Pooling [55] have been designed based on deformable convolution to add nearby semantic information to output proposal features, thus improving the localization capability of the detection network.

### 2.6 Detection block

The detection block aims to produce final predictions based on proposal features generated by proposal extraction module. Many two-stage object detection frameworks adopt region-based convolutional neural network (R-CNN) [52] as the detection head. Later, R-FCN (Region-based Fully Convolutional Networks) [56] employs fully convolutional layers to replace fully connected layers in R-CNN to improve the efficiency of detection network. Both R-CNN and R-FCN share a head for both classification and bounding box regression. This leads to misalignment problems between the

two branches, which in turn limits the detection performance of the detection head. Recently, IoU-Net [57] and Learning-to-Rank [58] have been proposed to improve classification scores for proposals with high localization accuracy to prevents them from being suppressed while Cascade R-CNN [26] applied R-CNN several times with increasing IoU thresholds to produce high-quality detection results. On the other hand, Grid R-CNN [55], D2Det [59], TSD [60], and Double-Head [61] proposed to design different structures for different tasks in the detection head.

### 2.7 Post-processing

Since anchor or proposal boxes usually overlap with each other, the proposal generation and detection network produce many duplicate results. Therefore, a method is needed to remove these duplicate boxes. Most deep object detectors employ GreedyNMS (Greedy non-maximum suppression) as duplicate removal method. Recently, Soft-NMS [62], IoU-guided NMS [57], and Adaptive-NMS [63] have been designed based on modifying GreedyNMS algorithm to overcome the inherent drawback of GreedyNMS. Different from NMS-based methods, MaxpoolNMS [64] applies max pooling operations to extract peak locations in the objectness score map to obtain a meaningful set of object proposals. Another line of research is to design a learnable deep network to replace GreedyNMS so that the model can be trained fully end-to-end [65-67].

### 2.8 Sampling strategy

Most early multi-stage object detection approaches employ random sampling schemes to generate training samples. In random sampling scheme, positive and negative samples are randomly sampled. As a result, training samples generated by this scheme are easy to be dominated by easy samples. Recent strategies for producing training samples focus on hard samples [25] or prime samples [68]. These sampling techniques exploit IoU values or classification scores to define positive and negative samples, thus focusing the training process on hard samples. Alternatively, ATSS [69] defines positive and negative samples based on mean and standard deviation of IoU values between candidate samples and ground truth. Recently, SC-RPN [51] has designed a size-aware dynamic sampling method to ensure the sampling consistency in terms of location, size and quantity between the two stages of a two-stage object detection framework.

## 3. PROPOSAL GENERATION STAGE

The proposal generation stage aims to produce a set of high-quality proposal boxes to represent object scales and locations from an input image. In this stage, an input image is fed into a deep backbone network to produce feature maps through different layers. Based on feature maps generated by the backbone, a proposal generation network is employed to generate a set of proposal boxes. Early deep CNN object detection methods usually exploit a single feature map generated by the backbone for producing proposal boxes. Recently, various techniques have been proposed to enhance semantic information of the backbone feature maps. In this paper, the feature extraction module includes the backbone network and extended modules. As an object can appear at any position and scale in an image, object representation methods are designed to produce initial guesses of object locations and shapes. Anchor box-based methods, which use a rectangular bounding box to represent an object, have led the fashion in the past few years. Recently, object detection methods based on point representations have emerged and attracted the attention of the research community due to their efficiency. As the final module in the proposal generation stage, proposal generation network uses object representations and input feature maps to generate high-quality proposal boxes. Current leading methods for proposal generation network are usually based on improving the region proposal network. In the following, we review recent optimization strategies in each module of the proposal generation stage.

### 3.1 Recent techniques in feature extraction

Deep learning-based object detection methods usually adopt a deep convolutional neural network pretrained on an image classification task as the backbone network to extract features from input images [30. 52]. In recent years, various deep architectures have been designed to improve classification performance. AlexNet [70] proposed an eight layers network with learnable parameters. ReLU activation and dropout layers are also adopted in the network to reduce the computational cost and prevent overfitting. VGG [18] proposed to use small convolution filters (i.e., 3×3 convolution filters) to increase the depth of the network. ResNets [19] introduced residual block which consists of a series of layers and a shortcut connection adding the input and output of the block. This design is very efficient to build a deeper network. ResNets is now still one of the most widely used backbone architectures for object detection models. Recently, various studies have focused on designing a light-weight deep architecture based on pointwise and depthwise convolutions for reducing computational complexity [71-73]. Li et al. [74] proved that using pretrained models designed for image classification tasks is not suitable for object detection tasks since object detection tasks not only need to classify the objects but also exactly localize the boundaries of the objects in image. The authors also proposed a deep network specially designed for object detection with high resolution feature maps to locate multi-scale objects. In general, object detection methods based on a single-level feature map (Figure 3) have a decline in detection accuracy due to scale variation of objects in natural scene images [23]. Moreover, the feature maps generated by the backbone usually contain high-level semantic information in deep layers and low-level semantic information in shallow layers. As a result, object detection methods usually apply the detection network on deep feature layers. However, due to low resolution in deep layers, the structure of objects may be destroyed, especially for small objects. This may compromise the detection performance of the detection network. In recent years, various methods have been proposed to improve the detection performance based on feature maps generated by the backbone network. According to the way of exploiting the backbone feature maps for detecting objects, these methods can be roughly divided into two groups: methods based on multi-layer backbone and methods based on feature pyramid. Detection methods based on multi-layer backbone adopt different feature maps at different layers of the backbone to predict objects while detection methods based on feature pyramid construct a high-level semantic feature pyramid to produce final predictions. It should be noted that since we

focus on the two-stage object detection design, some techniques specially proposed for one-stage detection framework, such as deconvolution module [75], are not reviewed in this paper.

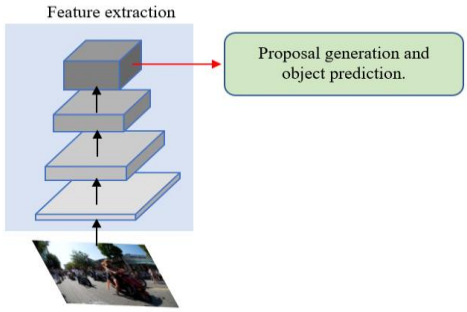



**Figure 3.** Object detection based on a single-level feature map [30, 52]

3.1.1 Detection methods based on multi-layer backbone

Multi-layer object detection methods propose to directly predict objects from different feature maps at different layers of the backbone network (Figure 4) [20-22]. These methods exploit features at different layers independently to overcome the scale variation problem since different feature layers encode information of objects at different scales. Multi-layer backbone object detection methods have received less attention in recent years. Most multi-layer backbone object detectors focus on producing high-level semantic feature maps by an optimization backbone network. MS-CNN [20] is the first two-stage object detection framework that applies the proposal generation network at different output layers of the backbone network. To bridge the scale gap between input image and convolutional layers, deconvolution layers are employed to increase the resolution of feature maps. However, RoI pooling scheme is still applied on a single feature map. This leads to misalignment between proposal boxes and object features, which compromises the detection performance. Li et al. [21] designed a scale-aware network which incorporates two different subnetworks into a unified architecture to deal with the scale variation of objects in image. Each subnetwork first uses convolution layers to further extract scale-specific feature maps based on feature maps generated by the backbone and then generates scale-specific detection results. Yang et al. [22] proposed a new approach to perform multi-scale object detection which first examines the size of each object proposal and then pools the features from a corresponding feature map. Three different layers in the VGG16 backbone are adopted to pool object features. To ensure the features are discriminative enough, a scale-dependent pooling scheme was introduced to provides strong supervision to enforce more discriminative convolutional filters.

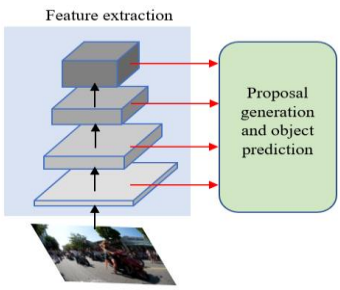
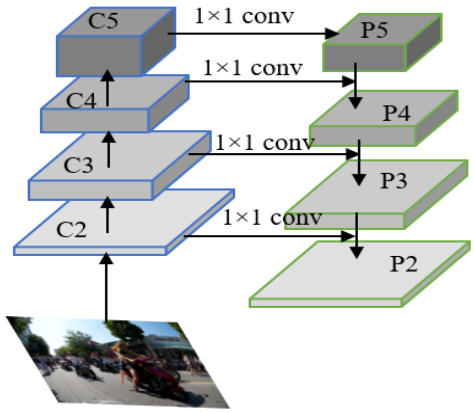



**Figure 4.** Object detection based on multi-layer features [20-22]

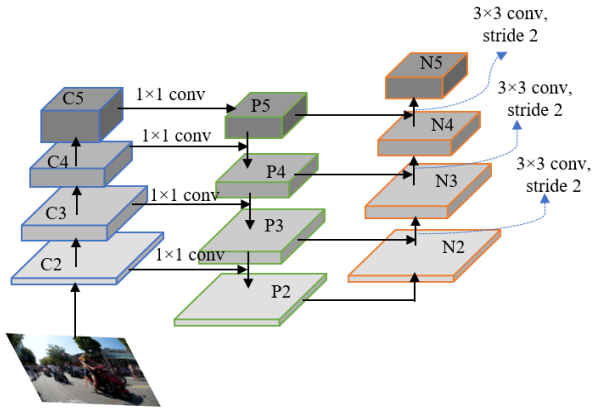
3.1.2 Detection methods based on feature pyramid

In deep convolutional neural networks, shallow layers generally contain weak semantic representations and rich geometric information while deep layers have rich semantic representations but weak geometric information. Since multi-layer detection methods adopt features at different layers independently without integrating different feature levels, they do not achieve good detection results. Recently, with the advent of feature pyramid network [23], many methods based on feature pyramid backbone have been proposed to generate high-level semantic feature maps to improve the detection performance. Instead of exploiting different feature layers independently, feature pyramid detection methods first build a high-level semantic feature pyramid and then adopt each feature map in the pyramid to produce predictions. FPN [23] is the first framework that combines low-level features with high-level features via a top-down pathway and lateral connections to generate high-level semantic feature maps at all scales of the backbone. Figure 5 (a) illustrates the structure of FPN. The top-down pathway includes upsampling operations to upsample feature maps from higher pyramid levels to increase their spatial resolution and lateral connection operations to enhance high-level features with features from the bottom-up pathway. In addition, a 3×3 convolution is applied on each merged feature map to generate final feature maps to reduce the aliasing effect of upsampling operations. By using FPN as the feature extraction network, Faster R-CNN achieves an AP of 33.9 on the COCO minival set. Compared with the original Faster R-CNN based on VGG16, Faster R-CNN with FPN improves AP by 2.3 points. FPN quickly becomes an essential component in the feature extraction network in modern object detectors [6, 76, 77]. Despite the benefits, the straightforward integration in FPN makes it suffer from feature-level imbalance. To be more specific, feature fusion by simple operation ignores the semantic gap between different feature maps at different levels. In addition, semantic feature maps suffer from information loss due to 1×1 convolutional layers in lateral connections, especially with high-level layers. Various techniques have been proposed to overcome the shortcomings of FPN. In PANet [24], the authors suggested that low-level features are helpful in improving the localization capability of the detection network. Based on the idea, they designed a bottom-up path augmentation (Figure 5 (b)) to enhance entire feature pyramid with accurate localization signals existing in low-level features and shorten information path. The bottom-up path augmentation starts from the lowest feature map generated by FPN and gradually reaches the highest feature map. In each block, a 3×3 convolutional layer with stride 2 is first used to reduce the spatial size. Each feature map of FPN is then fused with the down-sampled feature map by lateral connection. Finally, the fused feature map is fed into a 3×3 convolutional layer to generate intermediate map for following subnetworks. Libra R-CNN [25] proposed to enhance multi-level feature maps generated by the FPN backbone by using integrated balanced semantic feature map (Figure 5 (c)). In this method, all feature maps from the backbone are first resized to an intermediate size. The balanced semantic feature map is obtained by averaging all resized feature maps from different layers. The balanced feature map is then refined to be more discriminative. Finally, the output feature maps are generated by rescaling the refined feature map. Overall, this technique will further strengthen the original features and reduce the imbalance at feature level which limits the overall detection

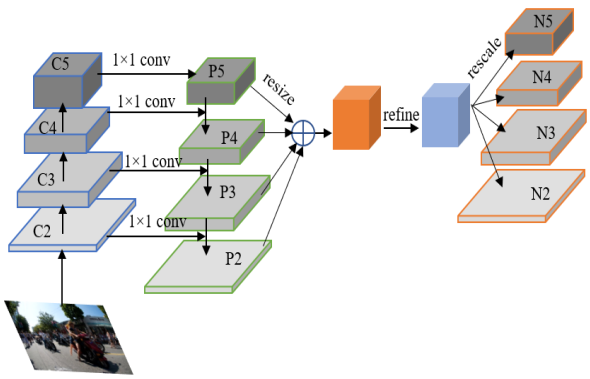
performance. On the other hand, based on the observation that feature map at the highest level of FPN suffers from information lost due to channel reduction of the feature map, AugFPN [26] proposed a residual feature augmentation subnet to enhance feature representations of the highest-level backbone feature map (i.e., *P5*) (Figure 5 (d)). In the residual feature augmentation subnet, ratio-invariant adaptive pooling based on PSP [78] is first attached to the last feature map of the backbone (i.e., *C5*) to generate different context feature maps with different scales. These context feature maps are then rescaled and combined by a novel adaptive spatial fusion module to produce the final context feature map, which is later fused with the last reduced feature map (i.e., *M5*) to generate enhanced feature map. The authors also introduced a consistent supervision scheme which applies the same supervision signals on the feature maps after lateral connection to narrow the semantic gaps between feature maps. Another approach is to alleviate unequal contribution of input features in multi-scale feature fusion. Tan et al. [27] introduced an effective weighted bi-directional feature pyramid network (BiFPN) (Figure 5 (e)) with learnable weights to learn the contribution of different inputs at different resolutions. Based on feature pyramid generated by the EfficientNet backbone [79], BiFPN removes feature maps in top-down path with only one original input feature map (i.e., *P3*, *P7*). In the bottom-up path, original input feature maps are added to corresponding output feature maps if they are at the same pyramid level. Each bidirectional group, which includes top-down and bottom-up path, is treated as one feature layer so bidirectional group is repeated multiple times to add more high-level feature fusion to the feature extraction network. BiFPN also introduced fast and efficient normalized fusion algorithm to add an additional weight for each input when fusing feature maps at different layers. Different from BiFPN, $A^2$-FPN [28] designed an attention aggregation pipeline based on FPN to refine multi-scale feature fusion through attention-guided feature aggregation (Figure 5(f)). The proposed pipeline extracts and fuses feature pyramid progressively through three modules. Multi-level global context module is first designed to replace 1×1 convolution in FPN to mitigate information loss due to channel reductions. Then, to reduce the effects of semantic gap between different layers in feature fusion stage, global attention module (GAM) is designed based on CARAFE [80] to improve the semantic consistency of adjacent feature maps before merging. Finally, global attention content-aware pooling (GACAP) is introduced in the bottom-up path to aggregate more discriminative information for output feature maps. Another approach is proposed by Qiao et al. [29]. The authors introduced Recursive Feature Pyramid (RFP) based on recursive convolutional network (Figure 5(g)) which adds feedback connections from output features into bottom-up path of the FPN network. The RFP modifies the first block of each stage of the ResNet backbone to update bottom-up patch with RFP features via atrous spatial pyramid pooling (ASSP) [81]. A feature fusion module is designed to update output feature maps of the RFP. In addition, all standard 3×3 convolutional layers in the bottom-up path of FPN are replaced by switchable atrous convolution to effectively enlarge the field-of-view of filters.



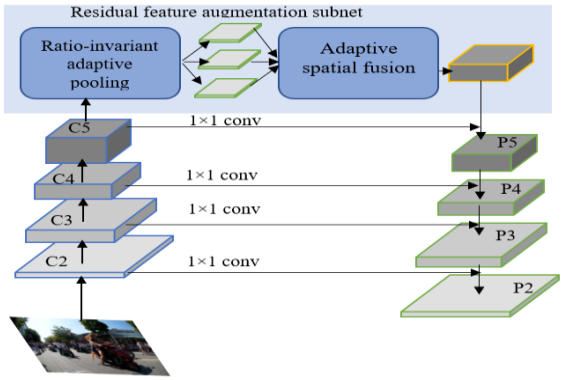

(a) FPN



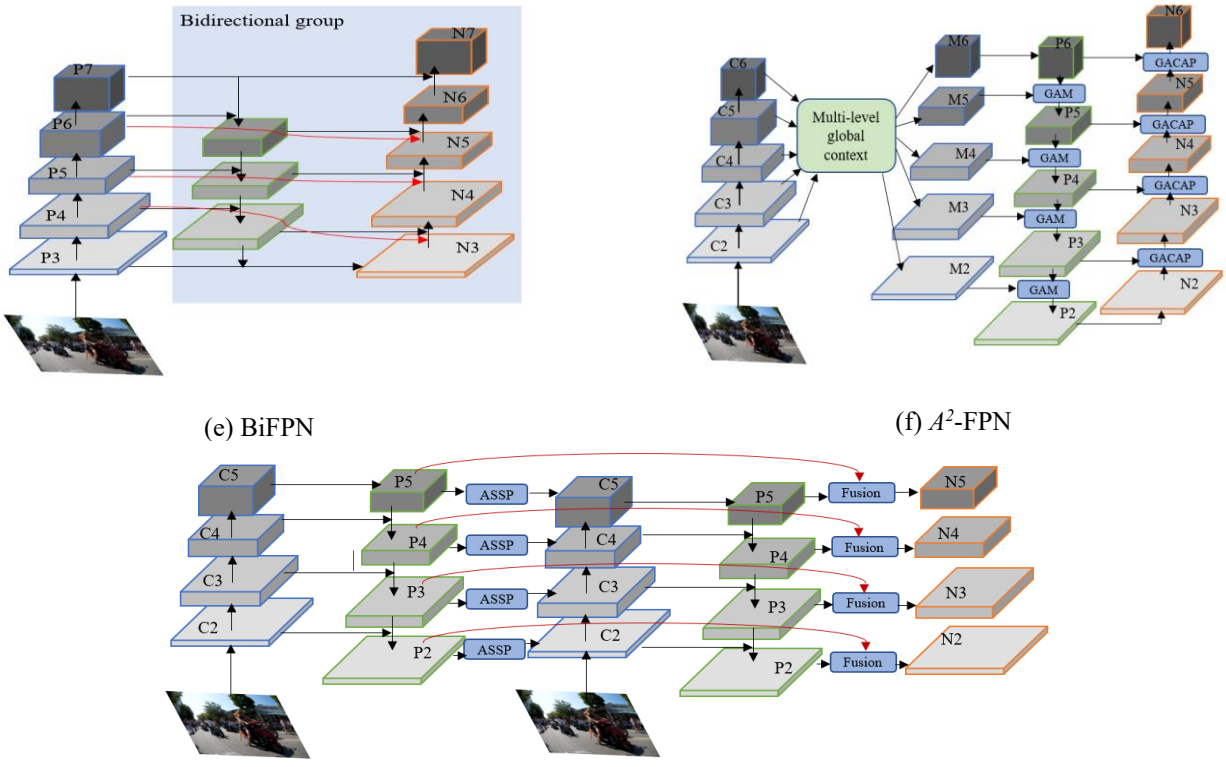

(b) PANet




(c) Libra R-CNN




(d) AugFPN




(e) BiFPN (f) $A^2$-FPN

(g) RFP

**Figure 5.** Object detection based on high-level semantic feature pyramid: (a) FPN. (b) PANet; (c) Libra R-CNN; (d) AugFPN; (e) BiFPN; (f) $A^2$-FPN; (g) RFP

3.1.3 Comparison and analysis

Current state-of-the-art detectors utilize feature pyramid networks due to their robust performance and balance between speed and accuracy. Compared to Faster R-CNN with a ResNet-101 backbone, Faster R-CNN with FPN shows a slight improvement of 0.5 points in Average Precision (AP) on the COCO test-dev benchmark. While this increase is minor, it's achieved with no significant increase in computational cost. However, Libra R-CNN outperforms FPN, achieving 2.5 points higher AP on the same dataset. This showcases the enhanced accuracy of Libra R-CNN, but it should be noted that this improvement might come at the expense of increased complexity in model architecture and potentially higher computational cost. AugFPN, with its residual feature augmentation module, furthers the improvement by 1 AP point on the COCO validation set when compared to FPN, indicating its robustness in detection tasks. However, the model might involve more complexity due to the augmentation module, adding to the computational burden. BiFPN in EfficientDet offers significant enhancements, improving AP by 4 points compared to FPN-based ResNet-50 on the COCO validation set. Additionally, BiFPN requires 3× fewer parameters and 4× fewer FLOPs compared to FPN, presenting a major leap in computational efficiency. This positions it as a powerful choice for object detection tasks, particularly when computational resources are limited. By refining multi-scale feature fusion through three new modules, $A^2$-FPN improves the detection performance by 2.4 points AP compared with the feature extraction network in PANet on the COCO validation set. However, the increase in computational complexity with $A^2$-FPN might limit its applicability in resource-constrained settings, hence a lighter variant $A^2$-FPN-Lite is introduced as a compromise between speed and accuracy. DetectoRS and its RPF network show considerable promise, especially in localizing occluded objects in images, improving AP by 4.2 points over FPN on the COCO validation set. The ability to exploit nearby context information to locate occluded parts of an object gives RPF a distinct advantage. However, the specific mechanism to achieve this might increase the model's complexity and computational demand. Meanwhile, multi-layer object detection methods, though less common, demonstrate potential. For instance, the method by Yang et al. [22] reported a significant relative improvement of 9% over Faster R-CNN on the KITTI test set. These methods, however, are deeply reliant on the backbone architecture for their performance, which might restrict their adaptability. It's worth noting that some feature extraction networks, originally introduced for one-stage object detection frameworks for efficiency, can also be effectively applied to multi-stage frameworks. This demonstrates a level of flexibility in deploying these networks, although the specific pros and cons will largely depend on the context of the application and the specific framework in use.

3.1.4 Open issues

Object detection methods that employ multi-level backbone features have received limited attention in recent years due to the inherent inefficiencies in independently predicting objects at different layers. Most state-of-the-art object detectors currently utilize a feature pyramid approach, prioritizing enhancement of the representation capacity of feature pyramids so that each feature level contains strong semantic information. However, merging feature maps of varying resolutions and representations from these pyramids poses a significant challenge due to potential loss of semantic information. Recent developments, such as A2-FPN [28] and

DetectoRS [29], have addressed this issue by designing additional networks to mitigate information loss and tackle semantic gap problems. These methods have demonstrated promising results, but their increased complexity-with the addition of extra networks-invariably leads to more parameters and hyperparameters. This, in turn, introduces larger computational costs and requires more manual adjustments, an aspect that may not be feasible in all scenarios, particularly in resource-constrained environments. Another approach to improving the effectiveness of feature pyramid utilization is the use of Neural Architecture Search (NAS) [82] to identify the optimal FPN structure [83]. While promising, this approach is also computationally intensive, especially during the search process. To address these issues, future work could explore methods that balance accuracy and computational efficiency. For instance, lightweight neural architectures could be developed to minimize the parameters while maintaining detection performance. Techniques such as network pruning and quantization could also be used to optimize existing architectures. Furthermore, automated hyperparameter tuning algorithms, like Bayesian optimization, could mitigate the need for extensive manual tweaks. Ultimately, the objective is to develop an efficient method that fully exploits each feature level to generate increasingly powerful feature levels without exacerbating computational demands.

### 3.2 Recent techniques in object representations

According to the way of representing objects in the proposal generation stage, existing object detection methods can be categorized into two groups: anchor box-based methods and point-based methods. Anchor box-based methods represent each object as a rectangular bounding box on feature maps while point-based methods employ points on feature maps to depict objects. Early two-stage object detectors usually employ a rectangular bounding box representation scheme in the proposal generation stage to produce proposal boxes. Recently, with the emergence of FPN [23] and Focal loss [6], point representations have attracted increasing attention from researchers due to their efficiency. It should be noted that although point representations are usually employed in one-stage object detection pipelines, they can be modified to adapt to two-stage object detection pipelines to improve the overall detection performance [37, 38, 44, 50, 51, 84].

#### 3.2.1 Rectangular bounding box representations

Early two-stage object detectors are usually based on rectangle bounding boxes (i.e., anchor boxes, proposal boxes, and final detection boxes) to represent objects at different recognition stages of the detection process [30, 55, 85]. A bounding box $B=\{x, y, w, h\}$ is a *4-d* representation containing the spatial location of an object in image, with $(x, y)$ representing the center points and $(w, h)$ representing the width and height of object in image. The rectangular bounding box is the dominant type of object representation in early deep learning-based object detection frameworks, especially in two-stage detection methods, due to its convenience and efficiency [44]. In the proposal generation stage of two-stage object detection pipelines, input rectangular bounding boxes are usually called anchor boxes which are hypothesized to represent objects at different scales and aspect ratios. For each anchor box, features are extracted as object representations which are then used to generate proposal boxes by a proposal generation network. In general, a set of anchor boxes are generated at each location on the feature map by predefined scheme [7, 23, 30] or learnt scheme [31-33]. Anchor box generation based on predefined scheme is first introduced in Faster R-CNN [30], where an anchor box is centered at the sliding window and associated with a scale and aspect ratio. There are three predefined scales (i.e., $128^2$, $256^2$, $512^2$) and three predefined aspect ratios (i.e., 1:1, 1:2, 2:1), resulting nine anchor boxes at each position on the feature map in Faster R-CNN. In the study [23], RPN employs different scales at different feature layers to define anchor boxes. For anchor shape, three predefined aspect ratios are used, yielding 15 anchor boxes at each location over the feature pyramid. Recently, ThunderNet [7] has used a single scale for each feature layer and five aspect ratios (i.e., 1:2, 3:4, 1:1, 4:3, 2:1) to generate anchor boxes.

In the predefined anchoring scheme, the number of anchor boxes and the size of each box need to be designed carefully. To be more specific, too few anchor boxes may be insufficient to cover a large range of objects in various sizes and ratios, thus hindering the detection accuracy of the detector. In contrast, too many anchor boxes generated at the beginning require more parameters, which may lead to overfitting and significant computational cost due to a large number of false candidates. In addition, the anchor box shapes have to be manually tweaked to improve detection accuracy of the detector on specific domains. For example, since texts may have large aspect ratios compared with generic objects, Liao et al. [86] used seven predefined aspect ratios (i.e., 5:1, 3:1, 2:1, 1:1, 1:2, 1:3, 1:5) for scene text detection. In the study [87], the authors used one aspect ratio (i.e., width/height=5) for license plate detection since license plates are usually rectangular in shape. Wang et al. [88] employed single aspect ratio of 0.41 for pedestrian detection as this value is the average aspect ratio of pedestrians. To mitigate the issues of predefined anchoring scheme, learnt anchoring scheme [31-33] proposes to generate learnable anchor boxes, where anchor shape is learnt during the training process to generate high-quality anchor boxes to boost the detection performance. MetaAnchor [31] introduced an anchor function generator which maps any box prior to corresponding anchor function. The anchor function generator is formulated as a simple two-layer network and computed from customized prior boxes. Based on neural network weight prediction mechanism, anchor function generator could be implemented and embedded into existing object detection methods for joint optimization. The proposed mechanism is shown to be robust to anchor settings as it may cover various kinds of object boxes with any shape. However, the method shows minor improvements for two-stage anchor box-based object detection pipeline and requires customized prior boxes to be chosen by handcraft. In addition, it introduces an extra network for predicting weights, which leads to increased parameters and computational cost. Similar to the MetaAnchor scheme, Zhong et al. [32] introduced an anchor optimization scheme in which anchor shape is automatically learned during the training process. A localization loss is introduced which is to minimize the error between the ground truth box and the predicted offset relative to the anchor box. The error is then backpropagated to the anchor shapes and other parameters in the whole network to automatically learn the anchor box. The anchor shape is warmed up by soft assignment and online clustering scheme. Since the anchor shape is learned during training, the optimization scheme is more suitable for the specific data and network structure. In addition, the anchor

shape initialization needs to be designed carefully to achieve the best detection performance. Recently, Sparse R-CNN [33] defined a fixed small set of learned candidate boxes to represent objects. Each candidate box includes a learnable proposal box and a learnable proposal feature. Proposal box is a *4-d* representation containing initial center coordinates, height and width of object while proposal feature is a high-dimension vector which encodes rich information of object. The parameters of proposal boxes and features will be updated with the backpropagation algorithm during the training process and optimized together with other parameters in the whole network.

Although rectangular bounding box representations are easy to implement and facilitate computational process, they provide only coarse localization of objects. As a result, the feature extraction process in the object prediction stage will produce coarse representations of objects. The coarse extracted features may be heavily influenced by background information and foreground regions that contain little semantic information, which leads to inferior detection performance in object detection.

3.2.2 Point representations

To tackle the shortcomings of rectangular bounding box representations, various methods have been proposed to develop more efficient object representations. Recent state-of-the-art object detection methods directly employ points on the feature maps to represent objects and form bounding boxes based on proposal points. For simplicity, we call these methods as point representations. Point representations can be grouped into two groups: anchor-point representations and key-point representations. Anchor-point representations, including DenseBox [34], FSAF [35], Guided Anchoring [36], FCOS [37], FoveaBox [38], and SPAD [39], employ each pixel on a feature map as object representation. The proposal generation network first classifies each pixel into foreground or background class, and then directly regresses the distances from the foreground point to the four sides of the ground truth bounding box to generate final prediction. DenseBox [34] (Figure 6 (a)) proposed object detection based on FCNs [89]. DenseBox directly predicts a *4-d* vector and a confident score of being an object at each location on a feature map. The *4-d* vector represents the relative offsets from the top-left and bottom-right boundaries of the target bounding box to pixel location. To solve shape variations of object, DenseBox employs image pyramid by cropping and resizing input images to different sizes. Different from DenseBox, Zhu et al. [35] introduced a novel feature selective anchor-free module (FSAF) (Figure 6 (b)) which takes locations on the feature pyramid as inputs and directly feeds these locations into two convolutional branches: a classification branch for predicting $K$ class scores for each location and a regression branch for producing 4 offsets encoding the distances between the current pixel location and the top, left, bottom, and right boundaries of the target bounding box. The FSAF module can be inserted into an anchor box-based detector with a feature pyramid backbone to help learning objects which are hard to be modeled by anchor-based mechanism. Another approach, Guided Anchoring [36] (Figure 6(c)), predicts each location on the feature map by using probability maps generated by an anchor location prediction network. At each active location where the center of objects is likely to exist, anchor shape prediction network is designed to predict the best shape for corresponding object. Recently, FCOS [6] directly classified each point on a feature map and regresses the target bounding box for positive points, thus eliminating complicated problems related to anchor bounding boxes. A point is considered as a positive point if it falls into any ground truth box and the class label of the point is the class label of the ground truth box (Figure 6 (d)). If a location falls into multiple ground truth bounding boxes, FCOS based on the multi-level features generated by FPN to assign an appropriate bounding box. In addition, FCOS introduced a novel center-ness branch with only one convolution layer to generate a centerness score for a positive point based on the distance from the point to the center of the corresponding object. This score is then used to eliminate low-quality bounding boxes. FCOS achieved state-of-the-art detection performance with much less design complexity and computational cost, which encourages researchers to follow the anchor-free mechanism in designing object detection pipeline. Similar to FCOS, FoveaBox [38] directly predicts the object score and the corresponding boundaries for each spatial location on the feature pyramid. FoveaBox uses fovea area, which is generated by shrinking ground truth box based on a shrunk factor, to define positive anchor points (Figure 6 (e)). In SAPD [39], the authors proved that the training strategy used in most anchor-point detectors, which treats anchor points independently, may lead to compromise the detection performance due to different contributions of anchor points to the network loss according to their spatial location. Based on this, they proposed a soft-weighted scheme which adds an attention weight for each anchor point based on its geometrical relation with the instance boundaries. The soft-weighted scheme forces the network to focus more on positive points near the center of object instance.

Alternatively, key-point representations, including CornerNet [40], CenterNet [41], ExtremeNet [42], Objects as Points [43], RepPoints [44], VarifocalNet [45], use keypoints such as center points, corners points, or extreme points to represent an object. These methods first predict the locations of keypoints of the bounding box and then group those key points to form a bounding box. CornerNet [40] proposed to represent an object as a pair of corner keypoints covering the object (i.e., the top-left corner and bottom-right corner of the bounding box) (Figure 7 (a)). The authors also introduced a novel corner pooling layer to better localize each type of corner of bounding boxes by employing prior knowledge. To group corner points that belong to the same object and produce a bounding box, the network first predicts an embedding vector for each detected corner and then groups corners based on the distances between their embeddings. Based on CornerNet, CenterNet [41] adds one extra keypoint (i.e., a center keypoint) to represent an object (Figure 7 (b)). While conner keypoints are used to generate bounding boxes, the center keypoint is employed to filter out incorrect bounding boxes. A center pooling layer is designed in CenterNet to capture rich semantic information within the center of the bounding box, thus improving the detection of the center keypoint. In addition, a cascade corner pooling layer is proposed to add center information to boundary information to improve conner localization capability of the network. Instead of using conner points to represent objects, ExtremeNet [42] defines four extreme points (i.e., top-most, left-most, bottom-most, right-most) and a center point to represent an object (Figure 7 (c)). Unlike conner points, extreme points usually lie on object, thus containing strong semantic information. This facilitates the extreme point detection performance. ExtremeNet also introduced a center grouping algorithm

which analyses the geometric structure of extreme points and their center to group extreme points and produce bounding box. To eliminate the grouping stage in key-point detectors, Zhou et al. [43] represented objects by a single point at the center of bounding box (Figure 7 (d)). Peaks in the heatmap generated by the keypoint estimation network correspond to object center point. The keypoint prediction network predicts the height and width of bounding boxes covering objects based on the keypoint values of the center point. Different from corner points and extreme points, RepPoints [44] defined a set of adaptive sample points (e.g., 9 points) with the deformable convolution [54] to represent an object (Figure 7 (e)). RepPoints starts from the center point and produces other points via regressing offset values over the center point. For refinement object bounding box, RepPoints produces a *2-d* regression vector (i.e., (Δx; Δy)) instead of *4-d* regression vector in most modern object detectors to alleviate the problem of scale differences among the bounding box regression parameters. For forming bounding boxes, RepPoints designed a converting function which produces bounding box based on adaptive sample points. Recently, VarifocalNet [45] defined a star-shaped bounding box with nine fixed sampling points to represent an object (Figure 7 (f)). The star-shaped bounding box representation can capture the geometric relations between a bounding box and its nearby contextual information. This facilitates the regression process which encodes the misalignment between the predicted box and the ground truth box.

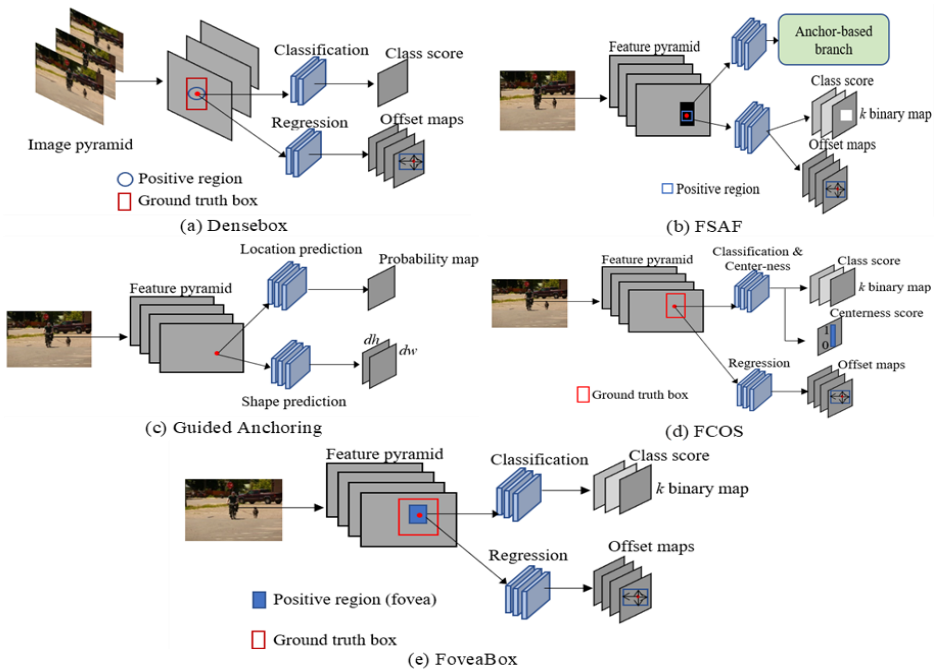



**Figure 6.** Object detection based on anchor-point representations: (a) Densebox; (b) FSAF; (c) Guided Anchoring; (d) FCOS; (e) FoveaBox

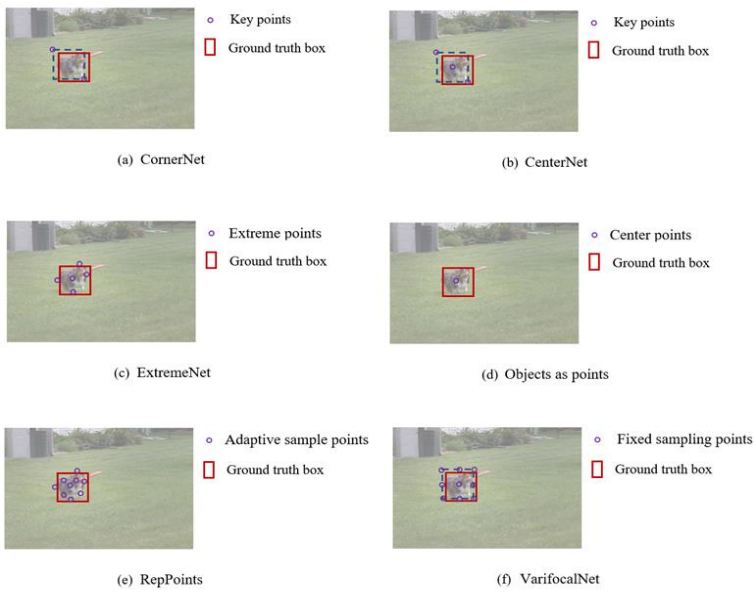



**Figure 7.** Key-points representation in object detection: (a) CornerNet; (b) CenterNet; (c) ExtremeNet; (d) Objects as points; (e) RepPoints; (f) VarifocalNet

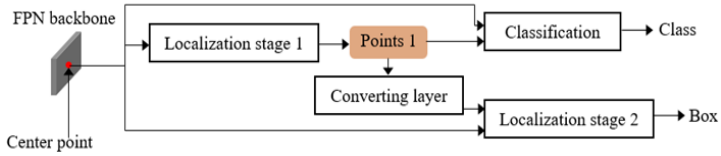



**Figure 8.** The architecture of RPDet [44]

Anchor-point and key-point representations are usually used in one-stage object detection frameworks due to their efficiency. However, they can be modified to adapt to two-stage object detection frameworks to improve the overall detection performance [33, 37, 38, 44, 50, 51, 84]. For example, FCOS [37] proposed to replace the anchor box scheme with the anchor point scheme in RPN. The results showed that RPN with the anchor-point scheme boosts the localization capability of the network by a large margin. In RPDet [44], a two-stage object detection approach (Figure 8), a set of refined points representing objects are produced by a localization subnet in the first stage. Proposal boxes are generated based on refined points by using a converting layer. These refined points and proposal boxes are then fed into a classification subnet at the second stage. Compared with baseline detectors based on bounding box representations, RPDet significantly improves the detection performance on the same dataset.

3.2.3 Comparison and analysis

Object detection methods utilizing rectangular bounding box representations have primarily pivoted towards learnt anchoring schemes in recent times. For instance, in the study [36], the Faster R-CNN improved detection performance by 2.7 points AP on the COCO test-dev set, by replacing the predefined anchor scheme with a guided one to automatically learn anchor shape and location. The benefit of this approach is the production of high-quality proposal boxes, increasing the efficiency of the training process. Yet, it also introduces more complexity to the model. In a similar vein, Zhong et al. [32] introduced a learning scheme to learn anchor shapes during the training process. The result was consistent improvements across different datasets and architectures with negligible extra training and inference costs. Notably, this learning scheme proved more robust with one-stage baselines compared to two-stage ones, due to the efficiency reduction caused by the second stage in two-stage baselines. Meanwhile, Sparse R-CNN [33], utilizing learnable proposal boxes, significantly outperformed Faster R-CNN based on predefined anchor boxes. However, its superior performance comes with the cost of a more complex network structure and possibly higher computational load.

Recently, the efficiency of point representations, aided by FPN and Focal loss, has been a focal point of research. Anchor-point representations, particularly recent methods [37-39], have achieved substantial improvements over anchor box-based methods. Notably, FCOS [38] outperformed both RetinaNet and Faster R-CNN by substantial margins on the test-dev split of the MS-COCO benchmark. Yet, these methods may potentially be more complex and resource-intensive than those using box representations. In the domain of key-point representations, CornerNet [40], an innovative one-stage approach, achieved competitive detection performance compared with several anchor box-based two-stage methods. The ExtremeNet [42] improved on this by combining extreme points and center point, achieving a modest increase in performance. Recently, VarifocalNet [45], which defines nine fixed sampling points to represent an object, has surpassed almost all state-of-the-art detectors. This result is promising for future research, but again, the complexity and computational cost may be a drawback. In comparison, anchor-point methods have simpler network architectures, which leads to faster training and inference speeds. However, key-point methods can offer superior accuracy by encoding geometric relations between an object and its nearby contextual information, highlighting a trade-off between complexity, computational cost, and detection performance in the field of object detection methods.

3.2.4 Open issues

Firstly, the efficiency of point representation strategies has led to their adoption in multi-stage object detection pipelines in several recent studies [33, 37, 38, 44, 50, 51, 84]. However, these strategies have not been explored as extensively in multi-stage object detectors as in one-stage object detectors. For example, FCOS [37] used the anchor-point scheme to replace the anchor-box scheme in RPN, which improved the localization capability of RPN. Yet, the effects of the anchor-point scheme on the second stage have not been thoroughly investigated. Potential solutions to address this issue could be conducting further research specifically focusing on the implications of the anchor-point scheme on the second stage and modifying the scheme to better suit the second stage based on the findings.

Secondly, the fine localization capabilities of point representations have drawn increasing attention. In the study [70], the authors identified the main difference between anchor box-based and point-based detectors as the sampling strategy (Section 5.3). Following this, they proposed an adaptive training sample selection strategy that automatically divides positive and negative samples based on statistical characteristics of the object. This sampling strategy allowed RetinaNet [6] with an anchor box representation scheme to achieve comparable detection performance with FCOS [37] with an anchor-point representation scheme. However, this strategy was proposed for one-stage object detectors, leaving the same issue for multi-stage object detectors unexplored. In response to this, future studies could aim to adapt this strategy to multi-stage object detectors and analyze its effect on detection performance.

Thirdly, despite key-point detectors being time-consuming and complex models due to their grouping algorithm, they have the advantage of superior detection accuracy compared to anchor-point detectors. Several studies proposed combining anchor-point and key-point representations to encode multiple levels of objects features and avoid the time-consuming grouping algorithm [90]. However, it is still unclear which key-points are beneficial for bounding box regression. To address this, an in-depth analysis of the role of each type of key-point is necessary. This could involve a comprehensive evaluation of the importance of each key-point in bounding box regression or the development of a new algorithm that

automatically selects the most beneficial key-points.

### 3.3 Recent techniques in proposal generation

Object proposals are the main results of the first stage in two-stage object detection pipelines. Based on object representations, proposal generation network takes corresponding features generated by the feature extraction network as inputs to generate proposals. Proposals are used by the second stage to generate final predictions. An ideal proposal generation method should generate as few proposals as possible while covering all object instances in the input image. To generate proposals, early object detection methods are usually based on merging super-pixels (e.g., Selective Search [46]) and sliding windows (e.g., Edge Boxes [47]). Although these approaches have been broadly used as the proposal generation methods of choice by many object detectors, they exhibit issues and limitations. First, they require significant computation to process millions of proposals per image. Second, since these methods have no learnable parameters, they are external modules independent of the detector. To solve these issues, Ren et al. [30] introduced novel region proposal network (RPN) which has become the paradigm for designing two-stage object detection pipeline. Figure 9 (a) illustrates the structure of RPN. It includes a 3×3 convolution layer followed by two parallel 1×1 convolution layers for regression and classification. For each anchor $B = \{x_B, y_B, w_B, h_B\}$, where $\{x_B, y_B\}$ is the center point of the anchor and $\{w_B, h_B\}$ is the width and height of the anchor, the classification branch outputs two predictions: the score of it being background and the score of it being foreground. The regression branch aims to predict the transformation $\Delta$ from the anchor $B$ to the target ground truth bounding box $G = \{x_G, y_G, w_G, h_G\}$ represented as follows:

$$\Delta_x = (x_G - x_B)/w_B, \Delta_y = (y_G - y_B)/h_B \quad (1)$$

$$\Delta_w = \log(w_G/w_B), \Delta_h = \log(h_G/h_B) \quad (2)$$

RPN has been used as a proposal generation network in many deep learning-based object detectors. However, RPN is weak at locating objects with extreme shapes or objects in difficult environments because of information loss caused by pooling layers in CNN structure and object representations with predefined shapes. To tackle the shortcomings of RPN, Zhong et al. [48] proposed a cascade architecture in proposal generation stage to improve score and location of proposals (Figure 9 (b), denoted as Iterative RPN in this paper). The proposed cascade structure includes two RPNs. The second RPN takes proposals produced by the first RPN as inputs and further classifies and regresses to generate high-quality proposals. The cascade structure improves the localization capability for objects with various sizes. However, since the anchor location and shape change after each RPN, this structure causes mismatch between anchor boxes and their representations. Another approach, Cascade RPN [15] (Figure 9 (c)) employs one anchor box at each spatial location on a feature map and performs box refinement through multi-stage refinement scheme. In the first refinement stage, dilated convolution layer is used to produce anchor with more semantic information. Since only one anchor is defined at each location in the first stage, Cascade RPN uses the center of the anchor and ground truth box to define positive anchor instead of IoU threshold. Based on the regressed box and features produced by the first refinement stage, the second refinement stage with adaptive convolution layer is used to classify and regress each active bounding box to produce final proposals. To improve the localization capability of RPN, Qiu et al. [49] introduced BorderRPN (Figure 9 (e)) which adds two border alignment modules (BAM) into the original RPN. The first BAM produces border offsets which are then combined with coarse box offsets generated by RPN to enhance bounding box locations. The second BAM produces border classification scores which replace bounding box scores generated by RPN. In both BAM branches, BorderAlign [49] module is adopted to extract and exploit border features which are crucial for achieving better detection accuracy.

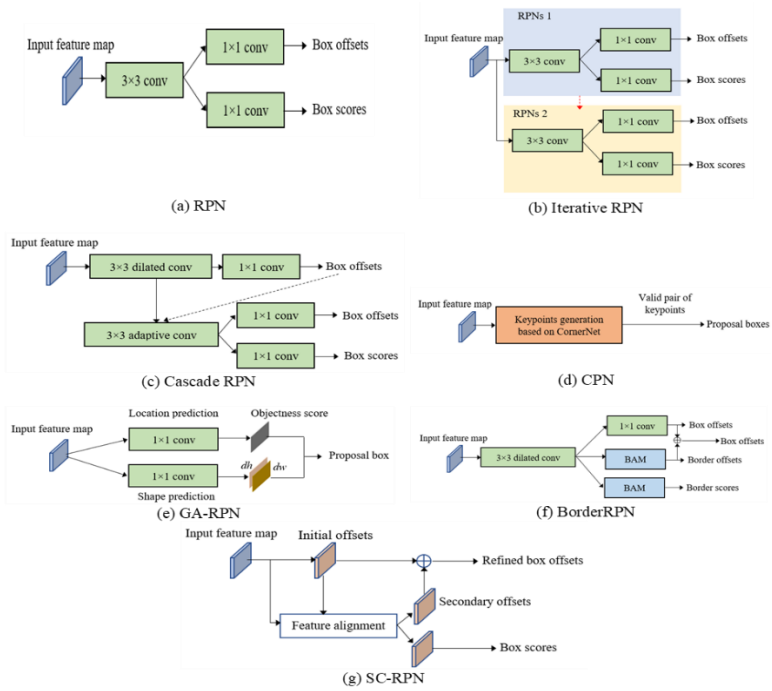



**Figure 9.** The structure of proposal generation networks: (a) RPN; (b) Iterative RPN; (c) Cascade RPN; (d) BorderRPN; (e) CPN; (f) GA-RPN; (g) SC-RPN

By observation that object detectors based on point representations usually obtain high recall since they can locate objects of different geometries, especially those with rare shapes, Corner Proposal Network (CPN) [50] (Figure 9 (d)) proposed to use CornerNet [40] as the proposal generation network to generate proposals. Based on keypoints generated by CornerNet, CPN assigns each valid pair of keypoints as a proposal box. These proposal boxes are then classified by the second stage of the network. In the study [36], GA-RPN (Figure 9 (e)) was developed to generate high-quality proposal boxes with learnable shapes. GA-RPN includes two subnets for predicting location and shape of proposal boxes. In the location prediction subnet, a 1×1 convolution layer is first applied to each feature map on the feature pyramid to obtain corresponding objectness score map. Each objectness score map is then converted to probability values via an element-wise sigmoid function. The probability maps generated by the location prediction subnet indicate possible locations of objects. Proposal boxes are generated based on these possible locations so that most false boxes are eliminated. In the shape prediction subnet, a 1×1 convolutional layer is applied to base feature map to generate a two-channel map that contains the values to update the width and height of objects by an element-wise transform layer. GA-RPN is applied to multiple feature maps, and parameters are shared across all feature levels. Recently, SC-RPN [51] introduced a novel region proposal approach (Figure 9 (f)) which aims to tackle the correlation issue of classification score and location accuracy in RPN as well as the shortcomings of anchor-based representations used in RPN. SC-RPN is specially designed for object detectors based on point representations. The initial offsets, which represent coarse object boundary, are first predicted from input feature maps. A feature alignment operation is then carried out to produce secondary offsets and classification scores. Final offsets are generated by combining initial offsets and secondary offsets to form refined bounding box. The experimental results showed that SC-RPN is very effective when replacing RPN in two-stage object detectors.

3.3.1 Comparison and analysis

In comparison with the original Region Proposal Network (RPN), both Iterative RPN [48] and Cascade RPN [15] present significant improvements. Iterative RPN enhances recall @0.7 by 8.8 points and average recall (AR) by 4.8 points on the ImageNet DET val2 set. However, it introduces increased complexity due to the iterative proposal refinement process, leading to a trade-off between performance and computational cost. On the other hand, Cascade RPN, which incorporates dilated and adaptive convolution at each refinement stage, augments anchor features with more semantic information, enhancing the localization capabilities of the network. Despite the 13.4 points AR improvement on the COCO 2017 val split set, Cascade RPN introduces a higher level of complexity and computational cost compared to traditional RPN, making it less optimal for resource-constrained applications. Recently, SC-RPN [51] integrated point representations into the region proposal approach, improving the localization capability of the region proposal network and outperforming previous proposal generation methods. Specifically, it enhances $AR_{1000}$ significantly compared to RPN, Iterative RPN, GA-RPN [34], and Cascade RPN on the COCO 2017 val split set. The integration of SC-RPN boosts Faster R-CNN detection performance by 3.8 points mAP on the COCO 2017 test-dev split. However, this technique may have higher computational demands due to the additional point representation integration process. Despite this, SC-RPN presents a promising research direction for generating high-quality object proposals, an area which has received relatively less attention recently, thereby demonstrating its potential for driving advances in object detection tasks.

3.3.2 Open issues

The quality of the proposal boxes generated by the proposal generation network plays a vital role in the overall detection performance of two-stage object detection frameworks. Despite this, the proposal generation network has somewhat been overshadowed by the detection network in recent research. Addressing this imbalance, we suggest focusing on several key areas:

Firstly, we need to delve into the distinction in optimization targets between classification and regression within the proposal generation network. The potential discord between these two branches might lead to the generation of substandard proposals. To tackle this, an adaptive loss function could be developed that can balance the optimization between the two tasks. This might involve further research into multi-task learning and loss functions that can dynamically adjust their weights based on the training progress.

Secondly, the results of SC-RPN [51] demonstrated that the proposal generation network using point representation scheme produces proposals with enhanced fitting ability. Hence, there is a need to design an optimized proposal generation structure grounded in the point representation scheme. This could involve the development of algorithms that can better capture spatial and semantic relationships between different points. Alternatively, the application of advanced point cloud processing methods or attention mechanisms might further enhance the representational power of the point-based proposals. These suggested areas of research may contribute to the development of high-quality proposal generation strategies that can ultimately enhance the overall performance of two-stage object detection frameworks.

## 4. OBJECT PREDICTION STAGE

In the object prediction stage, proposal features are first extracted by a proposal extraction method based on proposals and feature maps generated by the proposal generation stage. A detection head is then employed to classify each proposal to one of the classes and further regress its bounding box. Since multi-stage object detection frameworks usually adopt fully connected layers in the detection head, proposal extraction method needs to extract fixed-sized feature maps for each proposal so that the detection head can share parameters over proposals. Many early deep object detectors employ RoI pooling scheme as proposal extraction method. However, quantization process in RoI pooling scheme causes misalignments between proposal and extracted features. Recently, RoI Align, Deformable RoI Pooling, and Discriminative RoI Pooling have been proposed to alleviate this problem. For the detection head, region-based convolutional neural network (R-CNN) dominates the field of two-stage object detection frameworks. R-CNN shares a head for both classification and bounding box regression, which causes spatial misalignment between the two branches. Various methods have been proposed in recent years to tackle this problem. These methods can be divided into two groups:

localization sensitive scores-based methods and task-specific structure methods. While localization sensitive scores-based methods produce a localization sensitive score for each proposal to update its classification score, task-specific structure methods propose to design different structures for different tasks.

### 4.1 Recent techniques in proposal extraction

Proposal extraction aims to generate proposal features based on proposal boxes and feature maps generated by the first stage. Proposal features are then fed to the detection network to produce final results. Since the detection head in multi-stage object detectors usually contains fully connected layers, a proposal extraction method needs to be designed to extract fixed-sized feature maps for each proposal so that the detection head can share parameters over proposals. Various object detection methods have been followed RoI Pooling [52] and RoI Align [53] scheme, which adopt quantization process, bilinear interpolation, and pooling operation to produce fixed-sized proposals. Recently, Deformable RoI Pooling [54] and Discriminative RoI Pooling [55] have been introduced based on deformable convolution to generate offsets added to proposal features to produce fixed-sized proposal feature maps with more nearby semantic information to improve the localization capability of the detection network. In the following, we analyze in detail these proposal extraction methods.

#### 4.1.1 RoI pooling

RoI Pooling was first introduced in Fast R-CNN [52]. This proposal extraction scheme is a special case of spatial pyramid pooling [89] with only one pyramid level. RoI Pooling is used to reshape input proposals with arbitrary size into output proposals with fixed size to overcome the size constraint in fully connected (FC) layers in R-CNN subnet. In RoI Pooling process, a $h \times w$ proposal is first divided into $h' \times w'$ proposal of approximate size $h/h' \times w/w'$, where $(h', w')$ is the proposal shape required by FC layers. Max pooling is then used to copy max values in input sub-regions to output value. The number of output channels is equal to the number of input channels for RoI Pooling layer. In Faster R-CNN [30], RoI Pooling takes two inputs: A feature map generated by the feature extraction network and $n$ proposals generated by RPN. Because proposals are generated based on input image size, RoI Pooling first rescales proposals to feature map size by quantitating of coordinates on the feature map. Next, if the size of a proposal is larger than the fixed size RoI, max pooling is used to copy max values in input sub-regions to output value. Otherwise, if the size of a proposal is smaller than the fixed size proposal, it is enlarged by replicating some values to fill extra spaces. Figure 10 (a) illustrates the RoI Pooling scheme. RoI Pooling scheme has been used as proposal extraction method in many modern object detectors [91, 92]. However, the RoI Pooling scheme is not suitable in some cases, especially for detecting small objects. Due to information loss through pooling operation in CNN, feature representations of small objects become weaker in deeper CNN layers. As a result, filling extra space in output proposal with replicated values leads to destroying the original structure of small objects. In addition, the quantization process in RoI Pooling scheme leads to inaccurate representations due to misalignments between proposals and extracted features. These inaccurate representations prevent the detection network from correctly classifying small objects, which leads to a decrease in the detection performance of the whole network.

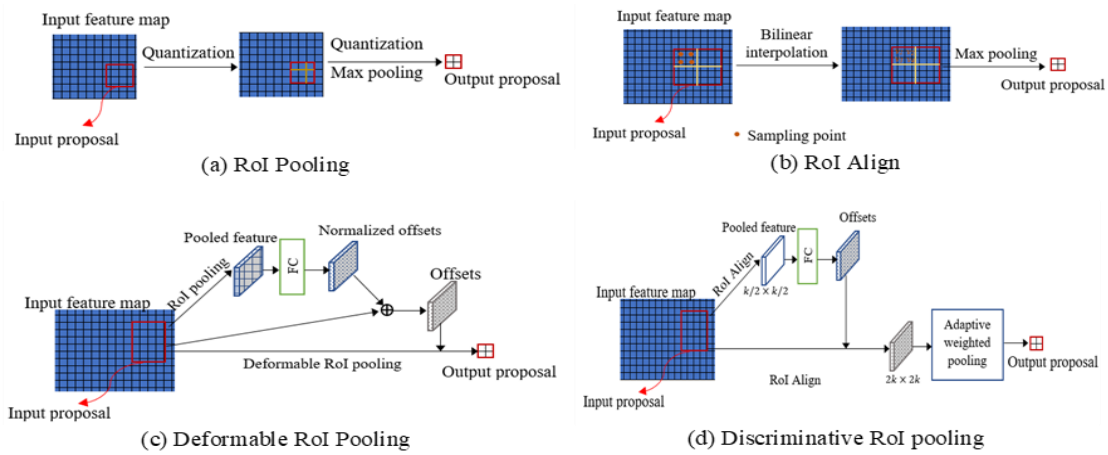



**Figure 10.** Proposal extraction methods: (a) RoI Pooling; (b) RoI Align; (c) Deformable RoI Pooling; (d) Discriminative RoI pooling

#### 4.1.2 RoI align

To alleviate misalignments between input proposal and extracted features due to quantization process in RoI Pooling scheme, RoI Align [53] first divides original proposals into $k \times k$ sub-regions based on the size of the fixed RoI and the size of the pooling layer. Next, four sampling points are created within each sub-region. Based on sampling points, bilinear interpolation is applied in each sub-region to sample data for each sub-region. Finally, max pooling or average pooling is used to calculate corresponding values in input sub-regions to output value. Figure 10 (b) illustrates RoI Align scheme. Since RoI Align removes quantization process in RoI Pooling, it properly aligns the extracted features with the input proposal. As a result, RoI Align significantly improves the detection performance compared with RoI pooling scheme. RoI Align scheme has become standard proposal extraction scheme in recent object detectors [93-95].

#### 4.1.3 Deformable RoI pooling

Deformable RoI Pooling [54, 96] (Figure 10 (c)) is a novel

proposal extraction scheme based on deformable convolution [54]. Deformable RoI Pooling adds a fully connected layer after the pooled feature map generated by RoI Pooling scheme to learn the normalized offsets which are then augmented with the RoI shape by element-wise product to generate offset values. The offset values are used to add an additional offset to each sub-region of the pooled proposal to produce output proposal. Deformable RoI Pooling helps to produce proposals with more nearby semantic information, thus enhancing the localization capability of the detection network, especially for non-rigid objects.

4.1.4 Discriminative RoI pooling

Based on Deformable RoI pooling, D2DET [55] introduced Discriminative RoI Pooling, a novel proposal extraction scheme (Figure 10 (d)) which adds adaptive weighting to input sub-region to produce output proposals with discriminative features. First, based on input proposals, RoI Align is applied after the input feature map to obtain $k/2 \times k/2$ sub-regions, where $k$ is the output size. These sub-regions are fed into a fully connected layer to learn corresponding offsets. At the bottom path, RoI Align is also used to produce $2k$ sub-regions from the input feature map. This pooled feature map is then augmented with offset values generated by the top path to produce output proposal. After generating 4 sampling points in each sub-region, an adaptive weighted pooling layer is applied to add an adaptive learnable weight for each sampling point. Finally, an average pooling operation with stride of 2 is used to produce output proposal. With learnable weights for each sampling point and the predicted offsets added to each sub-region, the output proposal contains discriminative information relevant to both the object and its context, which further improves the classification capability of the detection network.

4.1.5 Performance summary

Based on Mask R-CNN [53], RoI Align scheme improves AP by about 3 points over RoI Pooling scheme on the MS COCO minival set. By fixing the misalignment between proposals and extracted features, RoI Align scheme has a large impact on the subsequent detection network. On the other hand, Deformable RoI Pooling scheme produces minor performance gains, about 0.3 points mAP@0.5, compared with RoI Pooling scheme on the VOC 2007 test images. When using both deformable convolution and Deformable RoI Pooling scheme, Faster R-CNN achieves significant accuracy improvements, especially at the strict mAP@0.7. The results show that Deformable RoI Pooling scheme obtains the best performance when combined with deformable convolution layers for extracting object features. Since the Discriminative RoI Pooling scheme is specially designed for the subsequent classification network, it produces noticeable performance gains. Specifically, by replacing RoI Pooling scheme by Discriminative RoI Pooling scheme, Faster R-CNN with FPN baseline improves the detection performance from 38 to 39.3 AP on the COCO minival.

4.1.6 Open issues

Since proposal extraction module produces fixed-sized proposal feature maps for the subsequent detection network, extracted proposal feature maps need both precise localization and discriminative features so that the classification and regression branches in the subsequent detection network produce high-quality results. D2Det [55] is the first framework that addresses this problem. However, D2Det proposed Discriminative RoI Pooling scheme for producing proposal maps with more discriminative features to facilitate the subsequent classification performance. The regression branch is still based on proposal maps generated by RoI Align scheme. For this reason, it is necessary to develop an efficient proposal extraction method to extract proposal features with both precise localization and discriminative features.

### 4.2 Recent techniques in detection block

Detection block is the final module in two-stage object detection pipelines. It serves as the crucial finale in two-stage object detection pipelines, converting proposal features derived from the proposal extraction mechanism into final predictions. In contrast to one-stage object detectors that generally use a fully convolutional detection head for enhanced processing speed, two-stage counterparts opt for detection heads with fully connected layers to prioritize detection accuracy. In Fast R-CNN [52], a region-based convolutional neural network (R-CNN) (Figure 11 (a)) is designed to classify proposals into one of the classes and better regress the bounding box for the proposal according to the predicted class. R-CNN first takes proposal features generated by RoI pooling as inputs. Two fully connected layers with ReLU activation followed by two parallel fully connected layers are adopted for generating different outputs: a fully connected layer with ($N$+1) units, where $N$ is the total number of classes, is used to produce class scores and a fully connected layer with $4N$ units is used to generate regression offsets for each positive class. R-CNN shares a head for both classification and bounding box regression. We call this detection head as the shared detection head in this paper. R-CNN dominates the field of two-stage object detection frameworks as it has been used in many modern object detectors. Later, R-FCN [56] proposed to replace the fully connected layers in R-CNN by fully convolutional layers to further improve the efficiency of the detection network. R-FCN first generates position-sensitive score maps based on input feature maps and then employs a position-sensitive RoI pooling layer to calculate the class score and box offsets for each proposal. R-FCN achieves competitive accuracy compared with Faster R-CNN while improving inference and training speed. The shared detection head for classification and localization has become a standard component in deep learning-based object detectors and has been leading the fashion of the object detection community in the past years. However, the spatial misalignment between classification and regression branches in the shared detection head limits its detection performance. To be more specific, for each proposal generated by the proposal extraction module, features in salient locations contain rich semantic information which facilitates the classification performance while border features with more location information would improve bounding box regression performance. Thus, with the same spatial box, different optimization targets between classification and regression branches produce low-quality detection results. Recently, various methods have been tried to modify R-CNN architecture in many aspects to obtain better detection performance. These methods can be divided into two groups: methods based on localization sensitive scores and methods based on task-specific structure. The crux of this approach lies in generating a specific score for each proposal based on the intersection-over-union (IoU) between the proposal box and

its corresponding ground truth box. This score essentially acts as a measure of localization accuracy and is used to update the classification scores, thereby improving the accuracy of proposals and preventing accurate proposals from being disregarded in the post-processing stage. By focusing on the spatial alignment of the proposal box with the ground truth, these methods underscore the importance of accurate localization in object detection tasks. Alternatively, task-specific structure methods propose to design different structures for different tasks. These methods directly modify current structures in the classification or regression branch of the shared detection head or design new structures for generating different proposal features for different tasks.

4.2.1 Methods based on localization sensitive scores

Methods based on localization sensitive scores usually adopt IoU values between proposal boxes and the corresponding ground truth boxes to update the classification scores to alleviate the suppression failure due to the misalignment between classification and localization in the shared detection head. For this purpose, IoU-Net [57] (Figure 11 (b)) introduced an IoU predictor branch in the detection head which includes 2 FC layers to produce the IoU values between detected proposal boxes and their corresponding ground truth boxes. The predicted IoU values are used to replace classification score in post-processing stage to accurately suppress abundant proposal boxes. Classification scores of remaining proposals are also updated. Based on the IoU predictor branch, IoU-Net also designed an optimization-based bounding box refinement approach that replaces the traditional regression scheme to improve the localization accuracy. Similar to IoU-Net, Tan et al. [58] proposed a rank-NMS subnetwork (Figure 11 (c)) which can be inserted into the detection head of two-stage object detectors to produce a ranking score for each proposal. Ranking scores are first calculated based on proposals and the corresponding ground truth bounding boxes. Then, ranking scores are fused with classification scores to generate final confident scores which are used as reliable criteria for suppressing abundant bounding boxes in the post-processing stage. The authors also designed a ranking loss to supervise the generation of ranking scores, which encourages candidates with high IoU value to rank higher. To overcome high-quality object detection challenges, Cascade R-CNN [16] proposed a multi-stage object detection pipeline that repeats the detection head with increasing IoU thresholds to produce high-quality detection results (Figure 11 (d)). At each stage of the detection head, R-CNN with a classifier and a regressor is employed to produce predictions based on previous bounding boxes and corresponding IoU thresholds. The IoU threshold is increased through each stage so that the resampling progressively improves bounding box quality, thus producing precise object bounding box at the final detection stage.

4.2.2 Methods based on task-specific structure

Methods based on task-specific structure try to separate classification and regression branches in the detection head to mitigate the potential conflicts between the two subtasks. For this purpose, Grid R-CNN [59] proposed a grid guided mechanism based on fully convolutional network (FCN) [89] for high-quality localization. Based on the shared detection head mechanism in two-stage object detectors, Grid R-CNN replaces the traditional regression branch by a grid prediction branch for localization task (Figure 11 (e)). The grid prediction branch employs a fully convolutional network to produce a probability heatmap which can be used to locate grid points representing object bounding boxes. In addition, precise object bounding boxes are obtained through a feature information fusion based on grid points. Similar to Grid R-CNN, D2Det [55] proposed a two-stage detection pipeline that separates the classification and regression into two branches. D2Det replaces the traditional regression branch by a dense local regression branch which produces multiple offsets for each proposal based on a fully convolutional network (Figure 11 (f)). At each location of the candidate proposal, the dense regression branch produces offsets representing the distances from the location to the top-left and bottom-right corners of the ground-truth bounding box. To alleviate weak box representations due to background features, a binary overlap prediction is designed to eliminate local features that belong to background regions. Recently, Song et al. [60] proposed a task-aware spatial disentanglement module (TSD) to alleviate the inherent misalignment between classification and regression in the shared detection head scheme (Figure 11 (g)). Based on proposals generated by a proposal extraction layer, TSD first employs pointwise deformation and proposal-wise translation mechanism to produce disentangled proposals. Then, classification-specific feature map and localization-specific feature map are generated based on disentangled proposals. Finally, these task-specific feature maps are fed into two parallel branches with three FC layers for classification and regression tasks. By using different feature maps for different tasks, TSD enables each task to adaptively learn the optimal feature without hurting each other, thus improving the overall detection performance. Wu et al. [61] proved that a fully connected head produces a more accurate classification score while a convolution head generates more accurate bounding box regression. Based on this, they proposed a double-head method which splits the detection head into a fully connected head (fc-head) and a convolution head (conv-head) (Figure 11 (h)). The fc-head includes three FC layers for classification task. The conv-head contains $k$ residual blocks followed by an average pooling layer for bounding box regression task. Both heads adopt proposals produced by a RoI Align layer to generate final predictions.

4.2.3 Comparison and performance summary

Table 2 shows the comparison and performance summary. IoU-Net [57] is the first framework that explores the misalignment between the two branches in the shared head structure. By introducing an extra branch in the detection head to predict the IoU values which are used as the localization confidence scores, IoU-Net with ResNet101-FPN improves 2.1 points AP on the MS-COCO minival set compared to Faster R-CNN with the same baseline. Similar to IoU-Net, by adding a rank-NMS subnetwork in the detection head to update classification scores, Learning-to-Rank [58] improves about 3.3 points AP on the Pascal VOC 2007 compared with Faster R-CNN with the same baseline. Compared with Cascade R-CNN [16], Learning-to-Rank achieves a noticeable improvement at strict IoU value. On the other hand, Grid R-CNN [59], which replaces the traditional regression branch by a grid prediction branch for localization task, achieves 3.6 points higher AP than FPN on the Pascal VOC dataset. Recently, D2Det [55] with dense local regression branch outperformed Grid R-CNN by 5.4 points AP on the MS COCO test-dev set. One approach, TSD [60] reported improvements of 13.2 points and 4.9 points AP on the COCO test-dev set

compared with Faster R-CNN and Grid R-CNN with the same FPN baseline.

4.2.4 Open issues

Since IoU-Net explored the shortcomings of R-CNN, various studies have tried to modify the detection head structure in different aspects to improve its performance. While methods based on localization sensitive scores employ IoU values generated by an additional branch to update classification scores, methods based on task-specific structure directly modify current detection head to implement different subtasks. Although these methods achieve certain improvements, we argue that there is a room for developing more profound solutions to alleviate the constraint, such as an appropriate method that combines both localization sensitive scores and task-specific structure.

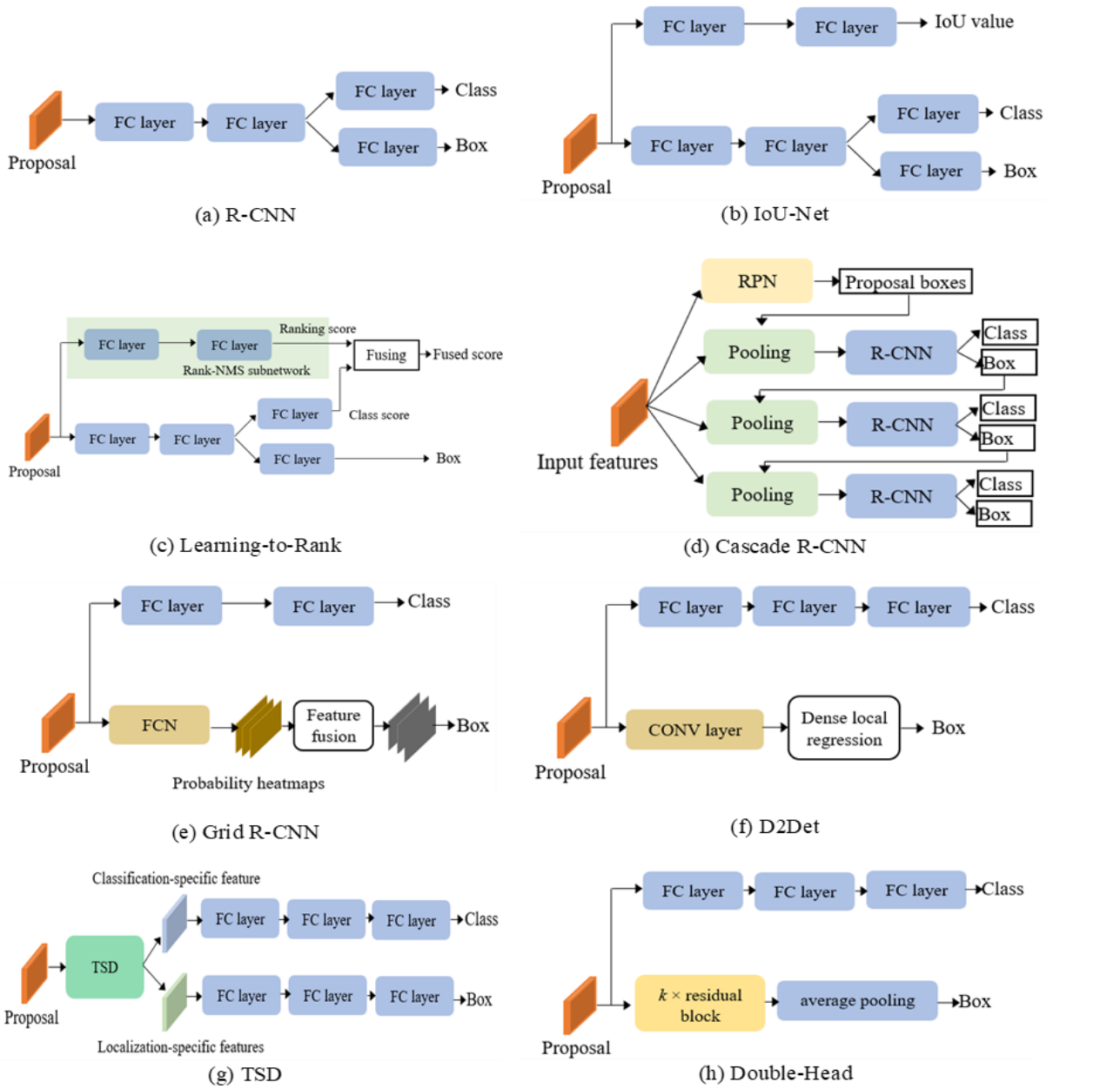



**Figure 11.** The structure of the detection head: (a) R-CNN; (b) IoU-Net; (c) Learning-to-Rank; (d) Cascade R-CNN; (e) Grid R-CNN; (f) D2Det; (g) TSD; (h) Double-Head

**Table 2.** Comparison and performance summary

| Method | Improvement Points (AP) | Benchmark Set | Comparison Baseline |
|---|---|---|---|
| IoU-Net [57] | 2.1 | MS-COCO minival set | Faster R-CNN |
| Learning-to-Rank [58] | 3.3 | Pascal VOC 2007 | Faster R-CNN |
| Grid R-CNN [59] | 3.6 | Pascal VOC 2007 | FPN |
| D2Det [55] | 5.4 | MS-COCO minival set | Grid R-CNN |
| TSD [60] | 13.2 against Faster R-CNN, 4.9 against Grid R-CNN | MS-COCO minival set | Faster R-CNN, Grid R-CNN |

## 5. OTHER PROBLEMS

### 5.1 Recent techniques in post-processing

In two-stage object detection frameworks, anchor box and proposal are fed into the proposal generation and detection network to determine whether the anchor box or proposal is associated to any ground truth class. This mechanism may produce many duplicate predictions since anchor boxes or proposals usually overlap with each other. As a result, a method is required to filter out duplicate predictions. Most early deep object detectors heavily rely on GreedyNMS algorithm as a post-processing stage to remove duplicate results. For example, Faster R-CNN employs GreedyNMS in both proposal generation module and detection head to suppress duplicate results (Figure 12). GreedyNMS starts with a list of proposal boxes $B = \{b_1, b_2, .., b_i\}$ with corresponding scores $S = \{s_1, s_2, .., s_i\}$. A proposal box $b_M$ with the maximum score $M$ is first appended to the set of final proposal box $F$. This proposal box is then removed from $B$. Next, all proposal boxes that overlap with $b_M$ and have IoU values larger than a threshold (e.g., 0.5) are also removed from $B$. This process is repeated until $B$ is empty to generate final proposal boxes $F = \{f_1, f_2, .., f_n\}$. GreedyNMS works well with the assumption that multiple objects rarely occupy the same location in an image. However, in many scenarios, especially in crowded or dense scenes, objects may heavily overlap with each other, thus some objects are very likely to be mistakenly suppressed by GreedyNMS, thus reducing the precision of the network. In some cases, we may increase the overlap threshold to reduce mistakenly suppressed objects. However, this may bring in plenty of false positive results, which also reduces the precision of the network. Recently, various studies have tried to modify GreedyNMS in many aspects to tackle the inherit drawback of GreedyNMS and obtain better detection performance. Instead of removing overlapped bounding box based on IoU threshold, Soft-NMS [62] updates bounding box scores at each iteration based on a continuous penalty function which reduces bounding box scores for high overlapped boxes and maintains bounding box scores for low overlapped boxes. This scheme decays the confident scores of overlapped bounding boxes rather than directly removing them, thus eliminating the mistakenly suppressed problems. Similar to Soft-NMS, IoU-guided NMS [57] proposed to rank detected bounding boxes by the predicted IoU values generated by the IoU predictor branch in the detection head instead of classification scores. After removing abundant proposal boxes, IoU-guided NMS updates classification scores for remaining boxes. IoU-guided NMS resolves the misalignment between classification confidence and localization accuracy since it keeps proposals with high localization accuracy. However, IoU-guided NMS can only be applied in the second stage of two-stage object detectors. Based on GreedyNMS or Soft-NMS, Adaptive-NMS [63] proposed to replace the fixed IoU threshold value by an adaptive threshold value for detecting pedestrians in crowded scenes. With the dynamic threshold value, overlapped bounding boxes are preserved in crowed regions and suppressed in sparse regions. To identify the density of each bounding box, an extra convolutional subnet is designed to produce a density map. The density map is also used to set different IoU threshold values for different bounding boxes. Adaptive-NMS requires an extra structure for predicting density and adaptive threshold, which takes more parameters and computational cost. Different from NMS-based methods, MaxpoolNMS [64] introduced a novel approach to suppress duplicate proposals in the proposal generation stage. The proposed approach is based on a novel multi-scale multi-channel max-pooling strategy which avoids computing the IoU between detected proposals and ground truth boxes, thus significantly improving the speed compared with GreedyNMS. After max-pooling operations, proposals are sorted by their scores, and a set of proposals with high scores are fed into the object prediction stage.

Another line of research is to design a learnable deep network to replace GreedyNMS. In Learning NMS [65], Gnet is designed to replace GreedyNMS to re-score all detected proposals based on proposal boxes and their scores. Gnet includes several blocks which first take features of detections as inputs and then produce updated features based on their neighboring features. The last block in Gnet generates a new detection score for each detection. With repeated blocks in the structure, Gnet performs a rescoring task which decreases the score of proposals that cover object that has been detected already. Similar to Gnet, Relation networks [66] introduced a light-weight relation network to replace the traditional NMS method. In relation to network, duplicate removal can be seen as a two-class classification problem. The network takes a set of detected proposal boxes as inputs and outputs a binary classification score for each proposal box which represents duplicate probability of the proposal. By employing relation module, relation network is an end-to-end learning method with information from different sources such as bounding boxes or classification scores.

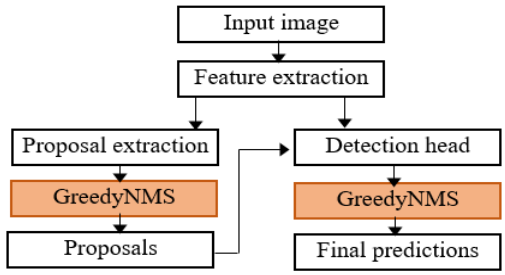



**Figure 12.** Faster R-CNN with GreedyNMS as a post-processing stage

### 5.2 Recent techniques in sampling strategy

During the training process, two-stage object detectors need to define positive and negative samples from the training samples to train each subtask in each stage of the network. The way of defining positive and negative samples is a crucial problem for training the network. Most early deep learning-based object detectors adopt random sampling as a sampling strategy. In Faster R-CNN [30], a mini-batch which consists of 128 positive examples and 128 negative examples is randomly sampled to train RPN. If the number of positive examples is less than 128, the mini batch is padded with random negative examples. Here, an example is considered as positive example if it has the highest IoU with a ground truth box or has an IoU higher than 0.7 with any ground truth box. An example with IoU lower than 0.3 for all ground-truth boxes is assigned as negative example. Although random sampling is simple and easy to implement, it is not an effective strategy to generate training samples since selected samples are easy to

be dominated by easy samples. Recently, it has been found that focusing on hard samples (samples that bring higher loss values) is an effective strategy to improve the training results of object detectors. Based on this idea, OHEM [97] and Focal Loss [6] proposed a new strategy and loss function to focus the training process on hard samples. OHEM and Focal Loss alleviate sampling problems in one-stage object detectors. However, these methods show little improvement when extended to two-stage object detectors since easy samples are filtered by the second stage. In this section, we focus on recent sampling strategies proposed to improve the two-stage object detection training process.

For the purpose of solving sample imbalance problem in training object detectors, Libra R-CNN [25] introduced a novel sampling method based on the IoU values of examples. In the IoU-based sampling method, the sampling interval is first divided into $K$ bins based on IoU values. Then, negative examples are sampled equally within each bin to promote the selected probability of hard negatives which have high IoU values. Compared with random sampling method proposed in Faster R-CNN, IoU-based sampling method improves AP by 0.9 points on the COCO 2017 val split set. Prime Sample Attention (PSA) [68] is an innovative approach to object detection that concentrates the training process on a specific subset of training samples, referred to as prime samples. These prime samples are characterized by their high Intersection over Union (IoU) values for positive samples and large foreground classification scores for negative samples, indicating that they provide substantial information for the detection process. The technical process of generating prime samples involves two distinct ranking systems. The first one, called the IoU-HLR scheme, sorts positive samples based on their IoU values. A high IoU score represents a substantial overlap between the ground truth and the predicted bounding box, indicating a better detection. The second ranking system, Score-HLR scheme, is employed for negative samples. In this scheme, samples are ranked based on their classification scores, where a large score implies that the negative sample has a high likelihood of being misclassified as a foreground object. This score provides a measure of potential confusion or misclassification, which is invaluable for improving the model's precision. Following the ranking process, the top samples in each ranked list, those with the highest IoU or classification scores, are selected as prime samples. To further improve the training process, PSA introduces a soft sampling strategy, which includes positive sample reweighting and negative sample reweighting. The strategy assigns different loss weights to prime samples based on their significance. Higher importance is assigned to those prime samples that are difficult to detect or classify, providing them with more attention during the training process. This approach ensures that the model learns more from challenging examples, thereby enhancing its overall detection accuracy and robustness. Another approach is to automatically produce positive and negative samples based on statistical characteristics of an object. For this purpose, Zhang et al. [69] proposed a novel sampling strategy called adaptive training sample selection (ATSS). The ATSS first defines a set of candidate positive samples for each object based on the center distance between samples and ground truth. The IoU threshold value for the ground truth is obtained based on mean and standard deviation of IoU values between candidate samples and ground truth. Finally, candidates are divided into positive and negative samples based on the IoU threshold value. More recently, motivated by the observation that the problem of sampling inconsistency between the two stages in two-stage object detectors limits their detection performance, SC-RPN [51] designed a size-aware dynamic sampling method to ensure the sampling consistency between the two stages while producing training samples. The Size-Aware Dynamic Sampling (SADS) method proposed in SC-RPN offers a two-fold approach to object detection. It involves two integral modules: a Positive Region Assigner (PRA) and a Size-Aware Threshold Assigner (SATA). PRA employs an anchor-free sampling strategy for the first stage of training. Anchor-free strategies do not predefine a set of anchor boxes, but instead allow the model to predict bounding boxes and their associated class labels directly. This approach tends to provide more flexibility and can help the model to better adapt to the varied scale and aspect ratio of objects. SATA, on the other hand, uses an anchor-based sampling strategy for the second stage. In this context, predefined anchors, typically in various shapes and sizes, serve as references for proposal generation. This anchor-based approach provides a set of starting points for prediction, which can help in tackling complex scenes with overlapping objects. What makes this strategy dynamic is the way it assigns different samples to different feature maps based on their sizes. For instance, smaller objects are assigned to higher resolution feature maps, while larger objects are assigned to lower resolution maps. This size-aware approach ensures that objects of all sizes are effectively detected. Moreover, SATA assigns different overlapping threshold values to different ground truth boxes. Overlapping thresholds help to determine whether a proposal should be considered a positive or negative example during the training process. Adjusting these thresholds based on the size of the ground truth boxes can further improve detection performance. This strategy ensures consistency between the two stages of the training process in terms of the location, size, and quantity of samples. The consistent application of these techniques allows for a more comprehensive and effective learning process.

## 6. CONCLUSIONS

This paper presents an exhaustive review of recent progress in deep learning multi-stage object detection, with a concentrated focus on the evolution of network architecture designs. The analysis reveals a significant shift towards anchor-free detectors and end-to-end trainable frameworks, reflecting the continuous pursuit of more efficient and precise models within the field. This trajectory—tracing from groundbreaking detectors such as R-CNN and Faster R-CNN to contemporary models like RetinaNet—has been meticulously explored. The progressive adaptation and enhancement of the R-CNN architecture by subsequent models were observed, with each iteration addressing specific shortcomings of its precursors. For example, Faster R-CNN enhanced the original R-CNN by integrating a Region Proposal Network (RPN) for proposal region generation, thereby significantly accelerating the detection process. In a similar vein, RetinaNet tackled the class imbalance issue intrinsic to one-stage detectors, leading to improved detection accuracy.

In reviewing a diverse range of optimization strategies for each module of multi-stage object detection frameworks, it was noted that each approach possesses its unique advantages and setbacks. Certain methods offer computational efficiency

but may not attain peak accuracy, while others prioritize accuracy, often resulting in extended training and inference times. Despite these advancements, substantial open challenges persist in object detection, such as efficiently training models on extensive datasets, creating frameworks that can adeptly manage objects of varied scales, and enhancing detection performance for small objects. Although various solutions have been proposed to address these challenges, ample room for enhancement remains.

In the foreseeable future, we anticipate the continued trend towards anchor-free detectors and end-to-end trainable frameworks, as these models have exhibited encouraging results. It is the aspiration that this survey will serve as an insightful reference for researchers and practitioners in the development of innovative object detection models. Moreover, it is hoped that this work will inspire further research on the identified challenges, propelling the boundaries of what is currently achievable in the domain of generic object detection.

## REFERENCES


[1] Zhang, X., Yang, Y.H., Han, Z., Wang, H., Gao, C. (2013). Object class detection: A survey. ACM Computing Surveys (CSUR), 46(1): 1-53. https://doi.org/10.1145/2522968.2522978

[2] Zhu, J. H., Munjal, R., Sivaram, A., Paul, S. R., Tian, J., Jolivet, G. (2022). Flow Regime detection using gamma-ray-based multiphase flowmeter: A machine learning approach. International Journal of Computational Methods and Experimental Measurements, 10(1), 26-37. https://doi.org/10.2495/CMEM-V10-N1-26-37

[3] Bahamon-Blanco, S., Rapp, S., Zhang, Y., Liu, J., Martin, U. (2020). Recognition of track defects through measured acceleration using a recurrent neural network. International Journal of Computational Methods and Experimental Measurements, 8(3), 270-280. https://doi.org/10.2495/CMEM-V8-N3-270-280

[4] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., Murphy, K. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7310-7311. https://doi.org/10.1109/CVPR.2017.351

[5] Lu, X., Li, Q., Li, B., Yan, J. (2020). Mimicdet: Bridging the gap between one-stage and two-stage object detection. In Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23-28, Springer International Publishing, Proceedings, Part XIV 16: 541-557. https://doi.org/10.1007/978-3-030-58568-6_32

[6] Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P. (2017). Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, pp. 2980-2988. https://doi.org/10.1109/TPAMI.2018.2858826

[7] Qin, Z., Li, Z., Zhang, Z., Bao, Y., Yu, G., Peng, Y., Sun, J. (2019). ThunderNet: Towards real-time generic object detection on mobile devices. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6718-6727. https://doi.org/10.1109/ICCV.2019.00682

[8] Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., Pietikäinen, M. (2020). Deep learning for generic object detection: A survey. International Journal of Computer Vision, 128: 261-318. https://doi.org/10.1007/s11263-019-01247-4

[9] Zou, Z., Chen, K., Shi, Z., Guo, Y., Ye, J. (2023). Object detection in 20 years: A survey. Proceedings of the IEEE, 111(3): 257-276. https://doi.org/10.1109/JPROC.2023.3238524

[10] Agarwal, S., Terrail, J.O.D., Jurie, F. (2018). Recent advances in object detection in the age of deep convolutional neural networks. arXiv Preprint arXiv: 1809.03193. https://doi.org/10.48550/arXiv.1809.03193

[11] Cao, J., Pang, Y., Xie, J., Khan, F.S., Shao, L. (2021). From handcrafted to deep features for pedestrian detection: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(9): 4913-4934. https://doi.org/10.1109/TPAMI.2021.3076733

[12] Zafeiriou, S., Zhang, C., Zhang, Z. (2015). A survey on face detection in the wild: Past, present and future. Computer Vision and Image Understanding, 138: 1-24. https://doi.org/10.1016/j.cviu.2015.03.015

[13] Ye, Q., Doermann, D. (2014). Text detection and recognition in imagery: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(7): 1480-1500. https://doi.org/10.1109/TPAMI.2014.2366765

[14] Yang, B., Yan, J., Lei, Z., Li, S.Z. (2016). Craft objects from images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6043-6051. https://doi.org/10.1109/CVPR.2016.650

[15] Vu, T., Jang, H., Pham, T.X., Yoo, C. (2019). Cascade rpn: Delving into high-quality region proposal network with adaptive convolution. Advances in Neural Information Processing Systems, 32.

[16] Cai, Z., Vasconcelos, N. (2019). Cascade R-CNN: High quality object detection and instance segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(5): 1483-1498. https://doi.org/10.1109/TPAMI.2019.2956516

[17] Gidaris, S., Komodakis, N. (2016). Locnet: Improving localization accuracy for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 789-798. https://doi.org/10.1109/CVPR.2016.92

[18] Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv Preprint arXiv: 1409.1556. https://doi.org/10.48550/arXiv.1409.1556

[19] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778. https://doi.org/10.1109/CVPR.2016.90

[20] Cai, Z., Fan, Q., Feris, R.S., Vasconcelos, N. (2016). A unified multi-scale deep convolutional neural network for fast object detection. In Computer Vision-ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, Springer International Publishing, Proceedings, Part IV 14: 354-370. https://doi.org/10.1007/978-3-319-46493-0_22

[21] Li, J., Liang, X., Shen, S., Xu, T., Feng, J., Yan, S. (2017). Scale-aware fast R-CNN for pedestrian detection. IEEE Transactions on Multimedia, 20(4): 985-996. https://doi.org/10.1109/TMM.2017.2759508

[22] 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. https://doi.org/10.1109/CVPR.2016.234

[23] Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117-2125. https://doi.org/10.1109/CVPR.2017.106

[24] Liu, S., Qi, L., Qin, H., Shi, J., Jia, J. (2018). Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8759-8768. https://doi.org/10.1109/CVPR.2018.00913

[25] Pang, J., Chen, K., Shi, J., Feng, H., Ouyang, W., Lin, D. (2019). Libra R-CNN: Towards balanced learning for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 821-830. https://doi.org/10.1109/CVPR.2019.00091

[26] Guo, C., Fan, B., Zhang, Q., Xiang, S., Pan, C. (2020). AUGFPN: Improving multi-scale feature learning for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12595-12604. https://doi.org/10.1109/CVPR42600.2020.01261

[27] Tan, M., Pang, R., Le, Q.V. (2020). Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10781-10790. https://doi.org/10.1109/CVPR42600.2020.01079

[28] Hu, M., Li, Y., Fang, L., Wang, S. (2021). A2-FPN: Attention aggregation based feature pyramid network for instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15343-15352. https://doi.org/10.1109/CVPR46437.2021.01509

[29] Qiao, S., Chen, L.C., Yuille, A. (2021). Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10213-10224. https://doi.org/10.1109/CVPR46437.2021.01008

[30] Ren, S., He, K., Girshick, R., Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems, 28. https://doi.org/10.1109/TPAMI.2016.2577031

[31] Yang, T., Zhang, X., Li, Z., Zhang, W., Sun, J. (2018). Metaanchor: Learning to detect objects with customized anchors. Advances in Neural Information Processing Systems, 31.

[32] Zhong, Y., Wang, J., Peng, J., Zhang, L. (2020). Anchor box optimization for object detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1286-1294. https://doi.org/10.1109/WACV45572.2020.9093498

[33] Sun, P., Zhang, R., Jiang, Y., Kong, T., Xu, C., Zhan, W., Tomizuka, M., Li, L., Yuan, Z., Wang, C., Luo, P. (2021). Sparse r-cnn: End-to-end object detection with learnable proposals. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14454-14463. https://doi.org/10.1109/CVPR46437.2021.01422

[34] Huang, L., Yang, Y., Deng, Y., Yu, Y. (2015). Densebox: Unifying landmark localization with end to end object detection. arXiv Preprint arXiv: 1509.04874. https://doi.org/10.48550/arXiv.1509.04874

[35] Zhu, C., He, Y., Savvides, M. (2019). Feature selective anchor-free module for single-shot object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 840-849. https://doi.org/10.1109/CVPR.2019.00093

[36] Wang, J., Chen, K., Yang, S., Loy, C.C., Lin, D. (2019). Region proposal by guided anchoring. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2965-2974. https://doi.org/10.1109/CVPR.2019.00308

[37] Tian, Z., Shen, C., Chen, H., He, T. (2019). Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9627-9636. https://doi.org/10.1109/ICCV.2019.00972

[38] Kong, T., Sun, F., Liu, H., Jiang, Y., Li, L., Shi, J. (2020). Foveabox: Beyound anchor-based object detection. IEEE Transactions on Image Processing, 29: 7389-7398. https://doi.org/10.1109/TIP.2020.3002345

[39] Zhu, C., Chen, F., Shen, Z., Savvides, M. (2020). Soft anchor-point object detection. In Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23-28, Springer International Publishing, Proceedings, Part IX 16: 91-107. https://doi.org/10.1007/978-3-030-58545-7_6

[40] Law, H., Deng, J. (2018). Cornernet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 734-750. https://doi.org/10.1007/978-3-030-01264-9_45

[41] Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q. (2019). Centernet: Keypoint triplets for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6569-6578. https://doi.org/10.1109/ICCV.2019.00667

[42] Zhou, X., Zhuo, J., Krahenbuhl, P. (2019). Bottom-up object detection by grouping extreme and center points. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 850-859. https://doi.org/10.1109/CVPR.2019.00094

[43] Zhou, X., Wang, D., Krähenbühl, P. (2019). Objects as points. arXiv Preprint arXiv: 1904.07850. https://doi.org/10.48550/arXiv.1904.07850

[44] Yang, Z., Liu, S., Hu, H., Wang, L., Lin, S. (2019). Reppoints: Point set representation for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9657-9666. https://doi.org/10.1109/ICCV.2019.00975

[45] Zhang, H., Wang, Y., Dayoub, F., Sunderhauf, N. (2021). Varifocalnet: An iou-aware dense object detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8514-8523. https://doi.org/10.1109/CVPR46437.2021.00841

[46] Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W. (2013). Selective search for object recognition. International Journal of Computer Vision, 104: 154-171. https://doi.org/10.1007/s11263-013-0620-5

[47] Zitnick, C.L., Dollár, P. (2014). Edge boxes: Locating object proposals from edges. In Computer Vision-ECCV 2014: 13th European Conference, Zurich, Switzerland,

September 6-12, Springer International Publishing, Proceedings, Part V 13: 391-405. https://doi.org/10.1007/978-3-319-10602-1_26
[48] Zhong, Q., Li, C., Zhang, Y., Xie, D., Yang, S., Pu, S. (2020). Cascade region proposal and global context for deep object detection. Neurocomputing, 395: 170-177. https://doi.org/10.1016/j.neucom.2017.12.070
[49] Qiu, H., Ma, Y., Li, Z., Liu, S., Sun, J. (2020). Borderdet: Border feature for dense object detection. In Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23-28, Springer International Publishing, Proceedings, Part I 16: 549-564. https://doi.org/10.1007/978-3-030-58452-8_32
[50] Duan, K., Xie, L., Qi, H., Bai, S., Huang, Q., Tian, Q. (2020). Corner proposal network for anchor-free, two-stage object detection. In European Conference on Computer Vision. Cham: Springer International Publishing, pp. 399-416. https://doi.org/10.1007/978-3-030-58580-8_24
[51] Zou, W., Zhang, Z., Peng, Y., Xiang, C., Tian, S., Zhang, L. (2021). SC-RPN: A strong correlation learning framework for region proposal. IEEE Transactions on Image Processing, 30: 4084-4098. https://doi.org/10.1109/TIP.2021.3069547
[52] Girshick, R. (2015). Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, pp. 1440-1448. https://doi.org/10.1109/ICCV.2015.169
[53] He, K., Gkioxari, G., Dollár, P., Girshick, R. (2017). Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, pp. 2961-2969. https://doi.org/10.1109/TPAMI.2018.2844175
[54] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y. (2017). Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, pp. 764-773. https://doi.org/10.1109/ICCV.2017.89
[55] Cao, J., Cholakkal, H., Anwer, R.M., Khan, F.S., Pang, Y., Shao, L. (2020). D2det: Towards high quality object detection and instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11485-11494. https://doi.org/10.1109/CVPR42600.2020.01150
[56] Dai, J., Li, Y., He, K., Sun, J. (2016). R-FCN: Object detection via region-based fully convolutional networks. Advances in Neural Information Processing Systems, 29.
[57] Jiang, B., Luo, R., Mao, J., Xiao, T., Jiang, Y. (2018). Acquisition of localization confidence for accurate object detection. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 784-799. https://doi.org/10.1007/978-3-030-01264-9_48
[58] Tan, Z., Nie, X., Qian, Q., Li, N., Li, H. (2019). Learning to rank proposals for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8273-8281. https://doi.org/10.1109/ICCV.2019.00836
[59] Lu, X., Li, B., Yue, Y., Li, Q., Yan, J. (2019). Grid R-CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7363-7372. https://doi.org/10.1109/CVPR.2019.00754
[60] Song, G., Liu, Y., Wang, X. (2020). Revisiting the sibling head in object detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11563-11572. https://doi.org/10.1109/CVPR42600.2020.01158
[61] Wu, Y., Chen, Y., Yuan, L., Liu, Z., Wang, L., Li, H., Fu, Y. (2020). Rethinking classification and localization for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10186-10195. https://doi.org/10.1109/CVPR42600.2020.01020
[62] Bodla, N., Singh, B., Chellappa, R., Davis, L.S. (2017). Soft-NMS-improving object detection with one line of code. In Proceedings of the IEEE International Conference on Computer Vision, pp. 5561-5569. https://doi.org/10.1109/ICCV.2017.593
[63] Liu, S., Huang, D., Wang, Y. (2019). Adaptive NMS: Refining pedestrian detection in a crowd. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6459-6468. https://doi.org/10.1109/CVPR.2019.00662
[64] Cai, L., Zhao, B., Wang, Z., Lin, J., Foo, C.S., Aly, M.S., Chandrasekhar, V. (2019). Maxpoolnms: getting rid of nms bottlenecks in two-stage object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9356-9364. https://doi.org/10.1109/CVPR.2019.00958
[65] Hosang, J., Benenson, R., Schiele, B. (2017). Learning non-maximum suppression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4507-4515. https://doi.org/10.1109/CVPR.2017.685
[66] Hu, H., Gu, J., Zhang, Z., Dai, J., Wei, Y. (2018). Relation networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3588-3597. https://doi.org/10.1109/CVPR.2018.00378
[67] Lee, Y., Park, J. (2020). Centermask: Real-time anchor-free instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13906-13915. https://doi.org/10.1109/CVPR42600.2020.01392
[68] Cao, Y., Chen, K., Loy, C.C., Lin, D. (2020). Prime sample attention in object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11583-11591. https://doi.org/10.1109/CVPR42600.2020.01160
[69] Zhang, S., Chi, C., Yao, Y., Lei, Z., Li, S.Z. (2020). Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9759-9768. https://doi.org/10.1109/CVPR42600.2020.00978
[70] Krizhevsky, A., Sutskever, I., Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 25. https://doi.org/10.1145/3065386
[71] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510-4520. https://doi.org/10.1109/CVPR.2018.00474
[72] Ma, N., Zhang, X., Zheng, H.T., Sun, J. (2018). Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 116-131. https://doi.org/10.1007/978-3-030-01264-9_8
[73] Mehta, S., Rastegari, M., Shapiro, L., Hajishirzi, H. (2019). Espnetv2: A light-weight, power efficient, and

general purpose convolutional neural network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9190-9200. https://doi.org/10.1109/CVPR.2019.00941

[74] Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., Sun, J. (2018). Detnet: A backbone network for object detection. arXiv Preprint arXiv: 1804.06215. https://doi.org/10.48550/arXiv.1804.06215

[75] Fu, C.Y., Liu, W., Ranga, A., Tyagi, A., Berg, A.C. (2017). Dssd: Deconvolutional single shot detector. arXiv Preprint arXiv: 1701.06659. https://doi.org/10.48550/arXiv.1701.06659

[76] Gong, Y., Yu, X., Ding, Y., Peng, X., Zhao, J., Han, Z. (2021). Effective fusion factor in FPN for tiny object detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1160-1168. https://doi.org/10.1109/WACV48630.2021.00120

[77] Zhu, B., Song, Q., Yang, L., Wang, Z., Liu, C., Hu, M. (2021). CPM R-CNN: Calibrating point-guided misalignment in object detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 3248-3257. https://doi.org/10.1109/WACV48630.2021.00329

[78] Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J. (2017). Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2881-2890. https://doi.org/10.1109/CVPR.2017.660

[79] Tan, M., Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In International Conference on Machine Learning, PMLR, pp. 6105-6114.

[80] Wang, J., Chen, K., Xu, R., Liu, Z., Loy, C.C., Lin, D. (2019). Carafe: Content-aware reassembly of features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3007-3016. https://doi.org/10.1109/ICCV.2019.00310

[81] Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(4): 834-848. https://doi.org/10.1109/TPAMI.2017.2699184

[82] Zoph, B., Le, Q.V. (2016). Neural architecture search with reinforcement learning. arXiv Preprint arXiv: 1611.01578. https://doi.org/10.48550/arXiv.1611.01578

[83] Ghiasi, G., Lin, T.Y., Le, Q.V. (2019). Nas-fpn: Learning scalable feature pyramid architecture for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7036-7045. https://doi.org/10.1109/CVPR.2019.00720

[84] Tychsen-Smith, L., Petersson, L. (2017). Denet: Scalable real-time object detection with directed sparse sampling. In Proceedings of the IEEE International Conference on Computer Vision, pp. 428-436. https://doi.org/10.1109/ICCV.2017.54

[85] Dai, X., Chen, Y., Xiao, B., Chen, D., Liu, M., Yuan, L., Zhang, L. (2021). Dynamic head: Unifying object detection heads with attentions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7373-7382. https://doi.org/10.1109/CVPR46437.2021.00729

[86] Liao, M., Shi, B., Bai, X. (2018). Textboxes++: A single-shot oriented scene text detector. IEEE Transactions on Image Processing, 27(8): 3676-3690. https://doi.org/10.1109/TIP.2018.2825107

[87] Li, H., Wang, P., Shen, C. (2018). Toward end-to-end car license plate detection and recognition with deep neural networks. IEEE Transactions on Intelligent Transportation Systems, 20(3): 1126-1136. https://doi.org/10.1109/TITS.2018.2847291

[88] Wang, H., Li, Y., Wang, S. (2019). Fast pedestrian detection with attention-enhanced multi-scale RPN and soft-cascaded decision trees. IEEE Transactions on Intelligent Transportation Systems, 21(12): 5086-5093. https://doi.org/10.1109/TITS.2019.2948398

[89] 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. https://doi.org/10.1109/CVPR.2015.7298965

[90] Lan, S., Ren, Z., Wu, Y., Davis, L.S., Hua, G. (2020). Saccadenet: A fast and accurate object detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10397-10406. https://doi.org/10.1109/CVPR42600.2020.01041

[91] Chu, W., Liu, Y., Shen, C., Cai, D., Hua, X.S. (2017). Multi-task vehicle detection with region-of-interest voting. IEEE Transactions on Image Processing, 27(1): 432-441. https://doi.org/10.1109/TIP.2017.2762591

[92] Li, J., Wang, Z. (2018). Real-time traffic sign recognition based on efficient CNNs in the wild. IEEE Transactions on Intelligent Transportation Systems, 20(3): 975-984. https://doi.org/10.1109/TITS.2018.2843815

[93] Wu, J., Zhou, C., Yang, M., Zhang, Q., Li, Y., Yuan, J. (2020). Temporal-context enhanced detection of heavily occluded pedestrians. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13430-13439. https://doi.org/10.1109/CVPR42600.2020.01344

[94] Chen, Z.M., Jin, X., Zhao, B., Wei, X.S., Guo, Y. (2020). Hierarchical context embedding for region-based object detection. In Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23-28, 2020, Springer International Publishing, Proceedings, Part XXI 16: 633-648. https://doi.org/10.1007/978-3-030-58589-1_38

[95] Gkioxari, G., Malik, J., Johnson, J. (2019). Mesh R-CNN. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9785-9795. https://doi.org/10.1109/ICCV.2019.00988

[96] Zhu, X., Hu, H., Lin, S., Dai, J. (2019). Deformable convnets v2: More deformable, better results. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9308-9316. https://doi.org/10.1109/CVPR.2019.00953

[97] 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. https://doi.org/10.1109/CVPR.2016.89