Electronic Research Archive

https://www.aimspress.com/journal/era

ERA, 34(4): 2674–2731.
DOI: 10.3934/era.2026124
Received: 07 November 2025
Revised: 06 February 2026
Accepted: 06 March 2026
Published: 30 March 2026

Review

Deep learning-based object detection: A comprehensive review of YOLO, RCNN, and SSD series

Wu Zeng[1], **Guojun Mao**[2,*], **Mei Li**[1,*] **and Shuaibing Yin**[1]

[1] School of Artificial Intelligence, China University of Geosciences (Beijing), Beijing 100083, China
[2] Faculty of Intelligent Transportation, Anhui Sanlian University, Hefei 230601, China

* **Correspondence:** Email: maximmao@hotmail.com, maggieli@cugb.edu.cn.

Abstract: In recent years, with the development of science and technology, deep learning technology has also been continuously advancing. As an important application field of deep learning, computer vision has increasingly broad application prospects. Among them, object detection is an extremely important branch of computer vision. This technology has been widely applied in many fields such as environmental monitoring, traffic management, and agricultural evaluation. This paper focused on introducing deep learning-based computer vision object detection and small object detection technologies. In general, object detection methods can be divided into two major categories, namely one-stage and two-stage object detection algorithms. In further subdivision, we roughly classified object detection technologies into three categories: 1) object detection frameworks based on the you only look once (YOLO) series; 2) object detection frameworks based on the region-based convolutional neural network (R-CNN) series; 3) object detection frameworks based on the SSD (single shot multibox detector) series. In addition, we also introduced a series of real application scenarios of small object detection algorithms, such as small object detection based on unmanned aerial vehicles (UAVs) and remote sensing images. Finally, we summarized object detection and small object detection, and we look forward to some possible future research or improvement directions of this technology.

Keywords: object detection; small object detection; unmanned aerial vehicles (UAVs); remote sensing image detection; deep learning

1. Introduction

In recent years, with the continuous development of science and technology, deep learning (DL) technology has made rapid progress in many fields of computer vision, for example, object detection [1–3], generative adversarial networks [4–6], image classification [7–9], sentiment analysis [10–12], natural language processing [13–15], image segmentation [16–18], long-tailed

visual recognition [19–22], etc. The development of these technologies has greatly facilitated our daily life. Among them, as one of the important branches of computer vision, object detection technology not only plays a crucial role, but also serves as an important prerequisite technology for subsequent applications in some practical scenarios. In general, object detection technology can accurately identify and locate target objects in images, providing key support for the analysis and decision-making of subsequent tasks. For example, this technology can provide key support for downstream tasks such as intelligent monitoring [23] and automatic driving [24].

In the research of object detection, small object detection is a challenging subtopic. Due to the small proportion of pixels of small objects in the image and the sparse feature information contained, they are vulnerable to background noise interference and other factors. Therefore, their detection accuracy is usually lower than that of objects of conventional scale. Then it is difficult to achieve ideal detection results using the traditional object detection model. These problems limit the application of object detection technology in some large detection range and small object scenes in real life.

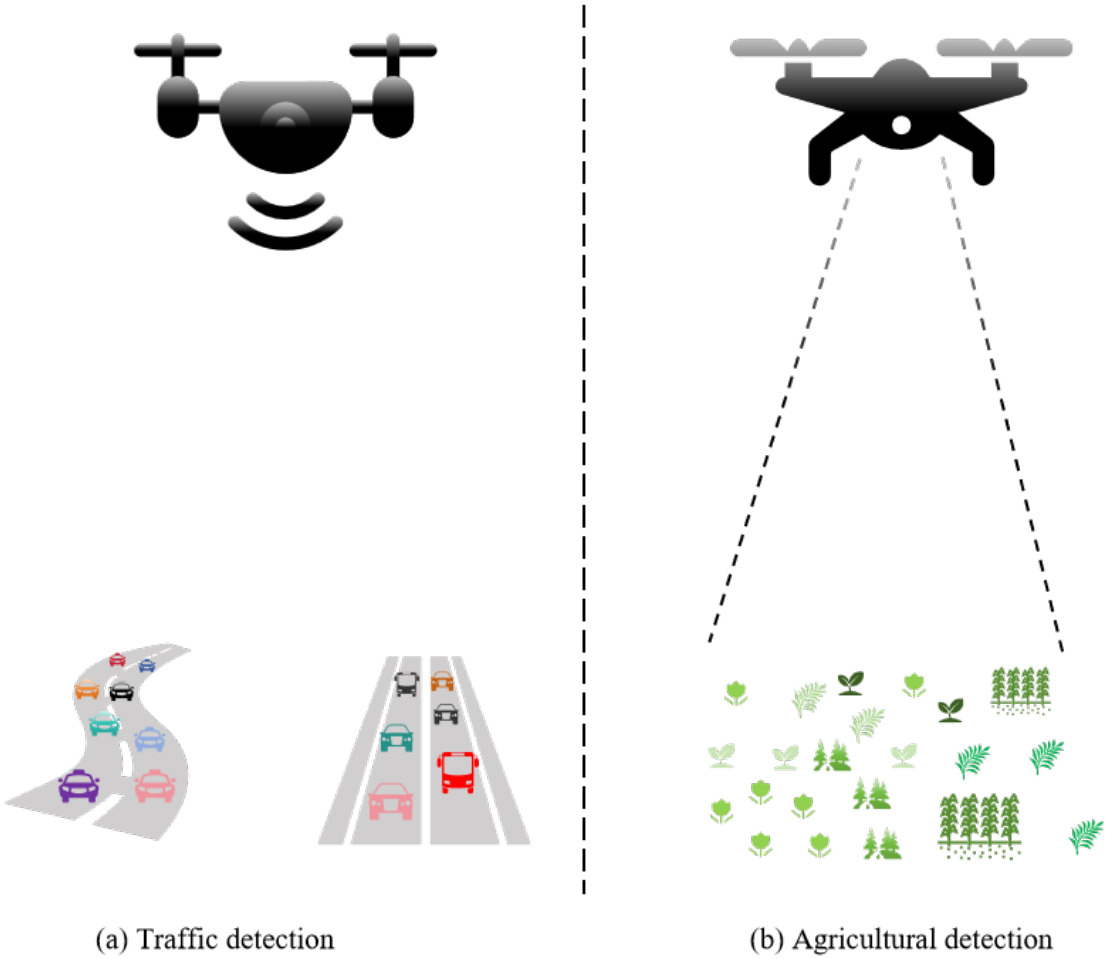
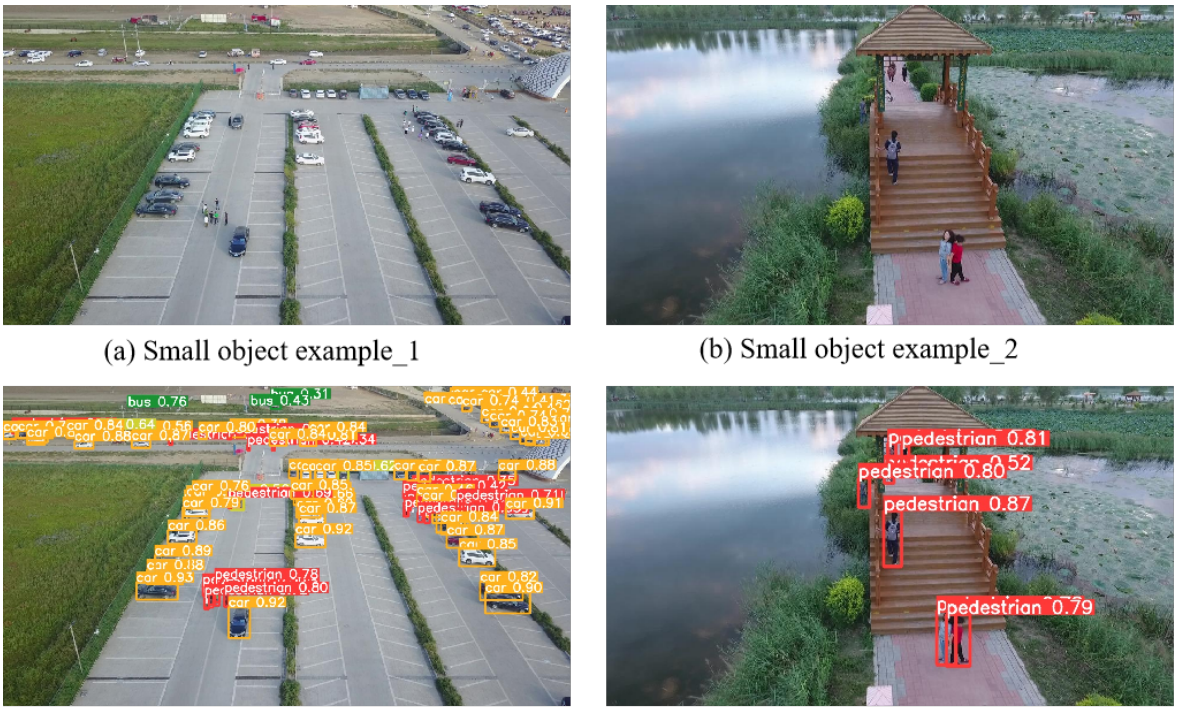


Figure 1. Examples of simulation scenarios where UAVs conduct road traffic detection and agricultural detection in the air.

At the same time, it is worth noting the development of unmanned aerial vehicles (UAVs) technology [25–27]. With its flexible mobility and large detection range, UAVs have become important tools in monitoring [28], patrol [29], rescue [30] and other use scenarios. However, it should also be noted that in the shooting scene of the UAV, the UAV is often at a high altitude and far away from the target object, so the target object often presents the characteristics of "small-scale" (as shown in the simulation scene of traffic detection and agricultural detection in Figure 1, the scale of vehicles and plants is very small in the perspective of the UAV and some equipment in the high altitude). This also makes small object detection under platforms such as UAVs and some equipment in high altitude become a research field with both theory and practical applications. For example, in traffic management, it is possible to detect small mobile vehicles on the road in a timely manner to ensure smooth traffic flow. In agricultural prevention and control, early inspection and testing of crops can effectively prevent large-scale outbreaks of pests and diseases. In wildlife conservation, it is possible to detect some protected animals in a wide environment to achieve precise protection of them. It is precisely because this technology (UAV+small object detection) has so many applications in real life that this technology has been widely pursued and studied.



(a) Small object example_1 (b) Small object example_2

(c) Small object detection example_1 (d) Small object detection example_2

Figure 2. Small object from the perspective of UAVs and examples of small object detection results.

After so many years of development, thanks to the continuous efforts of many scientific researchers, object detection technology with increasingly powerful performance has been constantly introduced (specifically, the performance of the model in the microsoft common objects in context (MS COCO) and pattern analysis, statistical modeling and computational learning visual object classes (PASCAL VOC) datasets has been continuously improved). In summary, there are roughly

two factors that contribute to success. First, these datasets contain a large number of target object area pixels related to various categories. A large number of target objects can help the model learn the visual presentation of target objects in different situations (such as different poses, backgrounds, sizes, etc.), thereby helping the model to learn and understand the feature information of target objects more deeply. Second, most of these datasets consist of images with a relatively large proportion of target size (usually only containing pixel information of one or a few target objects in one image). Therefore, the detection model can learn the features of the target object in the image more clearly.

However, in the field of small object detection, the perspective of equipment is often high. Therefore, the objects will show a "large number and small pixels" distribution (Figure 2 shows the small object image). The pixel value of a small object is usually less than 32×32, so this object is also called a small object [31]. Because the pixel size of the object is small, some feature information of these objects will not be clear enough. This further prevents the detection model from further learning the characteristics of these objects. At the same time, the pixel size of the object is too small, which will also make it blurred. This also makes it difficult for the detection model to learn the distinguishing feature information content among various categories of objects when learning them. Thus, it is difficult for the model to accurately locate and identify small objects. So there is also a problem: although most detection models have achieved good detection performance, this problem exists when shooting from a high altitude perspective. This leads to the problem of poor performance of these detection models when detecting small objects in the high altitude horizon. For these reasons, when dealing with the problem of small object detection, some models directly using general object detection methods may not achieve ideal detection results. Sometimes it is necessary to improve the model according to the specific use environment. Generally speaking, small object detection is an important branch of object detection. It is important in both academic theory and practical application. Therefore, this paper focuses on object detection and small object detection.

Object detection is an important and popular branch in the field of computer vision, and has therefore received extensive research. Many excellent reviews have emerged one after another. Zhao et al. [32] focused on the development of object detection technology before 2018 in their review. This article systematically reviews the application of deep learning in object detection and the performance of different methods on multiple datasets. The review article by Zou et al. [33] mainly introduces the development of object detection technology in the past 20 years, with a focus on some traditional and deep learning based object detection methods in 2022. The review article by Amjoud et al. [34] focuses on the articles on object detection from 2013 to 2022. At the same time, prospects were made for some possible research directions in the future. In terms of small object detection, Liu et al. [35] focused on the performance of some mainstream algorithms in small object detection, as well as their technical relevance in different research fields. The review by Wei et al. [36] analyzed some of the challenges and implementation difficulties faced by small object detection algorithms. Then, some practical application areas of small object detection algorithms were introduced.

2. The survey approach of this review

It should be noted that general object detection is usually focused on objects of conventional scale (usually 32×32 pixels or more). Small object detection focuses on small objects with very low pixel ratio (usually less than 32×32 pixels), which is more difficult.

In general, most reviews either focus on object detection alone or introduce small object detection in a relatively single way, and few of them combine the two. Different from this, this paper takes object detection as the entry point, and then extends to small object detection. Not only can readers understand the development of object detection, but also can readers understand some applications of these key technologies in important small object detection fields. We divide object detection methods into two categories: 1) one-stage object detection methods. We subdivide it into one-stage object detection methods represented by you only look once (YOLO) series and single shot multibox detector (SSD) series; 2) two-stage object detection method (object detection method represented by region-based convolutional neural network (R-CNN) series). On this basis, we introduce the small object detection, especially the small object detection under the high-altitude perspective (specifically the small object detection based on the UAV perspective, and the small object detection task of remote sensing images). Due to the importance of small object detection technology based on high-altitude perspective equipment for some current production work, we specially investigated the small object detection work based on this situation. In addition, we use a progressive approach to introduce, so as to facilitate better explanation.

The general structure of this overview is shown in Figure 3. The main arrangements for the rest are as follows:

In section 1: we briefly introduce the development, practical application and some existing difficulties of object detection and small object detection technology.

In section 2: the research methods and overall structure of this review are introduced.

In section 3: in this section, we first introduce some famous and classic object detection data sets (such as MS COCO [37] and PASCAL VOC [38]), providing basic support for technical analysis and experimental comparison in subsequent chapters. On this basis, the data sets (such as VisDrone [39] and DOTA (dataset for object detection in aerial images) [40]) frequently used in small object detection tasks based on high-altitude perspective are introduced. Finally, we introduce some important performance indicators of object detection and some related basic knowledge.

In section 4: as the key content of this paper, we systematically introduce the basic framework of object detection, which is the technical cornerstone of small object detection in subsequent sub-fields. We first introduced the one-stage object detection framework, namely YOLO series, SSD series and other strategies.

In section 5: in this section, we mainly introduce the two-stage object detection framework, namely R-CNN series object detection methods.

In section 6: we first focus on the small object detection methods of YOLO, SSD and R-CNN series. Then the small object detection method based on other strategies is supplemented.

In section 7: this section shows the performance of some classic and advanced object detection and small object detection methods in relevant data sets.

In section 8: we make a comprehensive summary of object detection and small object detection methods, respectively. It not only combs the achievements in this field, but also analyzes their existing shortcomings and possible future improvement directions.

In section 9: in the last part of this paper, we summarize the full paper.

In addition, the general strategies of the papers selected in this review are as follows:

1) Time span: our selected papers are mainly from 2015 to 2024. These papers are the current research hotspot and have been verified by a certain amount of time. Other papers are as early as 2015.

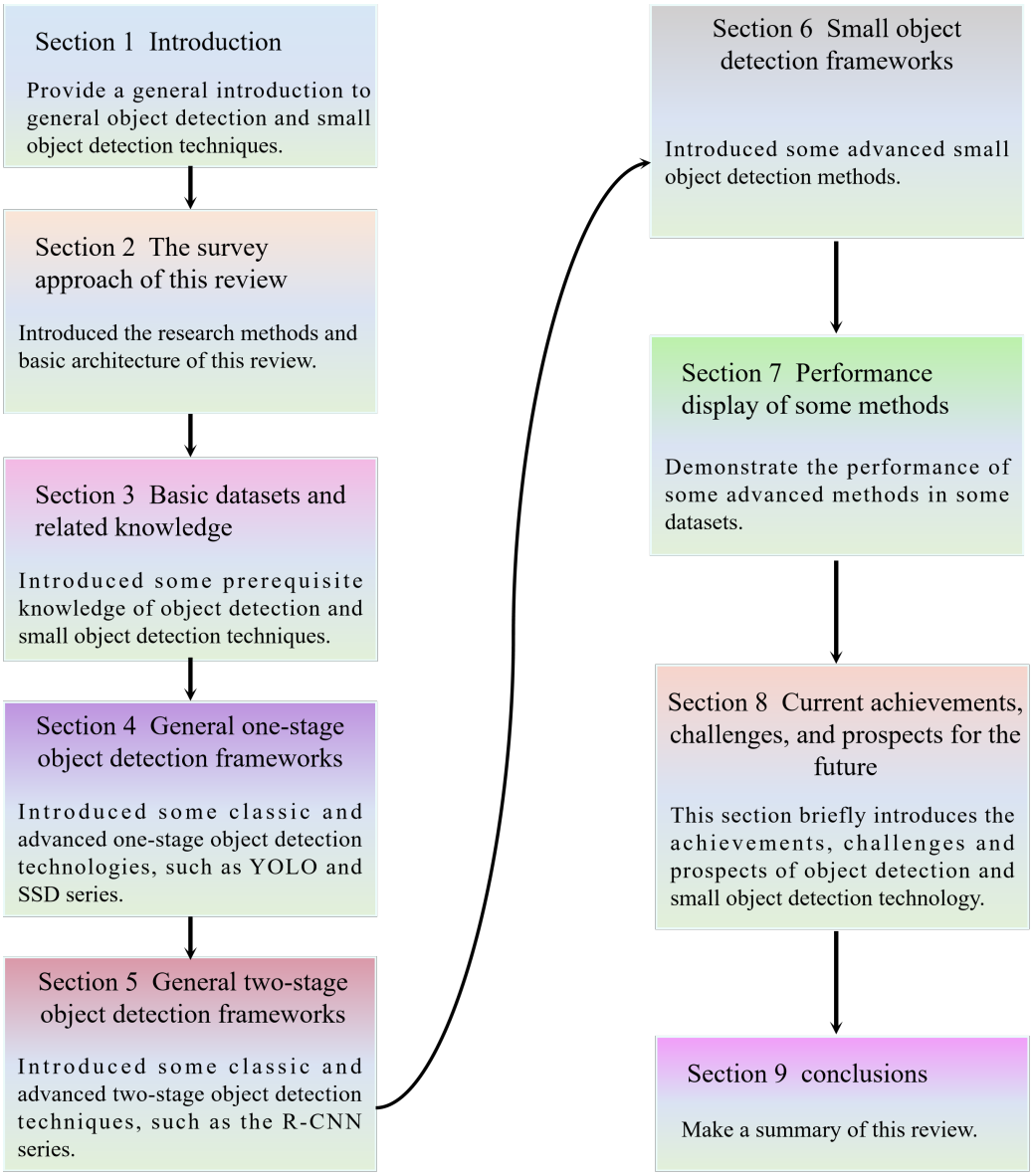


Figure 3. Schematic diagram of the structure of each section in this review.

Most of these papers are based on core technologies, which provide an important foreshadowing for further research of follow-up methods. In addition, another part of the papers are newer papers after 2024 to ensure the cutting-edge and timeliness of the review.

2) Search database: the articles we selected are mainly from the Engineering Village (EI) database, most of which are high-level achievements included in some top conferences in the computer field (such as conference on computer vision and pattern recognition (CVPR), international conference on computer vision (ICCV), european conference on computer vision (ECCV), etc.). In addition, some are from the Web of Science (WOS) database. At the same time, there are also some excellent achievements from arXiv preprint and some other excellent databases.

3) Literature screening strategy: we mainly use multi-keyword combination, reading recommendation of main team members, and core reference tracking. Multi-keyword combination is

mainly divided into deep learning, object detection, small object detection, UAV, etc. The literature recommended by the main members of the team mainly includes the literature focusing on deep learning, object detection and small object detection accumulated in the daily reading process, as well as some inspiring papers. There are also some core references, which mainly include the extremely important core foundational papers in some fields, as well as the subsequent expanded research around these core achievements and the supplement of key technologies.

3. Basic datasets and related knowledge

For the convenience of subsequent introductions, in this section we will first introduce some basic prerequisite knowledge, commonly used datasets, and some performance indicators.

3.1. Basic prerequisite knowledge

The birth of object detection technology is not achieved overnight, but is built on the accumulation of a series of deep learning pre-technologies. Among them, a series of breakthrough advances in convolutional neural networks (CNNs) have provided strong foundational support for object detection technology. AlexNet was proposed in 2012 and demonstrated powerful feature extraction capabilities. From then on, object detection technology ushered in the era of deep learning. The CNN module can be considered as consisting of four parts: 1) convolutional layer; 2) activation layer; 3) pooling layer; 4) fully connected layer. In addition, classic CNNs include LeNet [41], visual geometry group network (VGGNet) [42], residual network (ResNet) [43], MobileNet [44]. Some advanced networks are often used as backbone networks in object detection technology for detailed feature extraction. We provide a schematic diagram of the partial network architecture in Figure 4. Furthermore, we will briefly introduce the four core modules of CNN in the following.

Performing convolution operations on image samples is an important step. By performing regular sliding operations in the image, local features in the image can be obtained. Overall, convolutional neural networks have many advantages in image processing, such as sparse connections, parameter sharing, equivariant representation, and so on. Sparse connections can enable neurons to be connected only to local regions, achieving the goal of reducing redundant computation. By using parameter sharing, the parameters of the same set of convolution kernels can be reused in different regions of the image, thereby reducing the complexity of the model. At the same time, using equivariant representation can improve the stability of the model in handling images with geometric transformations. In addition, key parameters in the convolutional kernel, such as kernel size, sliding step size, and boundary filling, need to be manually preset. The size of the convolution kernel determines the receptive field of the model. The size of the stride determines the sliding distance of the convolution kernel during each action, thereby affecting the size of the output feature image. Boundary filling will fill the edges of the feature image, which can prevent the feature image from becoming smaller as convolution continues.

The activation function layer, also known as the non-linear mapping layer, is introduced to enhance the feature representation ability of the network model. Simple and strategy free linear stacking can only achieve simple linear mapping, making it difficult to further handle complex tasks. Therefore, it is necessary to continue adding activation functions after the positive layer. Some common activation functions include ReLU (rectified linear unit) [45], Mish [46], and leaky ReLU [47].

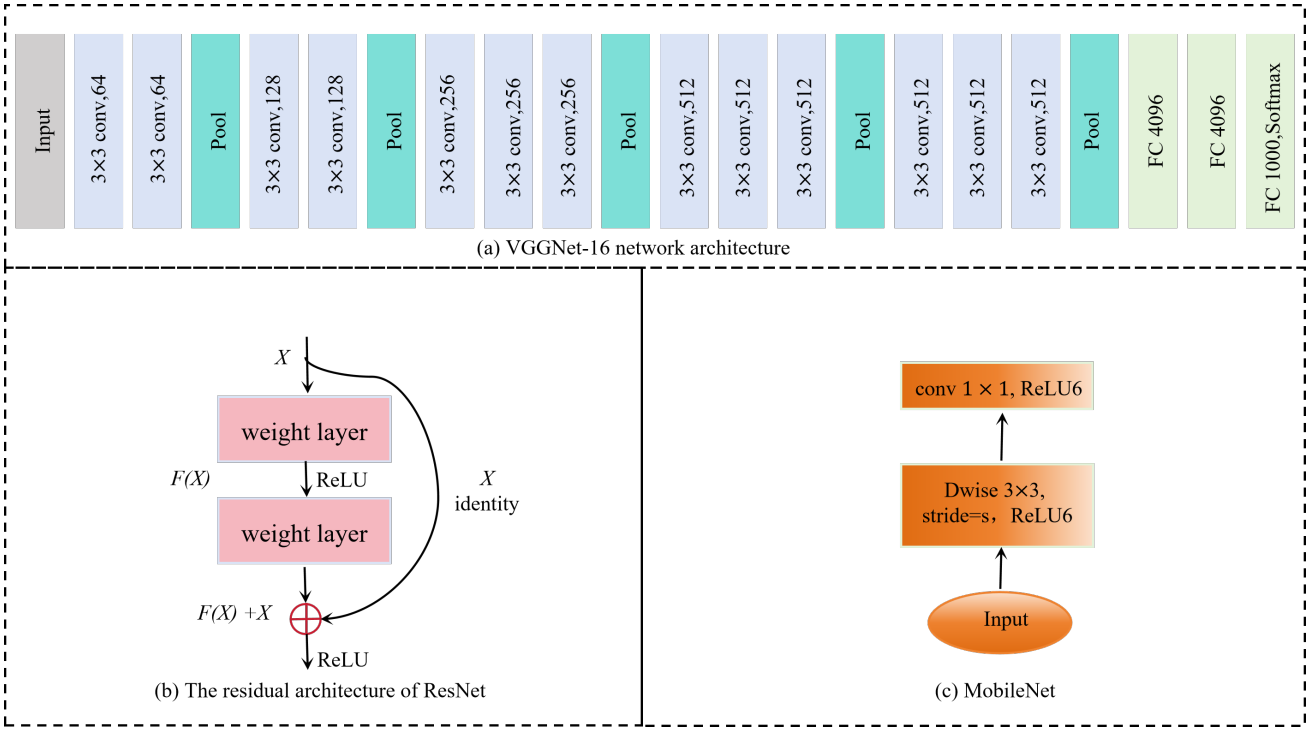


Figure 4. Schematic diagram of some important components of partial networks.

After multiple convolution operations, the dimensionality of the image feature map will be higher. This not only increases the computational load on the equipment, but also often leads to premature overfitting. Therefore, reducing the dimensionality of features while preserving key information in the feature map is an extremely important process. Commonly used operations include MaxPool, AvgPool, and AdaptivePool. Maximum pooling is selecting the maximum value within the pooling window. This pooling operation not only improves the mean error bias caused by too many convolutional layers, but also largely preserves more texture feature information. Mean pooling is the process of calculating the average value within a window. By pooling windows and moving them regularly within a specified area, the mean of each window region can be obtained, which can better preserve the global contour of the image. Adaptive pooling is the process of adaptively adjusting the size of the pooling window based on the size of the target object in the input image, thereby enhancing the model's adaptability to different input objects.

After convolution and pooling operations on feature images, high-dimensional features need to be compressed into fixed dimensional vectors. Afterwards, map it to a specific space (such as the category probability space). Overall, its essence is to achieve linear combination of feature information through a weight matrix. Afterwards, the integrated features are passed into a classifier (such as Softmax) to complete classification or regression tasks. In summary, the fully connected layer serves as a "feature aggregation device" that associates "global-local" features, providing effective feature representations for subsequent tasks.

Therefore, these four modules together constitute the basic module of CNNs.

Table 1. General information about some classic datasets for general object detection.

Dataset	Total number	Number of categories	Category examples
PASCAL VOC	9963	20	“person”,“train”,“chair”,etc.
MS-COCO	164K	80	“cat”,“boat”,“apple”,etc.
ILSVRC	517K	200	“dog”,“rose”,“car”,etc.

3.2. Object detection datasets

In this section, we introduce some information about classic object detection datasets (see Table 1 for details), which are roughly described as follows.

PASCAL VOC (2007) [38]: the 2007 version of Pascal VOC dataset is a well-known and classic dataset in the field of object detection. There are 20 categories in this dataset. The total number of samples is approximately 9963 images, with 24640 annotated target objects.

PASCAL VOC (2012): in the 2012 version, this dataset had 11540 image samples and a total of 20 categories. It is an upgraded version of the 2007 version.

MS-COCO (2017) [37]: the MS-COCO dataset contains approximately 164K images with 80 categories.

ILSVRC (2017) [48]: the images in the ILSVRC (imagenet large scale visual recognition challenge) dataset mainly come from the ImageNet dataset. There are approximately 517K images in this dataset, which includes 200 categories.

Table 2. General information about some classic datasets for general small object detection.

Dataset	Total number	Number of categories	Category examples	Application scenarios
DOTA	2806	15	“ship”,“large-vehicle”,“plane”,etc.	Remote sensing image detection
VisDrone	8629	10	“Pedestrian”,“bicycles”,“cars”,etc.	Unmanned inspection
TinyPerson	1610	2	“sea person”,“earth person”	Crowd monitoring

3.3. Small object detection datasets

In the previous section, we introduced some classic general object detection datasets. In this section, we will introduce the classic datasets commonly used in small object detection. We provide information on some classic small object detection datasets in Table 2, which will be roughly introduced below.

DOTA [40]: the DOTA dataset (v1.0 version) contains 2806 images with pixel sizes of 4000 $\times$ 4000. There are 15 categories in this dataset. The total number of targets is 188282. The images in this dataset are mainly obtained from satellite remote sensing, including ships, large and small vehicles, planes, and swimming pools. Typical application scenarios include remote sensing image analysis.

VisDrone [39]: the VisDrone (2019) dataset contains three subsets. There are 6471 images in the train set, 548 images in the val set, and 1610 images in the test set. There are mainly 10 categories in this dataset. This dataset is mainly obtained from drone aerial photography. The target objects mainly include pedestrians, people, bicycles, cars, etc. The main application scenarios of this dataset include drone inspection and small object detection from the perspective of drones.

TinyPerson [49]: the TinyPerson dataset contains 1610 labeled images. This dataset can be divided

into sea people and earth people, where people on board ships, in the water, and with more than half of their bodies in the water are considered sea people. There are a total of 72,651 annotated objects. This dataset is suitable for use in scenarios where small target populations are monitored.

3.4. Evaluation metrics and related knowledge

After introducing some basic modules of CNNs and some commonly used datasets, this section will mainly focus on the evaluation metrics commonly used in object detection. To evaluate the performance of the trained model, a unified evaluation metric will be introduced for assessment.

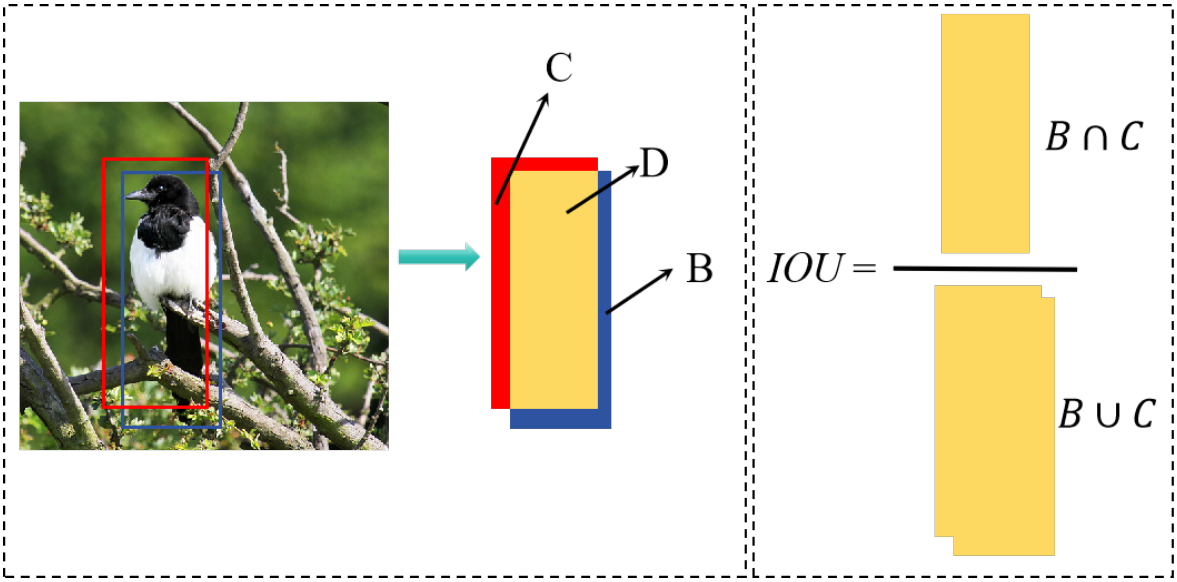


Figure 5. Schematic diagram of IOU calculation, where D is the intersection part between B and C. This indicator can be specifically regarded as the ratio of the "intersection" to the "union" between B and C.

The first indicator we introduced is intersection over union (IoU). To better illustrate it, we provide a schematic diagram in Figure 5. We assume that box B represents the true border of the target object in the image, and box C represents the predicted border of the model for the target object in the image. The specific formula is as follows:

$$IoU = \frac{B \cap C}{B \cup C} \tag{3.1}$$

Among them, the value of IoU ranges from 0 to 1. When performing object detection tasks, we make the following assumptions: the object to be detected (the target object within the real box) is the desired prediction sample, which we refer to as a positive sample. And the area outside the real box is called a negative sample. Based on these situations, there are four situations in the detection process: 1) true positive (TP): actual positive samples, and the model also predicts them as positive samples; 2) true negative (TN): actual negative samples that the model also predicts as negative samples; 3) false positive (FP): it is actually a negative sample, but the model predicts it as a positive sample; 4) false negative (FN): it is actually a positive sample, but the model predicts it as a negative sample. The above four possible scenarios can be applied to derive the following two indicators (see formulas 3.2 and 3.3

for details). Firstly, there is precision (P), which represents the ratio of all predicted positive samples that are actually positive samples.

$$P = \frac{TP}{TP + FP} \tag{3.2}$$

As for recall (R), it refers to the ratio of correctly predicted positive samples among all samples that are actually positive samples.

$$R = \frac{TP}{TP + FN} \tag{3.3}$$

Overall, precision and recall are interdependent. If higher precision is pursued, the recall rate will be relatively reduced. The PR curve is calculated with recall rate R as the horizontal axis and precision P as the vertical axis, and the average precision (AP) of each category is as follows.

$$AP = \int_0^1 P(R)dR \tag{3.4}$$

In addition to the above indicators, the mean average precision (mAP) indicator is equally important. The average precision of each category can be obtained through the following formula.

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{3.5}$$

where i represents the category.

4. General one-stage object detection frameworks

In Section 3, we provided a general introduction to some important prerequisite knowledge for object detection. In this section, we will officially begin the introduction of general object detection methods. In Section 4, we focus on introducing the one-stage object detection algorithm. This mainly includes the classic YOLO series, SSD series, and other strategy based one-stage detection methods. Unlike the rough introduction strategy used in some reviews, we strive to provide a detailed introduction to each method as much as possible.

4.1. YOLO series object detection methods

Before starting the introduction, the approximate chronological order of the classic YOLO series methods is shown in Figure 6.

(1) YOLO [50]: some two-stage object detection algorithms, such as faster R-CNN, have achieved superior detection speed and accuracy after multiple improvements. However, this type of algorithm is still time-consuming in the step of generating candidate regions, so its detection speed still falls short of expectations. YOLO (also known as the first version of the YOLO) has made improvements to address this deficiency. The YOLO series algorithm is an important class in object detection models, which can meet the real-time requirements of most detection scenarios and achieve end-to-end detection needs. The first version of YOLO algorithm proposed by Redmon achieved a detection performance of 45 frames per second. Compared to most previous deep learning object detection algorithms, YOLO's

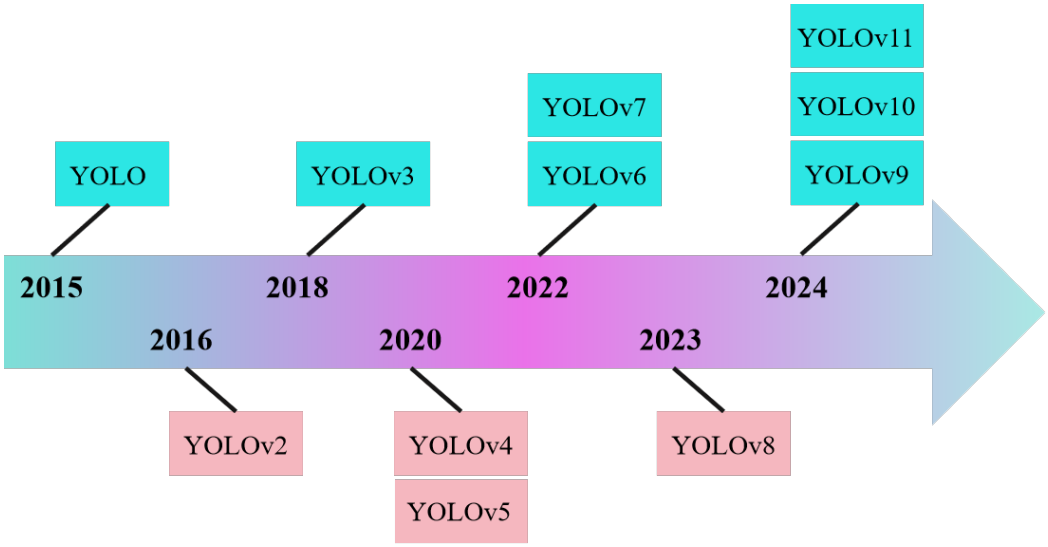


Figure 6. Timeline of YOLO series methods release. In contrast, each generation of YOLO versions has its own highlights. YOLOv2 introduced anchor boxes and Darknet-19. YOLOv3 introduced Darknet-53 and multi-scale prediction. YOLOv4 introduced spatial pyramid pooling (SPP), path aggregation network (PANet), and Mosaic data augmentation to enhance model performance. YOLOv5 introduced spatial pyramid pooling fast (SPPF). YOLOv6 introduced a more efficient EfficientRep network. YOLOv7 introduced (convolution batch normalization sigmoid-weighted linear unit (SiLU) CBS) and efficient layer aggregation network (ELAN). YOLOv8 introduced C2f and Anchor-Free design. YOLOv9 introduced programmable gradient information (PGI) and generalized efficient layer aggregation network (GELAN). YOLOv10 introduced NMS-free training. YOLOv11 introduced dual small convolution kernels to replace large convolution kernels.

main contribution is to transform object detection into a regression based model architecture, achieving faster detection speed and stronger robustness. Good results have been achieved in image detection in multiple scenarios. Overall, this method treats the detection of target objects in images as a regression problem. To better illustrate this method, we provide a rough flowchart of the YOLO detection model in Figure 7. The YOLO algorithm first segments the image in the input model using a grid format with a size of $S \times S$. Then calculate the confidence values of the bounding boxes where each grid is located. Furthermore, the model will predict the category of pixels in each block within each box to determine which category they may belong to. Subsequently, the NMS algorithm was introduced to eliminate overlapping boxes.

(2) YOLOv2 [51]: although the first version of YOLO has made good progress, there is still room for improvement in accuracy and other performance indicators. In order to improve some of its shortcomings, YOLOv2 was proposed by Redmon et al. YOLOv2 uses batch normalization (BN) after each convolutional layer to achieve higher training efficiency. In order to obtain more suitable prior boxes, the convolution and clustering analysis of anchor boxes are introduced to improve the detection performance of the detection model for target objects of different sizes. Meanwhile, by utilizing the new backbone network Darknet-19, higher computational efficiency can be achieved. This network can also reduce computational overhead while maintaining high computational

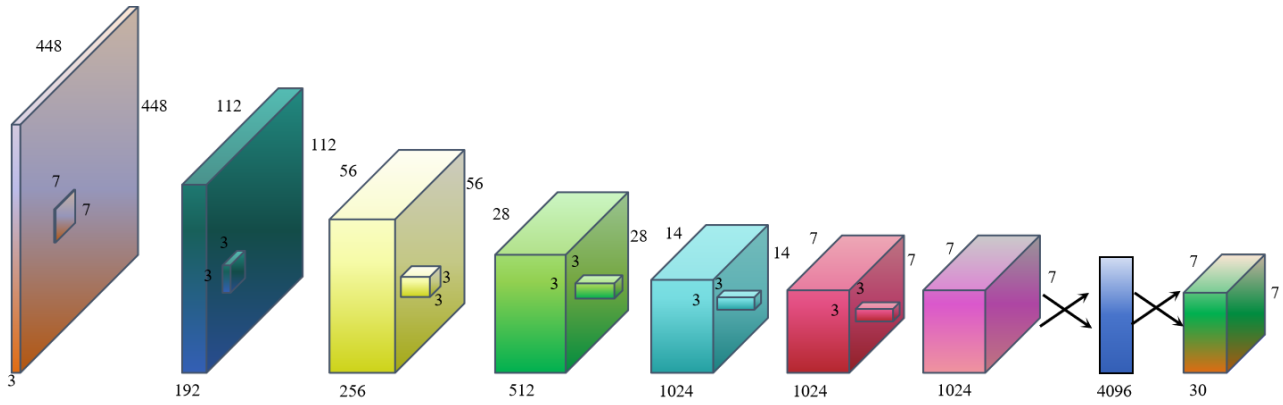


Figure 7. The general architecture diagram of YOLO.

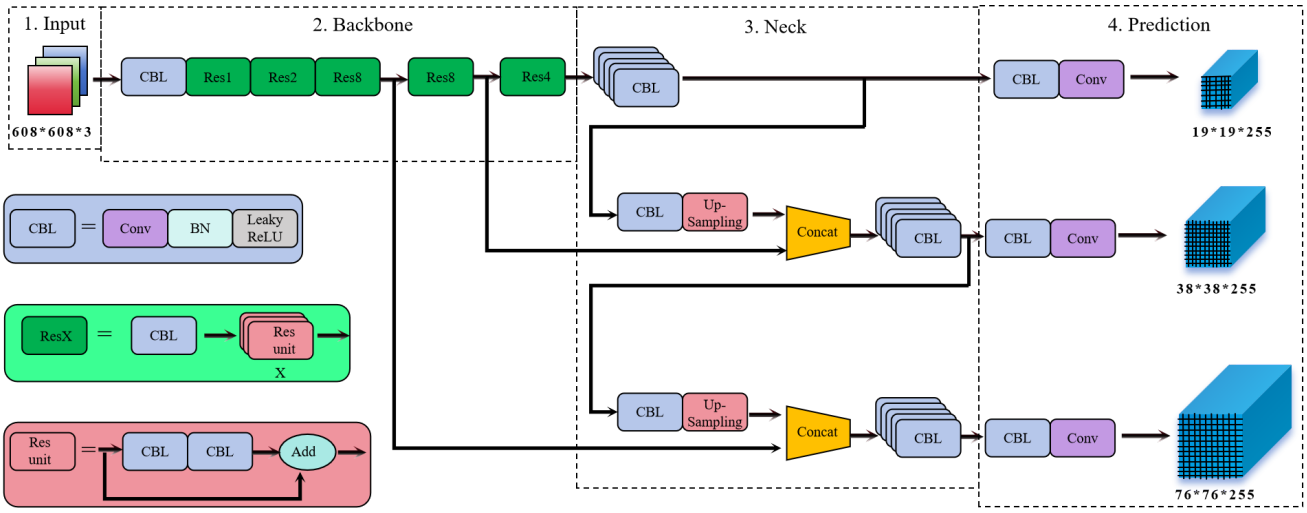


Figure 8. The general architecture diagram of YOLOv3. YOLOv3 introduced Darknet-53 and multi-scale prediction.

efficiency.

(3) YOLOv3 [52]: striving to better adapt to practical applications in certain scenarios, YOLOv3 has been proposed. The general architecture of YOLOv3 is shown in Figure 8. This algorithm uses Darknet-53 as a proxy for Darknet-19, increasing the number of layers in the network to enhance the model's ability to extract feature information. At the same time, in order to avoid the degradation of the network model as the number of layers increases, YOLOv3 introduces the idea of residual modules. In addition, in order to avoid the model only having good detection ability for single size target objects and neglecting other size target objects, YOLOv3 also introduces the idea of multi-scale. It is worth mentioning that the Softmax function is replaced by using logical regression. This also improves the detection performance of the model for multi label target objects in complex scenes.

(4) YOLOv4 [53]: after YOLOv3, Bochkovskiy et al. proposed YOLOv4, whose general architecture is shown in Figure 9. YOLOv4 enhances the feature extraction capability of the model by

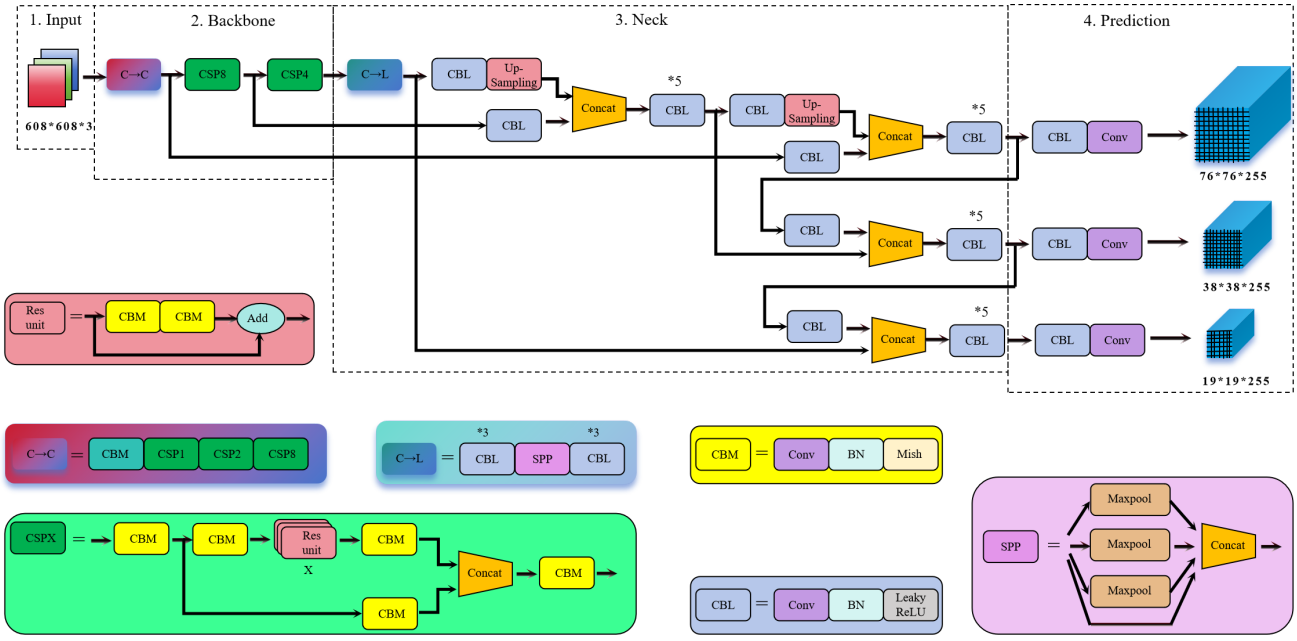


Figure 9. The general architecture diagram of YOLOv4. YOLOv4 introduced SPP, PANet, and Mosaic data augmentation to enhance model performance.

introducing SPP and PANet networks. SPP can enhance the receptive field of the model through multi-scale pooling operations, thereby promoting the model to better handle target objects of different sizes. The PANet network can better process feature information by transmitting features from top to bottom. At the same time, with the aim of further improving the performance of detection models, Mosaic data augmentation operations have been proposed. Effective data augmentation methods [54] can improve the performance and generalization ability of models. Not only that, YOLOv4 uses Mish activation function instead of LeakyReLU function to help the model acquire the ability to obtain information in complex distribution situations.

(5) YOLOv5 [55]: this algorithm inherits the advantages of the previous four versions and is an important watershed in the YOLO series. It is still used by many researchers in many practical detection tasks. We present the basic network architecture of YOLOv5s in Figure 10. The detection model mainly consists of a backbone network, a neck network, and a head network. In order to improve the computational efficiency of the model and its ability to extract feature information from images, the SPPF module is introduced, which can replace parallel operations with serial max pooling. YOLOv5 uses feature pyramid network (FPN) and PANet to aggregate features of different scales for better feature extraction of input images, thereby obtaining more effective feature information. In addition, there are four main architectures in the YOLOv5 series detection models, namely YOLOv5s, YOLOv5l, YOLOv5m, and YOLOv5x, with their model parameter sizes increasing in sequence. YOLOv5l and YOLOv5s exhibit superior detection performance when the number of model parameters is moderate or small. In addition, YOLOv5 introduces the CIoU (complete IoU) loss function, aimed at improving the regression speed of the model for detection boxes and the detection performance of the model.

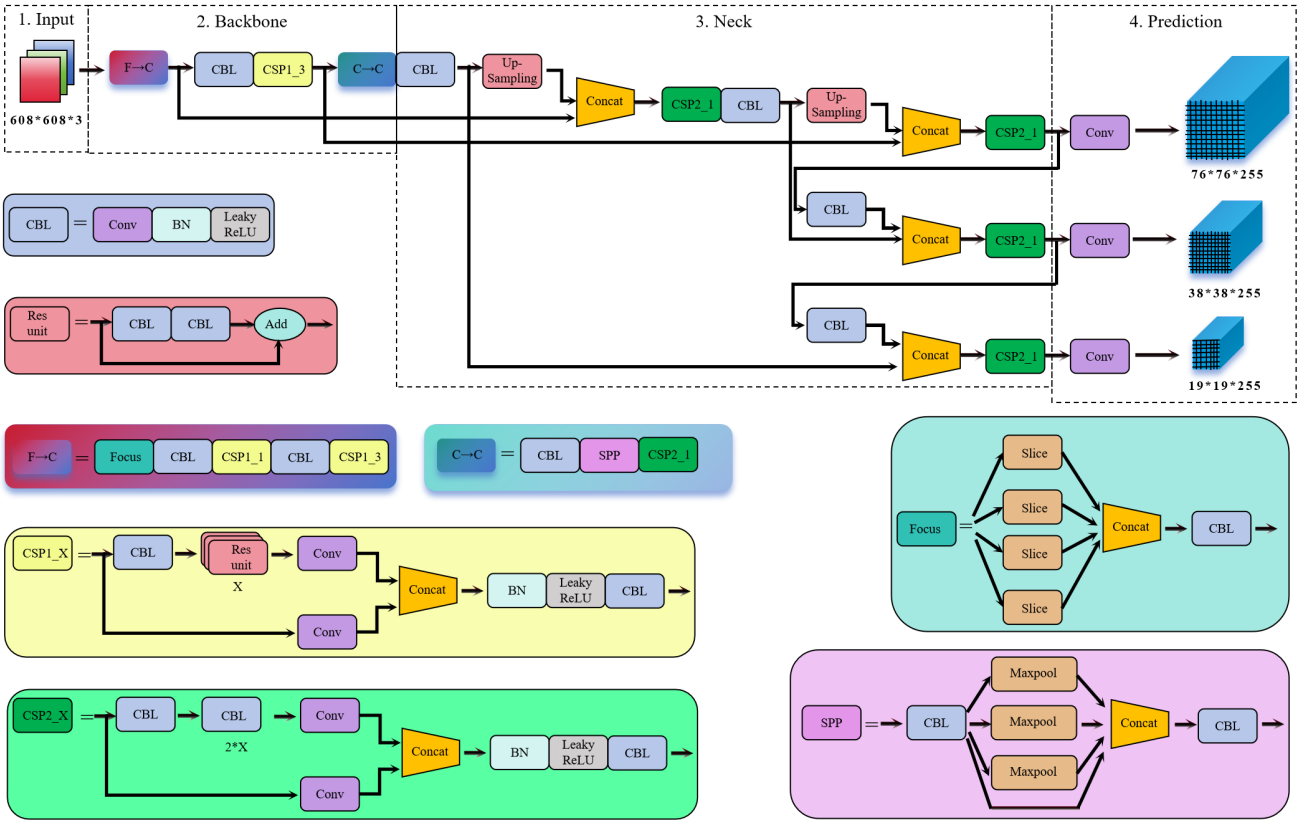


Figure 10. The general architecture diagram of YOLOv5s.

(6) YOLOv6 [56]: we provide a rough architecture diagram of YOLOv6 in Figure 11. This method adopts a deeper backbone network EfficientRep, aiming to accurately obtain the feature information of the target object in the image. Moreover, in order to reduce the time overhead of the model, this algorithm adopts a more concise decoupling head. Finally, in order to further improve the detection accuracy of the model, more effective label allocation strategies and bounding box regression are adopted. Through these improvements, YOLOv6-N achieved an AP of 35.9% in the COCO dataset.

(7) YOLOv7 [57]: the improvement points proposed by Wang et al. for YOLOv7 are slightly different. They first proposed extend and compound scaling in order to more efficiently utilize existing parameters. At the same time, the method also introduces the CBS module in the network to achieve better feature extraction of different hierarchical features by the model. In addition, they use ELAN to better control the gradient flow during model training, striving to make the trained model more robust. Through the above improvements, YOLOv7 maintains high detection performance and speed while achieving less parameter and computational complexity.

(8) YOLOv8 [58]: the approximate architecture of YOLOv8 proposed by Jocher et al. is shown in Figure 12. This method replaces the C3 module with a lighter C2f module. This can not only reduce the parameter count of the model and the computational complexity of the equipment, but also further achieve the lightweighting of the detection model. Not only that, YOLOv8 also optimized Neck's feature fusion mechanism, improving the efficiency of transmitting contextual information in the feature map. More importantly, they further optimized the anchor free design. The optimized

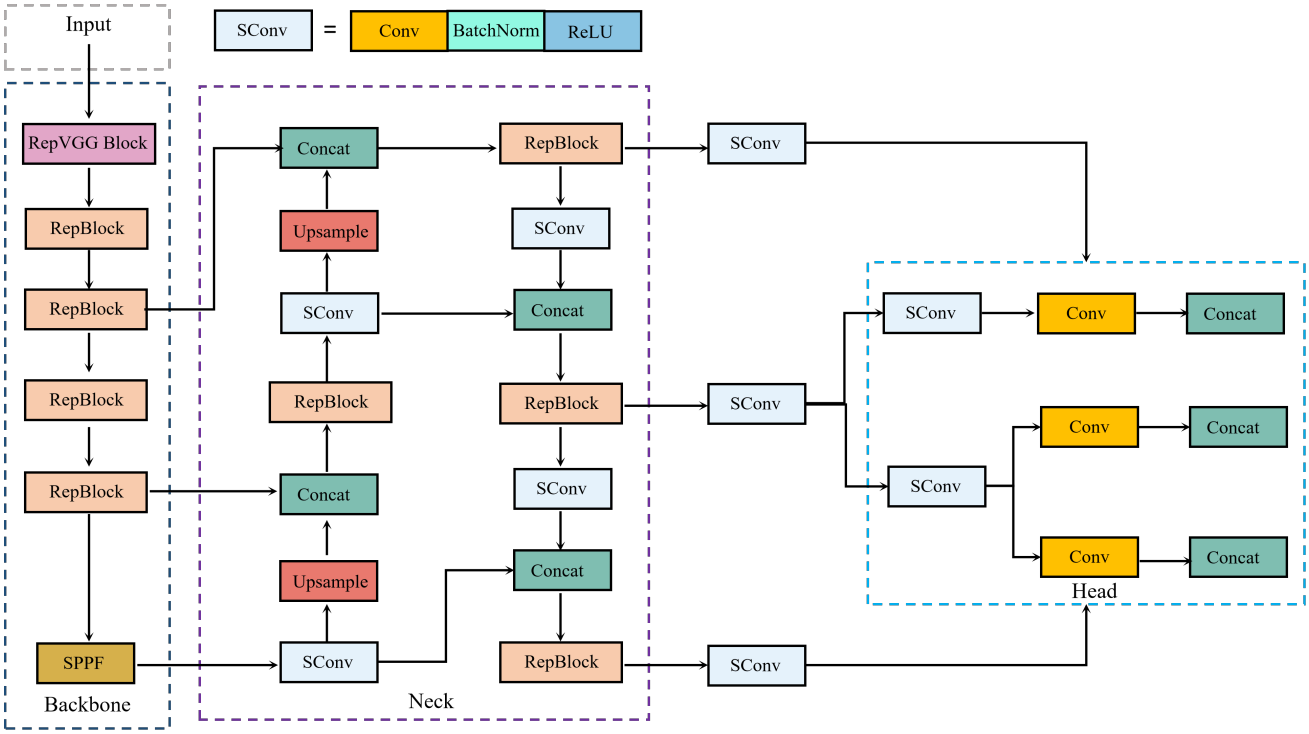


Figure 11. The general architecture diagram of YOLOv6.

anchor free module can better adapt to target objects of different scales.

(9) YOLOv9 [59]: the YOLOv9 proposed by Wang et al. uses PGI to improve the problem of partial information loss in feature maps. More importantly, the method model also uses GELAN. This module can better integrate various computing modules, thereby enabling the model to achieve higher parameter utilization.

(10) YOLOv10 [60]: we present the general architecture diagram of YOLOv10 in Figure 13. This model significantly reduces the computational consumption of the model by introducing a detection head that does not require NMS and optimizing other module components, achieving a relatively balanced model of "accuracy-latency".

(11) YOLOv11 [61]: YOLOv11 was proposed by Khanam et al., and its general architecture diagram is shown in Figure 14. This mechanism uses two small convolution kernels instead of one large convolution kernel, which not only improves the perception ability of local information in the feature map, but also enhances computational efficiency. Furthermore, they also introduced shortcut branches, whose principle is similar to that of residual connections. Then, the main branch and shortcut branch are fused to reduce the loss of important feature information.

Overall, the YOLO series methods have achieved good results, but there are also some shortcomings. This type of strategy uses grid partitioning to detect target objects. Therefore, for targets that only occupy a small number of pixels, there is limited effective information that can be obtained, resulting in poor detection performance. At the same time, when there are dense target objects in the image, they are prone to mutual interference and may also experience missed or false detections.

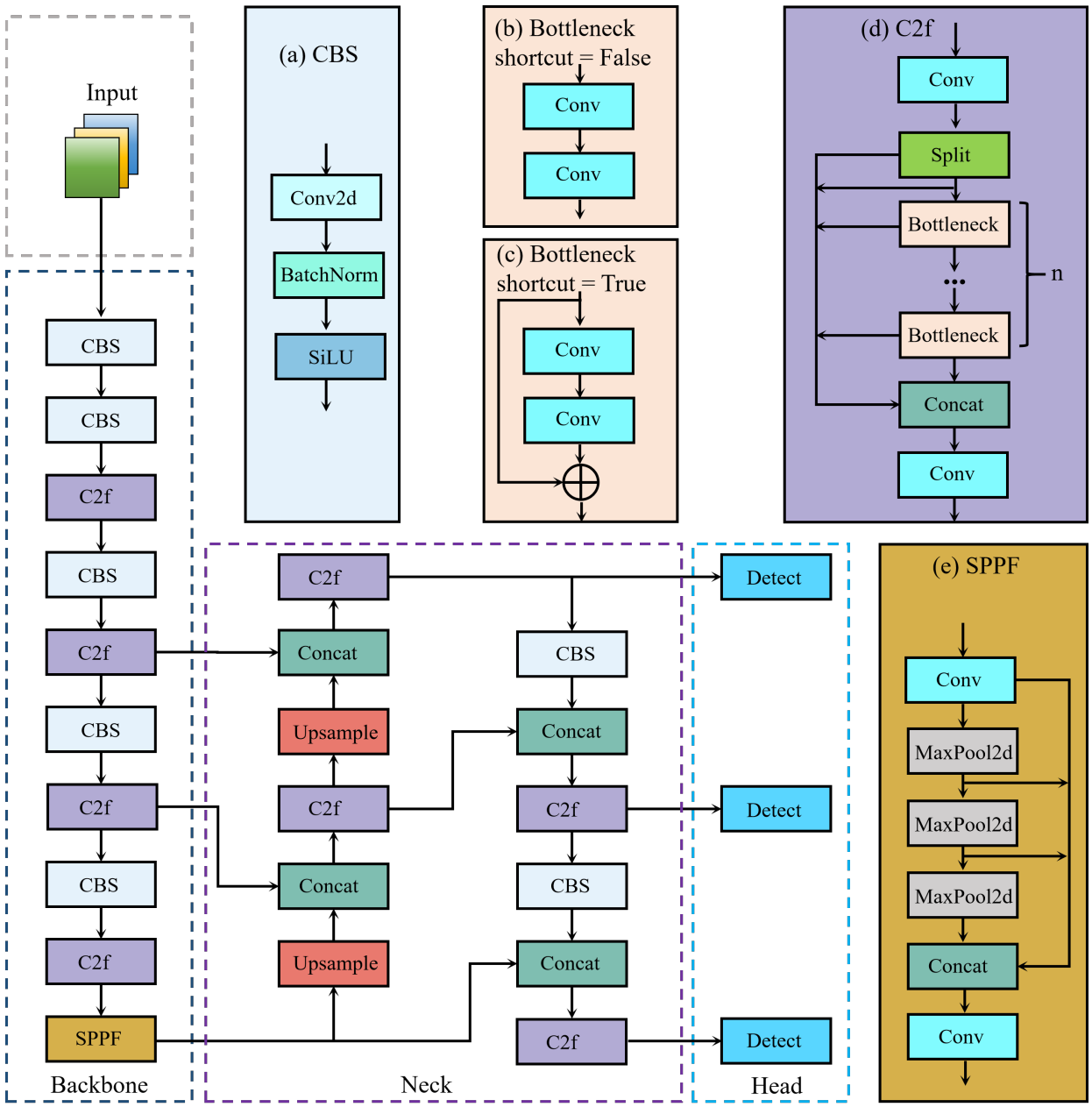


Figure 12. The general architecture diagram of YOLOv8. YOLOv8 introduced C2f and Anchor-Free design.

4.2. SSD series object detection methods

In section 4.1, we provided a detailed introduction to some classic YOLO series methods. In addition to the YOLO series methods, SSD series and other methods can also be used for object detection and have achieved good results. This is significantly different from the "grid partitioning + end-to-end regression" strategy mainly used in the YOLO series. The SSD series method takes multi-scale prediction as the starting point, allowing the model to efficiently detect target objects of different scales in network layers at different depths. We also provide a rough timeline of the SSD

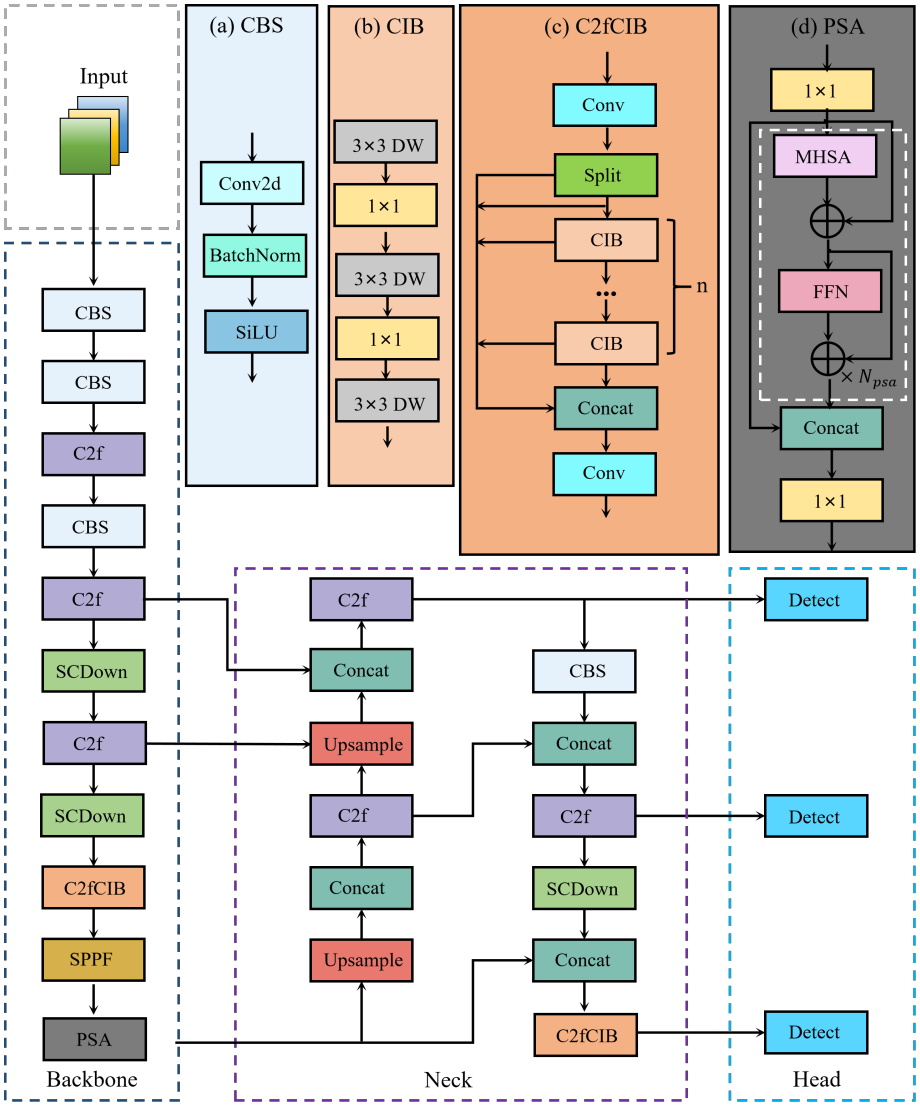


Figure 13. The general architecture diagram of YOLOv10.

classic method in Figure 15.

(1) SSD [62]: to achieve higher detection efficiency, Liu et al. proposed the SSD (single shot multibox detector) detection method. This method belongs to the one-stage algorithm, and its advantage is that it has a fast detection speed. This mechanism not only achieved high mean average precision (mAP) on the PASCAL VOC dataset, but also had a relatively high frame per second (FPS) transmission rate. The approximate model architecture of SSD is shown in Figure 16. VGG-16 was introduced as the backbone network, followed by the deployment of two convolutional layers to replace the original fully connected layer. Subsequently, by performing convolution operations at different scales, better feature information about different target objects can be obtained. Moreover, SSD uses NMS to process prior boxes fused at different scales. Different feature layers can better predict and detect target objects of different sizes.

(2) D-SSD [63]: the SSD method has achieved good results, but there are also some shortcomings. To effectively improve model performance, the D-SSD (deconvolutional SSD) method has been

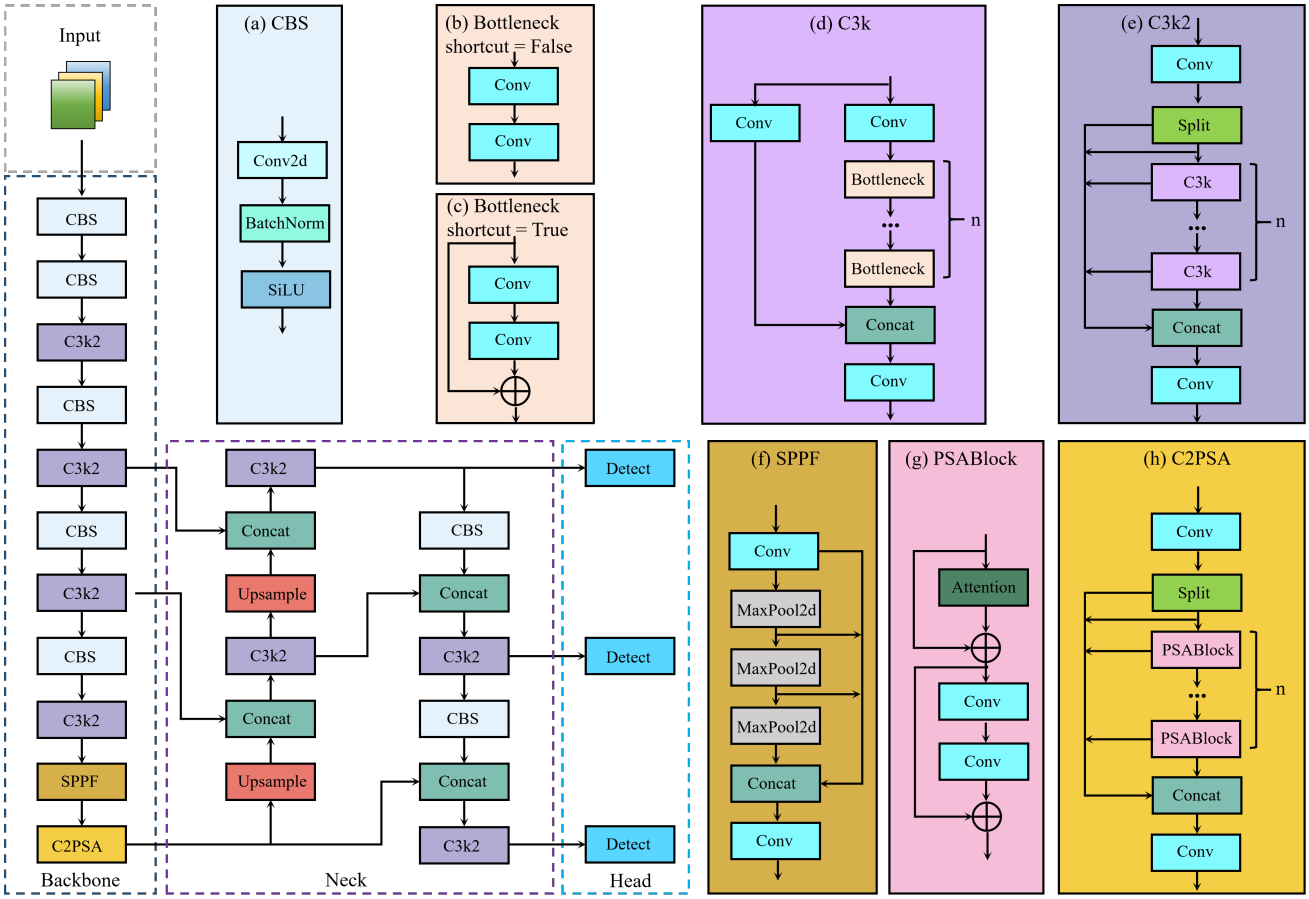


Figure 14. The general architecture diagram of YOLOv11. YOLOv11 introduced dual small convolution kernels to replace large convolution kernels.

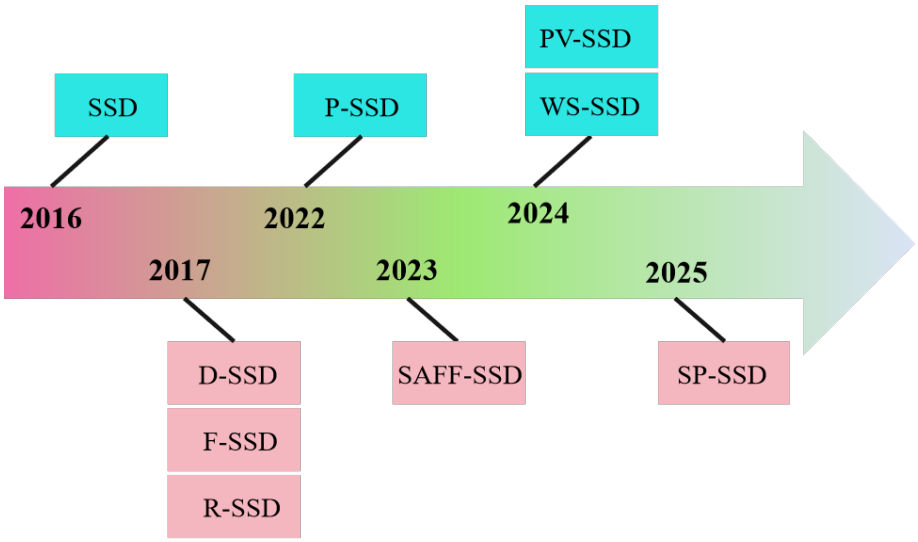


Figure 15. Timeline of SSD series methods release.

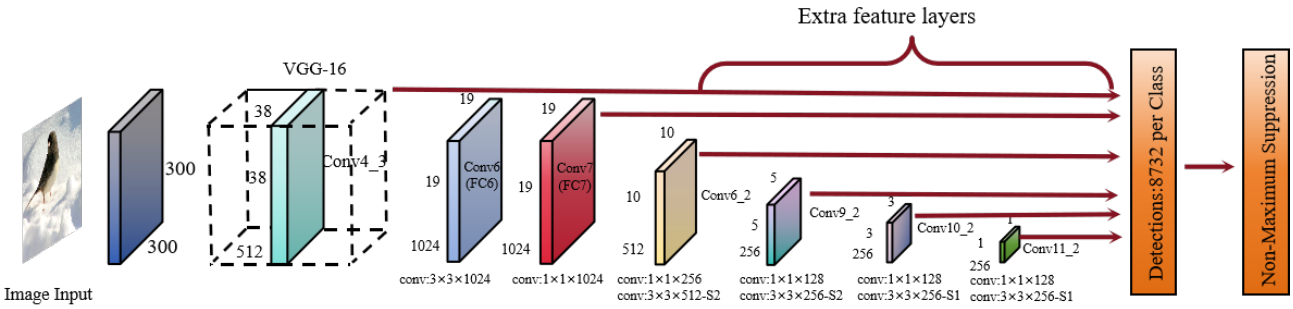


Figure 16. The general architecture diagram of SSD.

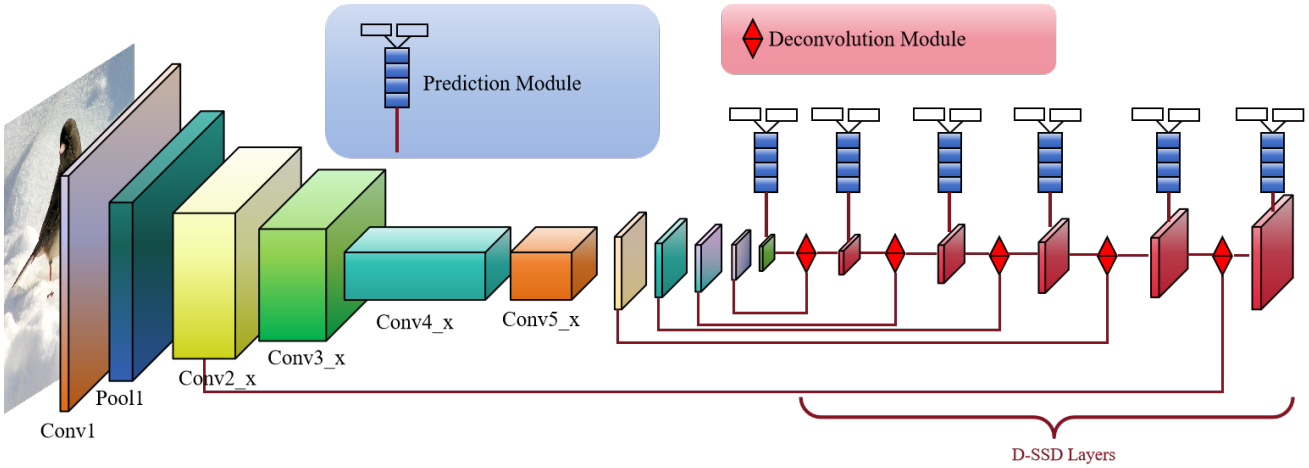


Figure 17. The general architecture diagram of D-SSD.

proposed, and its approximate architecture diagram is shown in Figure 17. On the one hand, in order to enhance the feature extraction network's ability to obtain feature information, D-SSD replaced the VGG-16 network of SSD with ResNet-101. Not only that, D-SSD further improves the detection performance of small target objects in images by adding deconvolution layers. The usage results on multiple datasets indicate that D-SSD has a certain improvement in performance compared to SSD.

(3) Feature fusion SSD (F-SSD) [64]: the F-SSD detection model is an improved one-stage object detection method based on the SSD method, and its model architecture is shown in Figure 18. F-SSD chooses ResNet-50 as the feature extraction network to obtain more semantically rich feature information. In addition, in order to obtain more scale feature information to improve the predictive performance of the model, a lightweight feature fusion module is added to the model.

(4) Rainbow SSD (R-SSD) [65]: one of the starting points of R-SDD is to better utilize the feature information between different levels. Therefore, it performs horizontal concatenation of feature maps at different levels within the same network, thereby better utilizing existing feature information. This can also enable the network to maintain relatively accurate detection in some complex scenarios. When processing the same image, networks at different levels can capture the information feature expressions that they focus on. Targeted concatenation and aggregation of these features can enable the model to

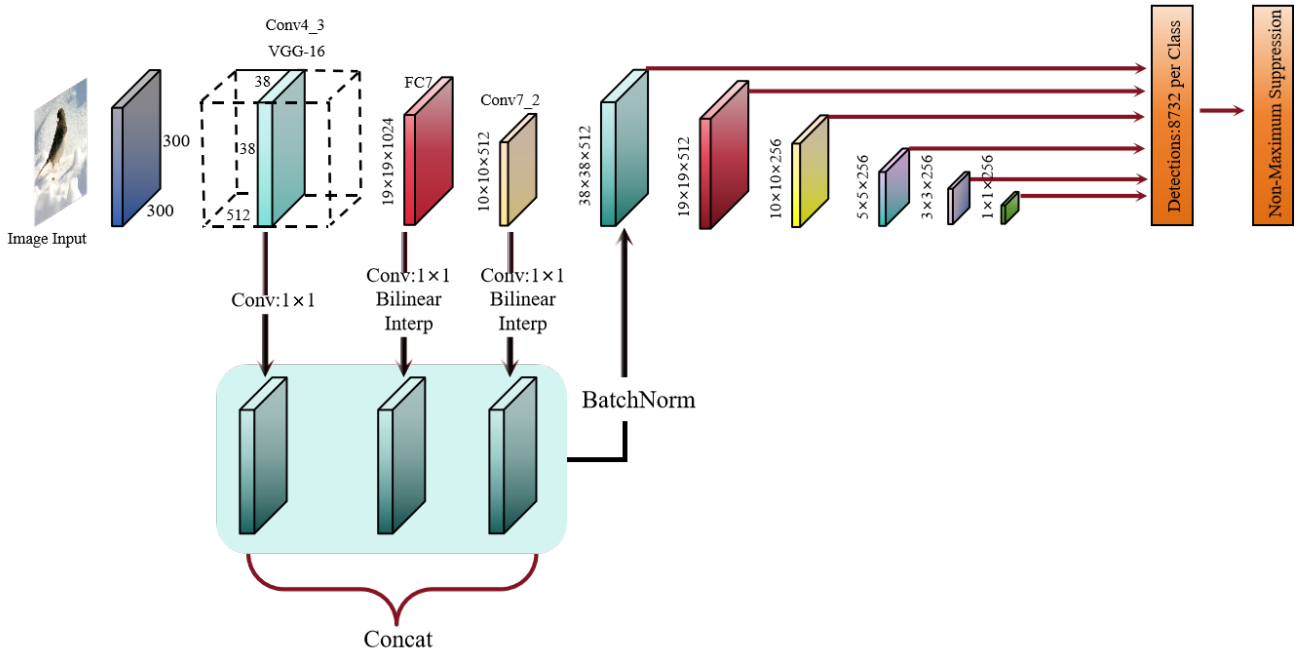


Figure 18. The general architecture diagram of F-SSD.

better understand the image.

(5) Precise single-stage detector (PSSD) [66]: PSSD considers better feature processing by adding an additional network layer. PSSD also constructs an effective feature enhancement module, enabling the model to better understand the rich information content between each level. More importantly, this strategy has designed a more suitable loss function to obtain better prediction boxes. The experimental results also indicate that this method can improve the detection performance of the model in complex situations.

(6) Self-attention combined feature fusion-based SSD (SAFF-SSD) [67]: aiming to better detect small objects in remote sensing images, Huo et al. proposed SAFF-SSD. This method uses the local lighted Transformer module to optimize the backbone network, in order to obtain better feature extraction capabilities. SAFF-SSD also uses cross layer connections to fuse features between different scales, thereby preserving feature information between different levels. Through these improvements, the detection performance for small-scale target objects has been further enhanced.

(7) Weighted sampling-SSD (WS-SSD) [68]: in the field of intelligent driving, Li et al. proposed WS-SSD for deployment on vehicle mounted autonomous driving devices with limited computing power. This method uses a weighted sampling single-stage detection architecture, which can reduce the model's dependence on high computing power devices while maintaining model performance as much as possible. Then, the weighted farthest point sampling strategy is used to achieve accurate sampling of the smallest objects even with fewer samples.

(8) SSD based on projection features and voxel features (PV-SSD) [69]: some object detection methods may suffer from information loss and other issues in 3D object detection. Shao et al. proposed PV-SSD to improve this problem. This model uses multi-branch feature extraction to obtain local

features and the correlation between local features for better feature processing. Meanwhile, in order to further reduce the loss of feature information during downsampling, this method uses multimodal feature fusion techniques for processing.

(9) Self-improvement SSD (SP-SSD) [70]: in the detection of garbage images, Tan et al. proposed SP-SSD to improve the responsiveness and detection accuracy of the model. This method uses depthwise separable convolution in the feature information extraction stage. SP-SSD also uses residual branches to enhance the generalization performance of the model. The experimental results show that this strategy has achieved good performance in the recognition of garbage images.

In summary, the SSD series methods have also achieved good results. This type of strategy mainly balances detection efficiency and scale adaptability through the strategy of "multi-scale feature map prediction". However, such methods also have some shortcomings. On the one hand, its detection scale is relatively single and cannot handle the feature information content between different scales well. Therefore, the detection stability is low in scenes where both extremely large and extremely small target objects exist simultaneously. On the other hand, due to the weak expression ability of this series of algorithms for shallow feature maps, their detection performance for small targets is weak.

4.3. Other object detection methods

In addition to the YOLO series and SSD series of one-stage object detection algorithms mentioned above, there are other effective and classic one-stage object detection methods. In this section, we will select some classic ones for introduction.

(1) CornerNet [71]: in order to improve the limitations of some methods that rely on a large number of anchor boxes, CornerNet proposes a method that utilizes keypoint detection. The main strategy is to fix the top-left and bottom-right corners of the target object. This method not only saves detection time, but also improves the overall efficiency of the model to a certain extent.

(2) CenterNet [72]: CenterNet considers that multi-point positioning may also lead to inaccurate positioning, so this method uses the center point positioning method to determine the boundary of the predicted box. In addition, this method has fast reasoning ability and detection speed, even without NMS. This method has achieved good results in multiple datasets. Similarly, this method is not only applicable to object detection scenes in 2D environments, but also to object detection scenes in 3D environments.

(3) RetinaNet [73]: in some daily object detection tasks, there is inevitably an imbalance between samples, which leads to unsatisfactory performance of the model. RetinaNet introduces Focal Loss to make positive and negative samples more balanced. The one-stage object detection method inevitably has certain shortcomings in the accuracy of detection, so RetinaNet adds ResNet network as the backbone network to improve the feature extraction performance of the model. In addition, in order to achieve more accurate recognition of targets at different scales, the model introduces a FPN. The results in practical applications show that this method can perform well in detecting complex backgrounds in some images.

(4) EfficientDet [74]: aimed at enabling detection models to have high detection accuracy and fast detection speed, Tan et al. proposed the EfficientDet detection model. This method innovatively incorporates weighted bidirectional feature pyramid network (BiFPN) module. This module has excellent performance and can better and more efficiently aggregate the required feature information of different scales, thereby achieving better multi-scale feature expression. The network model added

to this module can perform very well in multiple datasets.

The above methods have achieved good results in general object detection. In addition, in recent years, object detection strategies based on Transformer architecture have also been proposed, achieving good detection accuracy.

(5) DETR [75]: Carion et al. proposed the DETtection TRansformer (DETR) method. This method is based on the Transformer architecture and an end-to-end object detection approach, transforming detection tasks into direct set prediction problems. This method abandons manually designed component strategies such as anchor generation and simplifies the overall process. In addition, this method also draws on the encoder architecture in Transformer and combines global contextual inference. Meanwhile, DETR utilizes binary matching loss to ensure the reliability of prediction results.

(6) Deformable DETR [76]: the DETR method suffers from slow convergence speed and insufficient feature space resolution in image mapping. Therefore, Zhu et al. proposed Deformable DETR. The core strategy of this method is to use an efficient attention mechanism to focus on a small number of key pixel sampling points around the reference point, thereby improving the overall efficiency of the model.

(7) Conditional DETR [77]: to address issues such as slow convergence in DETR training, conditional DETR introduces a conditional cross attention mechanism, allowing the attention head to focus on specific key regions.

(8) DAB-DETR [78]: Liu et al. proposed the dynamic anchor boxes for DETR (DAB-DETR) method. This method utilizes dynamic anchor boxes for layer by layer dynamic updates, thereby solving problems such as slow convergence in the original model training.

(9) DN-DETR [79]: to effectively improve the stability of the model, Li et al. proposed the DeNoising DETR (DN-DETR) method. This method reduces the matching difficulty during model reconstruction by introducing noisy real borders.

The DETR series methods based on Transformer have achieved good results, but there are also some shortcomings. Due to its high model complexity, the parameter and computational complexity of its model itself are greater than most YOLO series methods. In addition, the real-time inference performance is insufficient, making it difficult to adapt to scenarios that require high implementation performance. Moreover, in some small object detection tasks, these limitations are further amplified due to the need for the model to be deployed on devices with lower computing power. The high demand for computing power limits the processing capability of such models for small target objects.

5. General two-stage object detection frameworks

In Section 4, we mainly introduced the use of a one-stage object detection model, and in this section, we will mainly introduce two-stage object detection methods. The one-stage object detection method has a faster detection speed, making it more suitable for real-time detection tasks. The two-stage object detection algorithm has higher accuracy in detection performance. In this section, we will provide a detailed and focused introduction to two-stage object detection methods, mainly based on the R-CNN series. We will introduce these classic and well-known methods in two parts. They are the methods named after the R-CNN series and other two-stage object detection methods.

5.1. R-CNN-named series methods

Unlike the one-stage strategy used by YOLO series and SSD series, the R-CNN series method mainly adopts a two-stage strategy. The two-stage strategy of R-CNN involves generating candidate regions that are independent of the category and performing two sub tasks of classifying and locating each region. Although this two-stage strategy increases computational complexity and reduces real-time detection performance, it also improves detection accuracy to a certain extent. In Figure 19, we present the timeline derived from the classic and advanced methods of the R-CNN series.

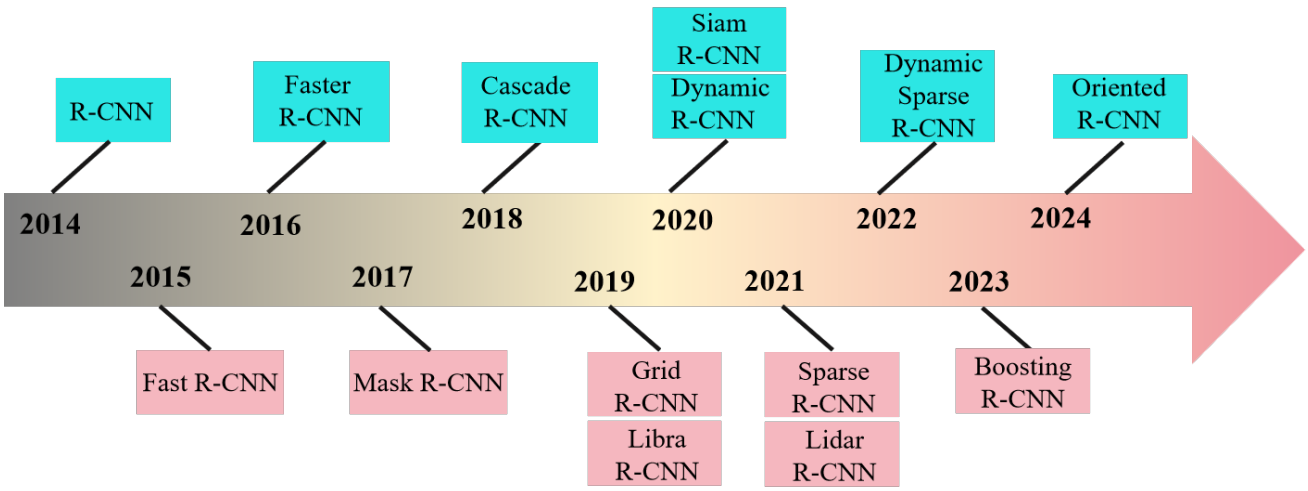


Figure 19. Timeline of R-CNN series methods release.

(1) R-CNN [80]: based on CNNs, Girshick et al. proposed R-CNN, and its basic process is shown in Figure 20. This algorithm became the beginning of deep learning in object detection. Compared to traditional object detection methods, this algorithm significantly improves detection performance. Good results have been achieved in multiple public datasets for object detection. R-CNN uses the selective search (SS) method to generate approximately 2000 candidate regions in the image, and then uses the AlexNet network to extract feature information from the generated candidate regions. Then, all extracted features are passed into a classifier mainly composed of support vector machines (SVM) for classification and judgment. In addition, in order to obtain better classification boundaries, R-CNN performs linear regression on the already classified recommendation boxes and adjusts the candidate boxes. The proposal of R-CNN has greatly promoted the development of the field of computer vision.

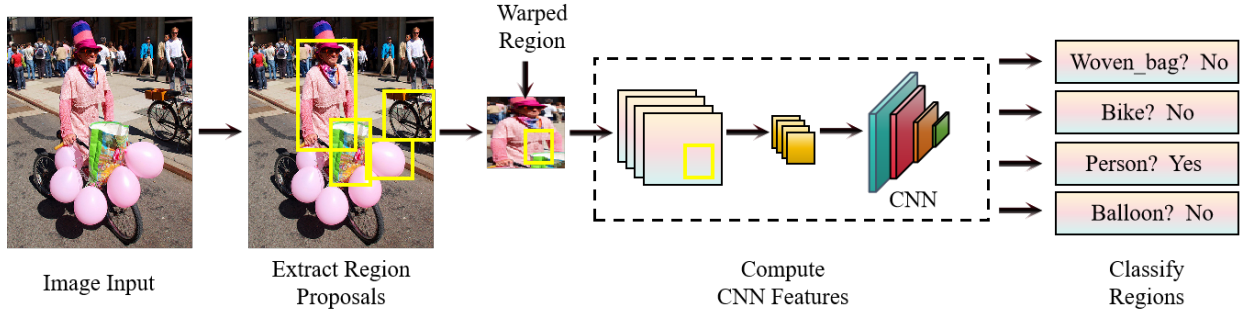


Figure 20. The general architecture diagram of R-CNN.

(2) Fast R-CNN [81]: although the R-CNN model has achieved good detection results, its detection speed is slow and difficult to meet the requirements of real-time detection. The main reasons for this problem are roughly as follows. On the one hand, the candidate boxes generated by the model need to be fed into the AlexNet network for feature extraction, which is not only redundant but also time-consuming. On the other hand, this method requires the use of separate CNN feature extractors, SVM classifiers, and correction operations for candidate box positions. These three operations need to be performed separately, and the entire process will consume a lot of time, making it difficult to meet the requirements of real-time detection. To improve the above shortcomings, fast R-CNN has been proposed, and its general architecture is shown in Figure 21. This method integrates the feature extraction network module, classification module, and regression model of R-CNN, and then improves the time-consuming problem caused by the need for separate feature extraction for each candidate box in R-CNN through shared convolutional layers. Due to these shortcomings, fast R-CNN has been improved from multiple aspects. In order to more efficiently utilize the subsequent fully connected layers, fast R-CNN uses a more efficient region of interest pooling layer (RoI Pooling). In addition, to make the model run more efficiently. When processing images, this model only requires one convolution operation without the need for repeated feature extraction. Moreover, in order to promote end-to-end deployment of the model, the classification module of the model uses Softmax instead of SVM.

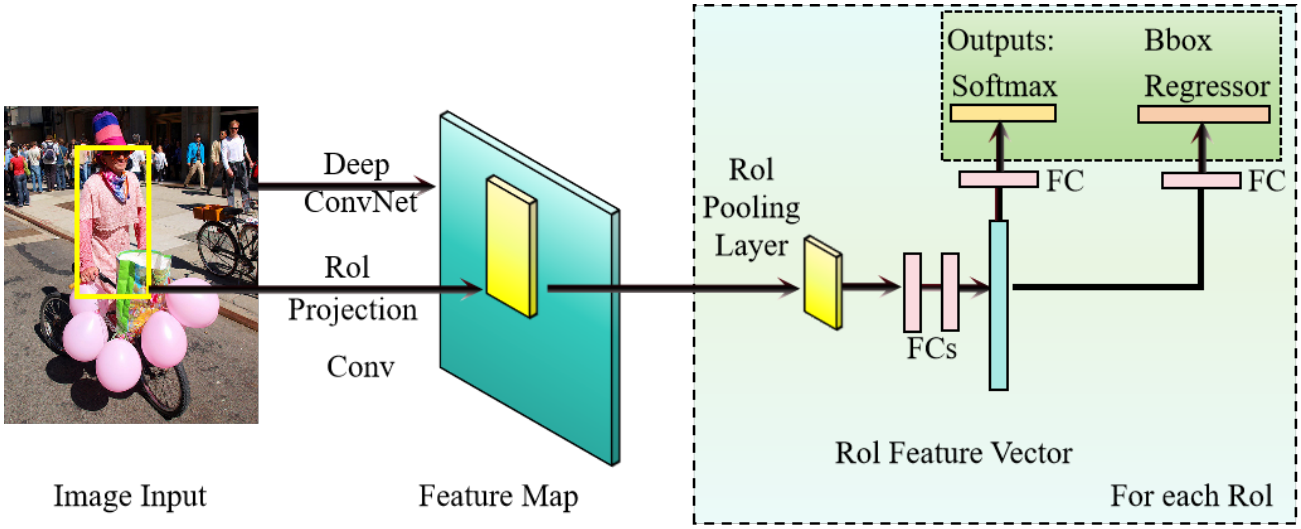


Figure 21. The general architecture diagram of Fast R-CNN.

(3) Faster R-CNN [82]: although fast R-CNN has made some improvements compared to R-CNN, it still has shortcomings. For example, the selective search strategy of this method also has problems such as long execution time. To further improve the algorithm, Ren et al. proposed faster R-CNN. This algorithm incorporates an edge extraction network into the model to replace the selective search strategy, and assigns the task of finding candidate boxes to the RPN network to improve the efficiency of searching for candidate boxes and significantly increase the detection speed of the model. In Figure 22, we present the general network architecture of faster R-CNN. In the feature extraction network section, faster R-CNN mainly utilizes the VGG-16 network for image feature extraction. This module

consists of convolutional layers, pooling layers, ReLU layers, and fully connected layers. It should be noted that R-CNN and fast R-CNN use a selective search strategy to generate candidate regions, which takes more time to generate candidate boxes. Unlike this, faster R-CNN utilizes an RPN network to generate candidate boxes. This not only improves the generation speed of candidate boxes, but also enables the generation of more accurate prior boxes about the target object based on the differences in feature information of the input image. After the input image undergoes feature extraction by the feature extraction network module, the size of its final feature map will be scaled proportionally. At the same time, the prior boxes generated by the RPN network will have different scales after undergoing regression operations again. However, when the feature information is transmitted into a fully connected network, it needs to conform to a certain size. Therefore, it is necessary to perform RoI Pooling on the feature information to ensure that it has a fixed size. In the classification regression module, this method uses a normalized exponential function and a fully connected layer to calculate the feature information contained in the prior box, in order to obtain the probability of the corresponding category. Then, a more accurate prior box about the image is obtained through prior regression operation to make the final result more accurate.

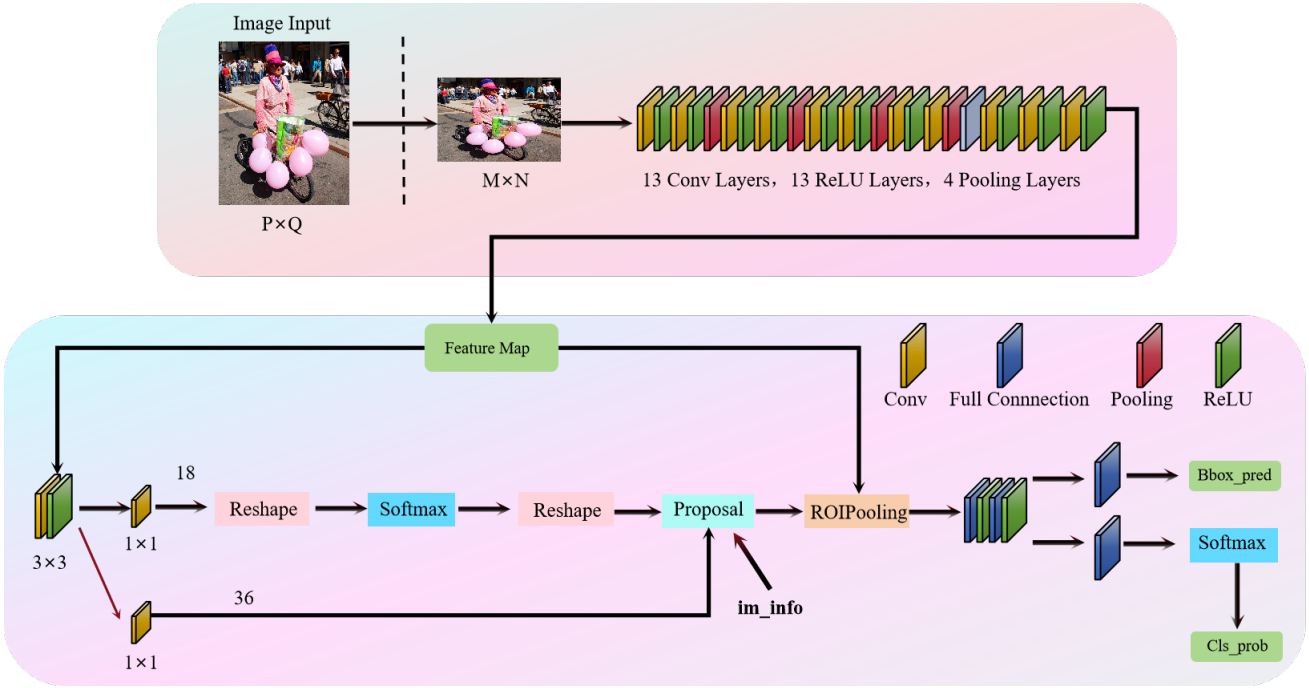


Figure 22. The general architecture diagram of faster R-CNN.

(4) Mask R-CNN [83]: He et al. improved upon faster R-CNN and proposed Mask R-CNN. The general process of this detection model is shown in Figure 23. In some previous works, some models had certain difficulties in obtaining the precise coordinate position of the target object. To address this issue, Mask R-CNN has designed a branch that enables the model to focus on the RoI mask in the image. Afterwards, in order to enable the model to obtain more accurate feature map information and better align RoI, the method introduced RoI Align technology.

(5) Libra R-CNN [84]: in the training process of some models, more attention is usually paid to the

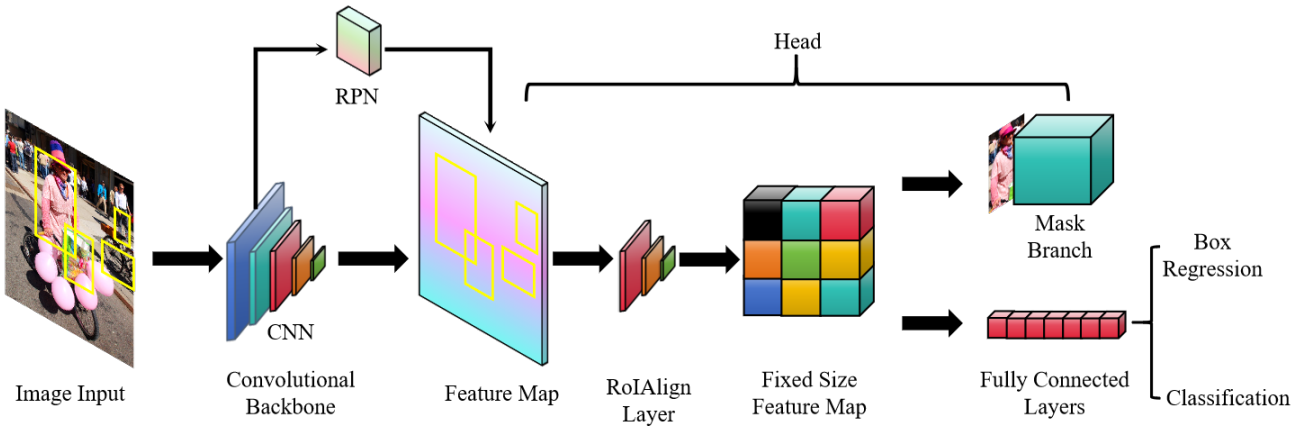


Figure 23. The general architecture diagram of Mask R-CNN.

categories with a larger number, while neglecting those with a smaller number and difficult to recognize categories. In order to better address the imbalance between these categories, Libra R-CNN has been proposed. Libra R-CNN improves the allocation of positive and negative samples by introducing a new balancing strategy. Subsequently, in order to ensure that the model does not overly focus or lean towards categories that are easily recognizable or have a larger quantity. This method assigns different weight coefficients to these categories (easily recognizable and difficult to detect), thereby increasing the model's focus on difficult class samples.

(6) Grid R-CNN [85]: Lu et al. proposed the Grid R-CNN object detection method in order to obtain richer feature information content and improve the final detection performance of the model. In order to obtain more feature information, this method uses a fully convolutional structural network with stronger ability to capture spatial position information. Then, the key positions of the target object in the image are located through grid points. It should be further explained that Grid R-CNN first uses pyramid pooling to process the feature information on the feature map after RoI Align operation, and then combines the strategies of dilated convolution and deconvolution. Subsequently, these processed feature information are generated into a heatmap, and the grid point with the highest confidence in the heatmap is selected as the final position point. In addition, this method aims to better handle the potential impact of grid points on the ground truth that may appear outside the region proposal. Grid R-CNN improves by adjusting the mapping relationship between the heatmap and the midpoint of the model input samples. The experiments on the COCO dataset also demonstrated the effectiveness of this method.

(7) Cascade R-CNN [86]: in order to improve the detection performance of the model, cascade R-CNN adopts the idea of cascading to optimize the model. This method has RPN and cascaded components at each stage to improve the detection performance of the model. First, select high-quality candidate regions through RPN operations in the cascade, and then pass them on to the model for the next step of operation. This model improves the detection performance and robustness of the detection model by continuously cascading operations.

(8) Dynamic R-CNN [87]: unlike most methods where the training parameters are fixed in one training cycle, Zhang et al. believe that the optimal parameter coefficients required should vary at

different stages of model training. For this, they proposed the dynamic R-CNN method. They first designed a dynamic label assignment module (DLA). Specifically, this module can dynamically adjust the threshold coefficient of IoU during model training, thereby enabling the model to generate too many high-quality candidate boxes. Afterwards, in order to better adapt the model to the distribution characteristics of the samples in the dataset and generate candidate boxes that are more suitable for the data samples, they designed a DSL (Dynamic SmoothL1 Loss) module that can dynamically adjust the parameters in SmoothL1 Loss at different times.

(9) Mesh R-CNN [88]: in order to achieve better detection performance in 3D object detection, this method adds a mesh prediction branch on the basis of Mask R-CNN. This not only enables more accurate feature representation of the target object, but also enhances the detection performance of the model.

(10) Siam R-CNN [89]: this method adopts a twin network architecture and innovatively proposes a strategy of re-detection. By introducing a classifier to classify the target, detection and tracking of the target objects can be restored even after they are occluded.

(11) Sparse R-CNN [90]: this method improves the running efficiency of the model by reducing the number of candidate boxes. In the processing of the final prediction results, a direct output method is used without the need for NMS processing. sparse R-CNN achieves a certain level of operational efficiency while ensuring a certain level of detection accuracy.

(12) Dynamic sparse R-CNN [91]: the dynamic sparse R-CNN method is an improvement on the Sparse R-CNN method. By introducing a dynamic sparsity mechanism, the region that best reflects the richer information in the image is selected, thereby reducing the redundant computation generated in the model. In addition, this method also uses dynamic label allocation to increase the number of positive samples during the training phase and generate more refined proposal boxes for subsequent training stages.

(13) Lidar R-CNN [92]: in 3D detection using LiDAR, Li et al. proposed Lidar R-CNN. This method uses a point cloud region proposal generation strategy. Then combine the data from different sensors to get the geometry and texture information of the object better.

(14) Boosting R-CNN [93]: this method introduces a multi-stage feature enhancement module to optimize the generation process of region proposals. In addition, this method also uses cascaded detection heads to gradually optimize the process of object detection. In terms of improving the performance of model methods, boosting R-CNN adopts a strategy of dynamic feature fusion.

(15) Pyramid R-CNN [94]: it is also a method for 3D object detection that uses feature pyramids to obtain points of interest in RoI. Afterwards, RoI attention is used to obtain sparse point information with richer information.

(16) Oriented R-CNN [95]: Oriented R-CNN uses a directional representation of the target, thereby reducing the complexity of the model during runtime and achieving highly competitive detection speed.

Overall, the R-CNN series of methods adopts a two-stage strategy of "region proposal+region recognition", resulting in significant improvements in detection accuracy performance. The shortcomings are also very obvious. The complexity of the model is high, the computational load is large, and the deployment difficulty on mobile devices is high, making it difficult to meet some scenarios that require real-time performance. At the same time, it is more dependent on the quality of candidate regions. When the background area in the image is cluttered or the target object is irregular, the accuracy will be affected to some extent.

5.2. Other named two-stage object detection methods

In addition to the R-CNN variant methods directly named R-CNN mentioned above, there are also some two-stage object detection methods that are not directly named R-CNN and have good results. In this section, we will introduce these methods.

(1) SPPNet [96]: there are two issues in R-CNN that limit its further performance improvement. On the one hand, this method will repeatedly calculate feature maps, which can lead to low efficiency and generate many invalid calculations. Therefore, the strategy adopted by SPPNet is to extract features from the feature map only once and then share them, thus avoiding repetitive operations and improving computational efficiency. On the other hand, R-CNN uses fixed sizes for computation, making it difficult to process images of different sizes. SPPNet introduces SPP to enable the model to handle input images of different sizes.

(2) R-FCN [97]: the purpose of the region-based fully convolutional networks (R-FCN) detection method is to solve the contradiction between displacement invariance in the task of classifying target objects and displacement variability in the detection task. Specifically, R-FCN introduces a RoI pooling layer that is sensitive to spatial location. By encoding each RoI position more accurately, the detection performance of the model can be improved.

(3) TridentNet [98]: generally speaking, in object detection, some object detection methods cannot detect objects of different sizes well. If it can detect target objects of different sizes well, it will undoubtedly greatly improve the detection performance of the model. Therefore, TridentNet mainly improves in two aspects. This method designs a trident block, which contains three branches, each with a different dilation convolution rate, used to process target objects of different sizes. Specifically, this method divides the size of the target object into three types, and then processes different types of target objects through different branches. Afterwards, TridentNet conducted in-depth research and extensive experiments on the receptive field. By continuously optimizing and adjusting some parameters of the dilated convolution, the final detection performance of the model can be improved.

(4) RefineDet [99]: RefineDet mainly improves the performance of the model by selecting better anchor points and obtaining accurate feature information. By improving the accuracy of anchor positioning, the performance of the model can be enhanced. RefineDet will fine tune the feature maps processed by the regional proposal network to make their localization more accurate. Striving to obtain more accurate feature information content through the detection model, the model will further process the RPN processed feature information to make it more precise. This enables the model to better identify features that are more distinguishable from each other.

6. Small object detection frameworks

In Sections 4 and 5, we respectively introduced some classic one-stage and two-stage object detection algorithms. However, currently these classic object detection models are not specifically developed and optimized for small object detection, so these detection models cannot achieve satisfactory performance in small object detection. It is worth noting that most of the current methods specifically designed for small object detection are based on improvements made to the general object detection model. In this section, we will focus on introducing some research achievements in small object detection and its research progress in various fields.

6.1. YOLO series small object detection models

In order to better obtain local and global contextual information of small targets, Yang et al. [100] proposed the ORIN-YOLOX method. Overall, the main strategy of this method is to obtain as much feature information as possible from the image. By combining the improvement of basic convolutional networks with vision multilayer perceptron, the model enhances the feature information representation of small target objects and obtains more effective feature information content. Then, the self attention mechanism Transformer is used to enhance the expression of these feature information. On this basis, the model will integrate these feature information contents from local to global form. The experiments on the VisDrone dataset also confirmed the effectiveness of this method.

Usually, aerial images often have occlusions and background interference. To alleviate this interference, Wang et al. [101] proposed YOLOX_w based on YOLOX. This method will enhance and expand the number of small target objects to increase their proportion. Then, a lightweight spatial attention module was added to better express the feature vectors of small target areas, making the small target objects more prominent and weakening the adverse effects of background interference. In addition, in order to better detect small target objects, a detection head is added to the architecture to make the detection model more sensitive to the perception of small target objects.

Hui et al. [102] proposed a method of using data augmentation strategy to improve the final performance of the model from the perspective of data augmentation. Simply put, this method first amplifies the initial image using super-resolution data augmentation techniques, and enhances it in a way that does not compromise its image quality. Then use a dense residual module to improve the quality of small target objects. Moreover, they also use modules that can optimize the basic features to reinforce the initial features. They integrated these strategies with YOLOv7 and proposed DSAA-YOLO (dense residual super-resolution and anchor frame adaptive regression YOLO). This method shows significant performance improvements compared to the baseline method in multiple metrics on the VisDrone dataset.

Jiang et al. [103] demonstrated and optimized the practical deployment of thermal infrared images and videos based on UAV platforms, and attempted multiple detection models from the YOLO series. Through practical experiments, it has been found that the lightweight network YOLOv5-s not only achieves high-precision detection (mAP of 88.69%), but also has fast detection accuracy, with an FPS of 50.

Similarly, Li et al. [104] improved YOLOv7 for better detection of drone images. To better detect the area where small targets are located, a detection layer specifically designed for small targets has been developed. In addition, an attention mechanism that is more sensitive to small target objects has been added to the detection head to obtain more prominent feature information about small target objects, in order to better detect them.

In UAV image recognition at higher altitudes, there are still some problems, such as insufficient clarity of image targets and a small proportion of required detection object pixels. Chang et al. [105] improved the lightweight detection model YOLOv5s. This model replaces the convolution module in the initial model to enhance the sensitivity of the backbone network to small target objects. Next, in order to enhance the sensitivity of feature information for small target objects, a coordinate attention mechanism that can strengthen the ability to extract feature information for small targets is added after the convolutional layer of the backbone network. The experimental results show that this method has good detection ability.

Sambolek et al. [106] considered the needs of search and rescue scenarios in mountainous areas and established a simulated search and rescue dataset, which was combined with some mainstream object detection methods. After multiple experiments, it was found that the YOLOv4 detection model can perform better and is suitable for use in search and rescue operations.

Zhou et al. [107] mainly improved the YOLOv4 model by optimizing the detection level. This strategy mainly modified the localization loss function in the backbone network to make the model more focused on small target objects and enhance the localization performance of small target objects. In addition, this strategy also uses data augmentation strategies to enhance the diversity of data samples.

In the field of forest management and detection, Wang et al. [108] improved the YOLO series and proposed a lightweight detection model lightweight and small object detection based YOLO (LDS-YOLO). This method first introduces an efficient pyramid pooling module to optimize the information extraction ability of the model for small target objects such as dead trees. In addition, this method also utilizes feature information content from the previous layer in the network to achieve efficient utilization of existing samples. On this basis, in order to make the detection model more lightweight, LDS-YOLO uses depthwise separable convolutions as convolutional blocks in the backbone network. The experimental results show that this method has strong detection performance and speed.

In order to better detect small target objects, Luo et al. [109] improved YOLOD based on YOLOv4 and proposed YOLOD (YOLO-Drone). Specifically, this method adds a channel attention module in the backbone network that can more effectively detect small target objects. Subsequently, based on the characteristics of different convolution levels, different activation functions are selected to make the model more efficient. Through these improvements, YOLOD has achieved better detection of small target objects based on UAVs.

In the field of agricultural applications, Gallo et al. [110] aim to improve productivity by efficiently removing weeds. This method mainly selects YOLOv7 as the benchmark model for detection. They mainly use drones to obtain images of chicory plant farmland areas and mark the weeds inside. After multiple experimental comparisons, it was found that YOLOv7 has high detection performance in weed detection.

The GhostConv-based lightweight YOLO network (GCL-YOLO) proposed by Cao et al. [111] uses the lightweight GhostConv network as the backbone to reduce the number of model parameters. In addition, in order to make the model more focused on small target objects in the image, two improvements were made to the model. First, GCL-YOLO reconstructs a detection head to focus on detecting small target objects. Moreover, this method also introduces a more optimal loss function as the localization loss. The experimental results on the VisDrone dataset show that this method has high detection performance. The general architecture of this method is shown in Figure 24.

Similarly, in the field of plant cultivation, Junos et al. [112] first obtained initial images through UAV photography and other methods. Then, various data augmentation strategies such as brightness adjustment and rotation are used to enhance and enlarge the original image. Subsequently, the challenge of activation function and the setting of multi-level network feature extraction are used to optimize the feature acquisition of small target oil palm. The YOLOv3-tiny model improved through the above strategies not only has good detection performance, but also is lightweight, which provides the possibility for deploying the detection model to small devices.

Zeng et al. [113] proposed spatial and coordinate attention enhancement YOLO (SCA-YOLO) to address the shortcomings of missed and false detections in UAV perspective detection. The general

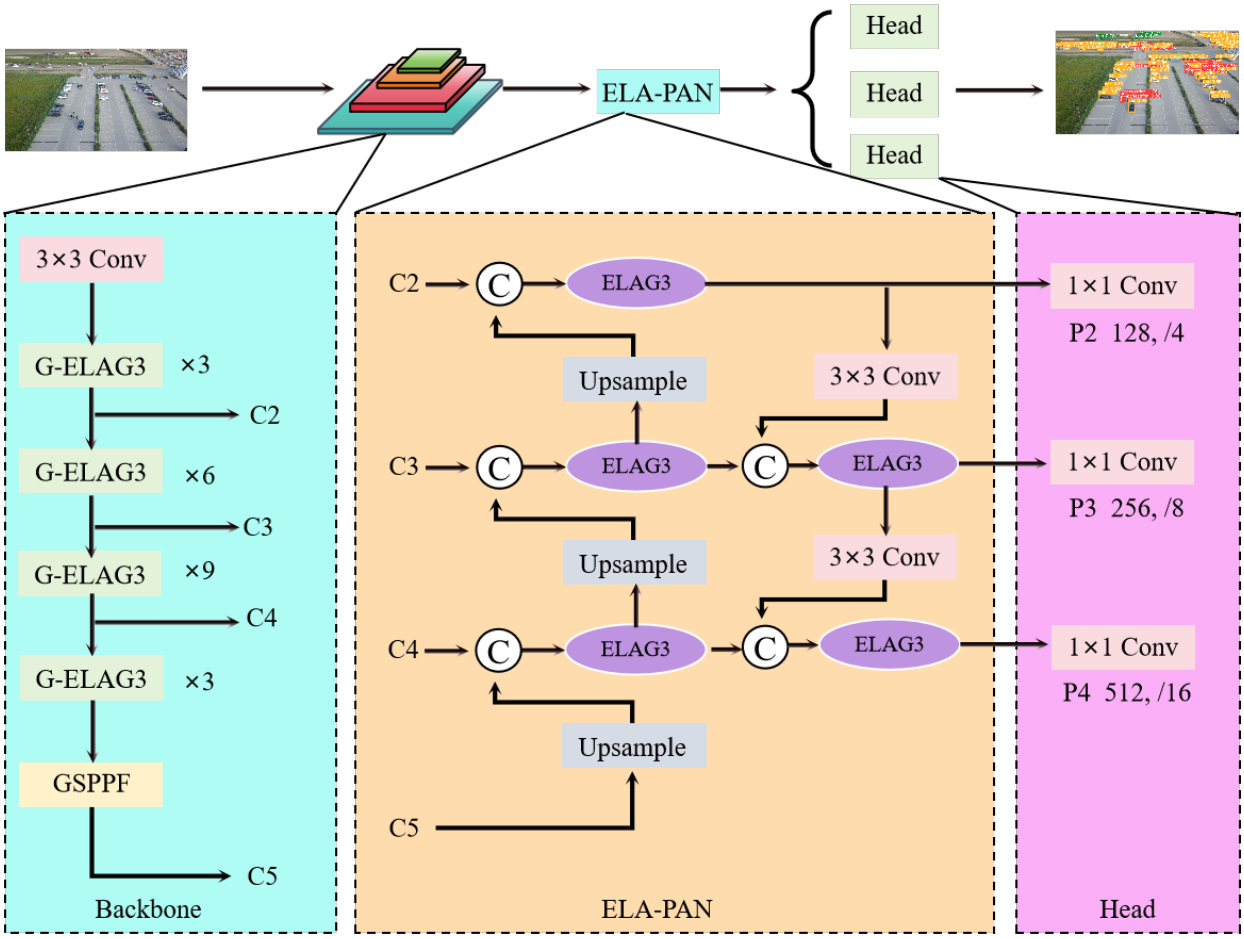


Figure 24. The general architecture diagram of GCL-YOLO.

architecture of this method is shown in Figure 25. Overall, this method mainly adopts the strategy of utilizing multi-scale feature fusion to improve the detection of small targets. This method uses spatial attention and coordinate attention modules to improve the problem of insufficient model localization ability. Afterwards, a more efficient bottleneck module is introduced to adjust the sensitivity of the model to important area objects. In addition, the adaptive feature network architecture in this method can enable the detection model to better represent the position of the target object.

Sun et al. [114] improved the benchmark detection model YOLOv3. By fusing shallow network layers with deeper network layers for representation, sufficient detailed features and global contextual features can be obtained. Then use the incentive layer in the attention mechanism to improve the detection performance of the model.

Hu et al. [115] proposed a lightweight detection model efficient-lightweight YOLOv5 (EL-YOLOv5) based on the YOLOv5 platform to enable the detector to be deployed on some low computing platforms. They introduced more efficient spatial pyramid pooling to optimize the model's feature information acquisition, enabling the model to obtain representation capabilities for small target object features. Subsequently, they employed a new loss function to address the issue of sample imbalance.

In the field of autonomous driving, in order to detect small targets more accurately and improve

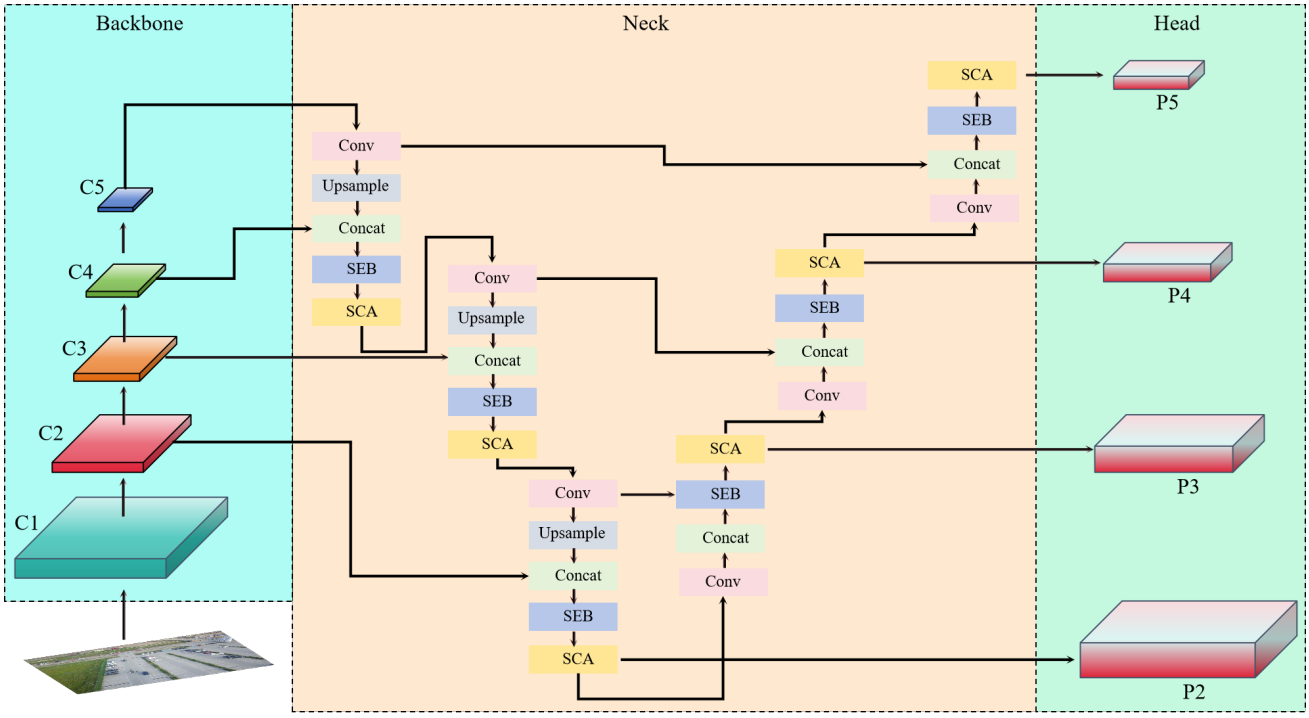


Figure 25. The general architecture diagram of SCA-YOLO.

safety during driving, Wang et al. [116] proposed YOLOv8-QSD. This method achieves better detection of small target objects at medium to long distances by integrating features of different scales.

In the defect detection of industrial products, Zhang et al. [117] proposed a detail-sensitive PAN YOLO (DsP-YOLO) detection model based on detail sensitive PAN. This method uses anchor-free YOLOv8 as the basic framework and employs a lightweight attention module to enhance the detection performance of the model. Not only can it enhance the performance of the model, but it can also provide support for its operation on devices with lower computing power.

Bai et al. [118] proposed the small object detection network based on fine-grained feature extraction and fusion YOLO (SFFEF-YOLO) method for addressing the challenge of small object detection based on UAV perspective. This method first reduces the parameter count and complexity of the model by using a smaller detection head. Then, by using a fine-grained feature information extraction module, the target can be more accurately distinguished. Subsequently, a strategy of using a multi-layer pyramid network with additional skip residual connections to branch is employed to enhance the model's recognition of small target objects.

Giri et al. [119] proposed a detection model small object-YOLOv8 (SO-YOLOv8) for small object detection. This model mainly focuses on optimizing hyperparameters and performing more advanced data augmentation on existing images to improve the performance of the model.

Hou et al. [120] proposed multi-stage feature enhancement lightweight-YOLO (MFEL-YOLO) to solve the problem of small object detection based on UAV perspective. This method mainly enhances the contextual feature information detection head by referencing efficient PAN to obtain better

information acquisition capabilities, thereby improving model performance.

6.2. SSD series small object detection models

In Section 6.1, we mainly introduced an improved UAVs-based small object detection model using the YOLO series model. In this section, we will mainly introduce some small object detection models improved on the SSD series.

Aiming to achieve both high-speed detection performance and more accurate detection results, Unel et al. [121] proposed an improved strategy based on SSD. This method introduces a more efficient PeleeNet network, which can achieve high performance even with limited computing resources, in order to efficiently detect small targets. Subsequently, they considered that the resolution of images in common datasets is usually low, while in reality, the resolution of images captured by cameras is usually high. In order to process high-definition images in a timely and fast manner in these situations, they proposed a method that utilizes tile segmentation. Simply put, this method divides high-resolution images into several equal parts. They use this method to segment high-resolution images into several smaller resolution blocks, and then use detection models to detect these small blocks before merging them. This strategy can alleviate the detection pressure of the model on high-resolution images and reduce the occurrence of many errors when detecting high-definition images. The experimental results also demonstrate the effectiveness of this method.

On the basis of F-SSD, Liang et al. [122] proposed feature fusion and scaling-based SSD (FS-SSD). In order to capture richer feature information about the original image in a more detailed manner, FS-SSD fuses the convolutional layer feature information between different levels. By using this method, richer detailed feature information and more comprehensive contextual global feature information can be obtained. Afterwards, FS-SSD added a branch with average pooling that can scale and has a deconvolution layer, while the original branch was focused on detecting small targets.

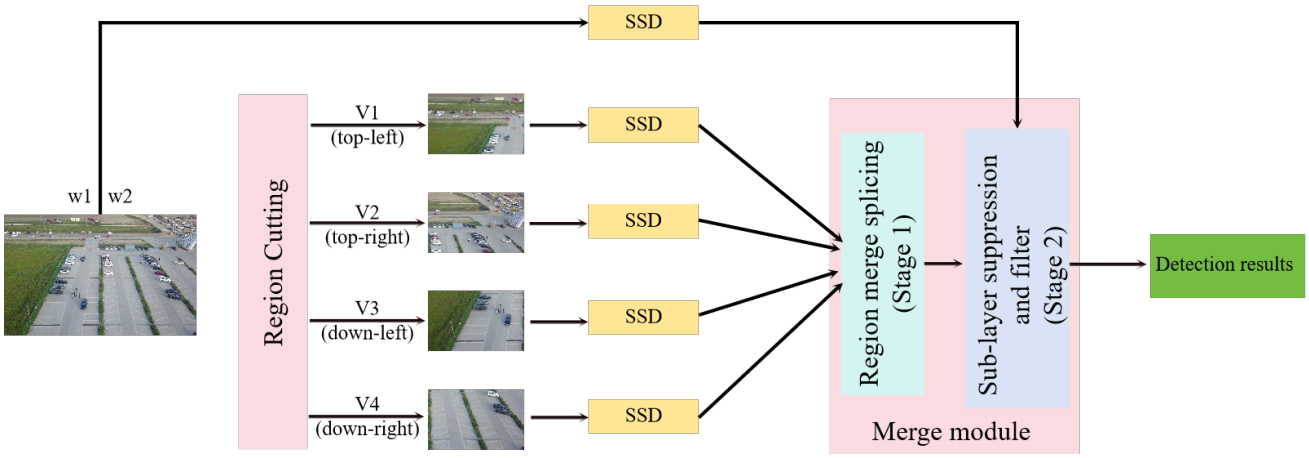


Figure 26. The general architecture diagram of multi-block SSD.

In the scenario of using drones to inspect railways, Li et al. [123] proposed a multi-block SSD method based on the SSD detection model. In order to conduct more precise inspections, this method has designed two steps for target recognition, and the general framework of this method is shown in Figure 26. First, this method detects the main image, and then divides the original image into 4

blocks. Each small block of the original image will be specially deployed with an SSD for detection and recognition. Because the large image is divided into 4 blocks, this improves the detection speed for each small block. At the same time, the method also uses NMS to eliminate overlapping boxes. It is worth noting that multi-block SSD use transfer learning to improve the problem of insufficient samples. The experimental results show that the detection accuracy of multi-block SSD has increased by 9.2% compared to the original SSD.

Song et al. [124] improved on SSD and proposed resnet self-attention detector (RSAD). In order to better extract feature information from small target objects and fully utilize the feature information in the original image, RSAD is mainly improved in two aspects. On the one hand, this method replaces the original backbone feature network with ResNet-50 to better obtain deep feature information. On the other hand, in order to better obtain important feature information from shallow and deep networks in images, they introduced a self-attention mechanism to obtain richer and more important feature vectors. The experimental results showed that the mAP of RSAD increased by 7.4% compared to SSD, indicating a significant improvement.

Nguyen et al. [125] designed a patrol detection model for partial faults on power lines, which was improved on the basis of SSD. Specifically, considering the issue of insufficient samples between datasets and imbalanced fault samples, they used data augmentation methods to amplify the samples. In addition, they used SSD based on ResNet network as the backbone as the detection model, which showed good results in the inspection of power lines.

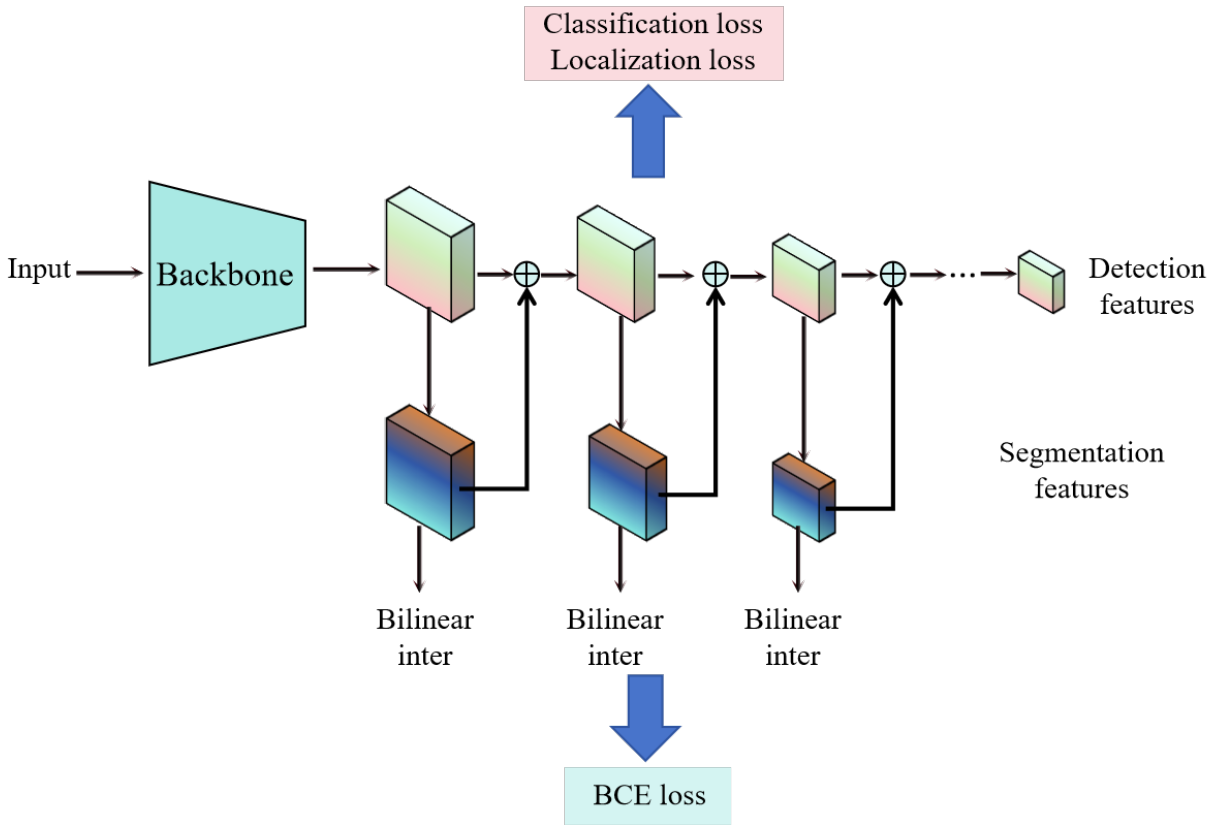


Figure 27. The general architecture diagram of Mask-SSD.

In some images, irrelevant and cluttered backgrounds often appear, which also affects the model's detection of these small target objects. To improve this issue, Sun et al. [126] proposed mask-guided SSD (mask-SSD), whose approximate framework is shown in Figure 27. This method introduces detection and segmentation branches, which fuse the features of detection and segmentation to better detect and locate the position of the target object.

Yang et al. [127] proposed FasterNet-SSD with the aim of improving the model's ability to obtain feature information of small target objects. This method uses partial convolution blocks to build the network, which not only improves the ability to obtain dimensional feature information of small target objects, but also reduces the computational complexity, achieving good results.

In order to better capture important regions of small targets in images, Gong et al. [128] proposed FCR-SSD (feature cross-reinforcement SSD). They designed an efficient channel attention to focus the model on non-background important areas in the image. Then, multi-scale information aggregation techniques are used to focus the model on important contextual information in shallow feature information.

6.3. R-CNN series small object detection models

In Sections 6.1 and 6.2, we mainly introduce small object detection methods based on YOLO series and SSD series, while in this section, we will focus on small object detection methods based on R-CNN series. The R-CNN series methods are based on two-stage object detection algorithms. As a supplement to one-stage object detection, these methods have the advantage of high detection accuracy.

In order to detect target objects in UAV images more accurately and accurately, Xiao et al. [129] improved on faster R-CNN. They designed a rotation region network (RRN). Specifically, currently most detection boxes are oriented in a single direction, which may result in insufficient fit of the detection boxes. Therefore, RRN is used to improve this problem. Similarly, in order to better process feature information, the model also deploys a network capable of extracting features from different convolutional layers to obtain multi-scale feature vectors. The experimental results on the DOTA and VisDrone datasets show that this method has strong detection performance.

Mask R-CNN has achieved certain results, but in order to meet the needs of small object detection based on remote sensing, Butler et al. [130] improved it and proposed keypoint R-CNN (KR-CNN). This method introduces additional keypoint detection by marking and locating some important or key features of small target objects, thereby making the model's localization more accurate. In addition, the model also includes a multi-branch structure, where one branch is used to process bounding boxes and the other branch is used to specifically process keypoints. This not only improves detection efficiency, but also enhances the model's detection performance for target objects of different scales and sizes.

In addition to the above application scenarios, in the field of agricultural pest prevention, Du et al. [131] proposed a Pest R-CNN detection model based on faster R-CNN to address the harm of agricultural pests to corn and other crops. The general framework of Pest R-CNN is shown in Figure 28. The pest training images used in this model were obtained by drones. In order to better identify different levels of pest infestations, the model adopts a multi-scale strategy to more accurately locate them. At the same time, in order to better locate key features, the Pest R-CNN method introduces spatial and channel attention mechanisms. Introducing attention mechanism can help the model obtain more accurate and important feature information, thereby enabling the model to make better decisions.

Similarly, based on agricultural scenarios, Machefer et al. [132] strive to better count the number

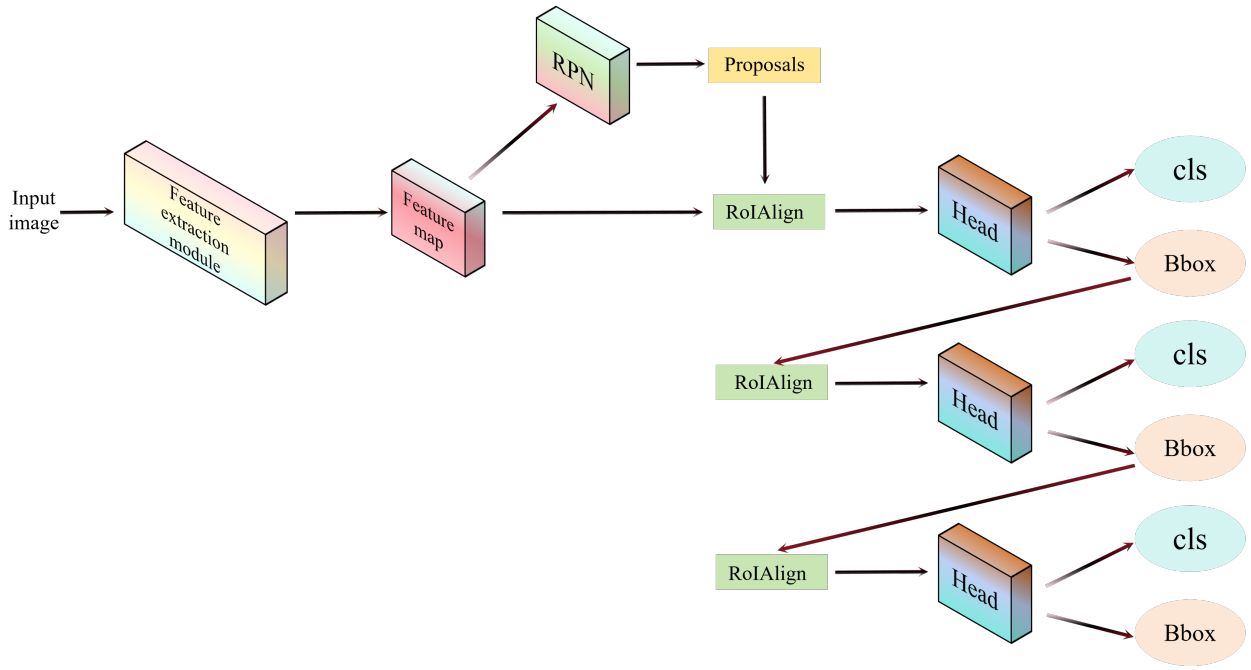


Figure 28. The general architecture diagram of pest R-CNN.

of plants in agriculture, in order to make more accurate yield predictions and subsequent planning. Machefer et al. adjusted the model based on Mask R-CNN, mainly using advanced strategies such as transfer learning to train the model. The final trained model can effectively distinguish and count different plants.

In order to improve the detection and tracking of targets in UAV background, Avola et al. [133] proposed multi-stream faster R-CNN (MS-Faster R-CNN) based on the faster R-CNN model. To address the issues of occlusion and background confusion of small targets, this model introduces the multi-stream architecture, which incorporates multi-stream at different scales to better simulate different scales. These improvements have successfully improved the detection performance and robustness of the model.

In the field of safety monitoring of railway network support devices, UAV based small object detection strategies have also played a certain role. The defects of the supporting device are usually small and difficult to observe with the naked eye, so using UAVs for close detection has become a good strategy. Liu et al. [134] used an improved model based on faster R-CNN for detection. Simply put, this method uses multi-scale feature information to increase the effective feature vector. This method integrates feature information from three scales, first using high-level convolutional layers to obtain more semantic information, and then using low-level convolutional layers to obtain richer detailed feature information, in order to improve the detection performance of targets.

To further improve the performance of the detection model, Liu et al. [135] cascaded the deployment of fast R-CNN to address detection challenges in various complex situations. Simply put, in order to better detect small objects and avoid the loss of important feature information, they first designed a multi-branch parallel feature pyramid networks (MPFPN) module to solve this problem. Similarly, in order to improve the detection performance of small target objects in complex

environments, this method also introduces a supervised spatial attention module (SSAM), which enables the model to better eliminate the influence of complex objects. This method has demonstrated good detection capability in some unmanned aerial vehicle small object datasets.

Zhang et al. [136] strive to improve the impact caused by changes in perspective and images of different resolutions, and achieve more accurate small object detection. They have made improvements from multiple aspects. On the one hand, in order to better extract feature information about the target object in the image, they added the backbone feature network of DeForm convolution layers for feature information extraction. Subsequently, for the challenge of densely populated small target areas, Zhang et al. used a cascaded network to distinguish and detect these densely clustered small target objects from coarse to fine. At the same time, with the aim of fully utilizing limited image data, they used equal cropping to crop and enlarge the original image into five images including the original image.

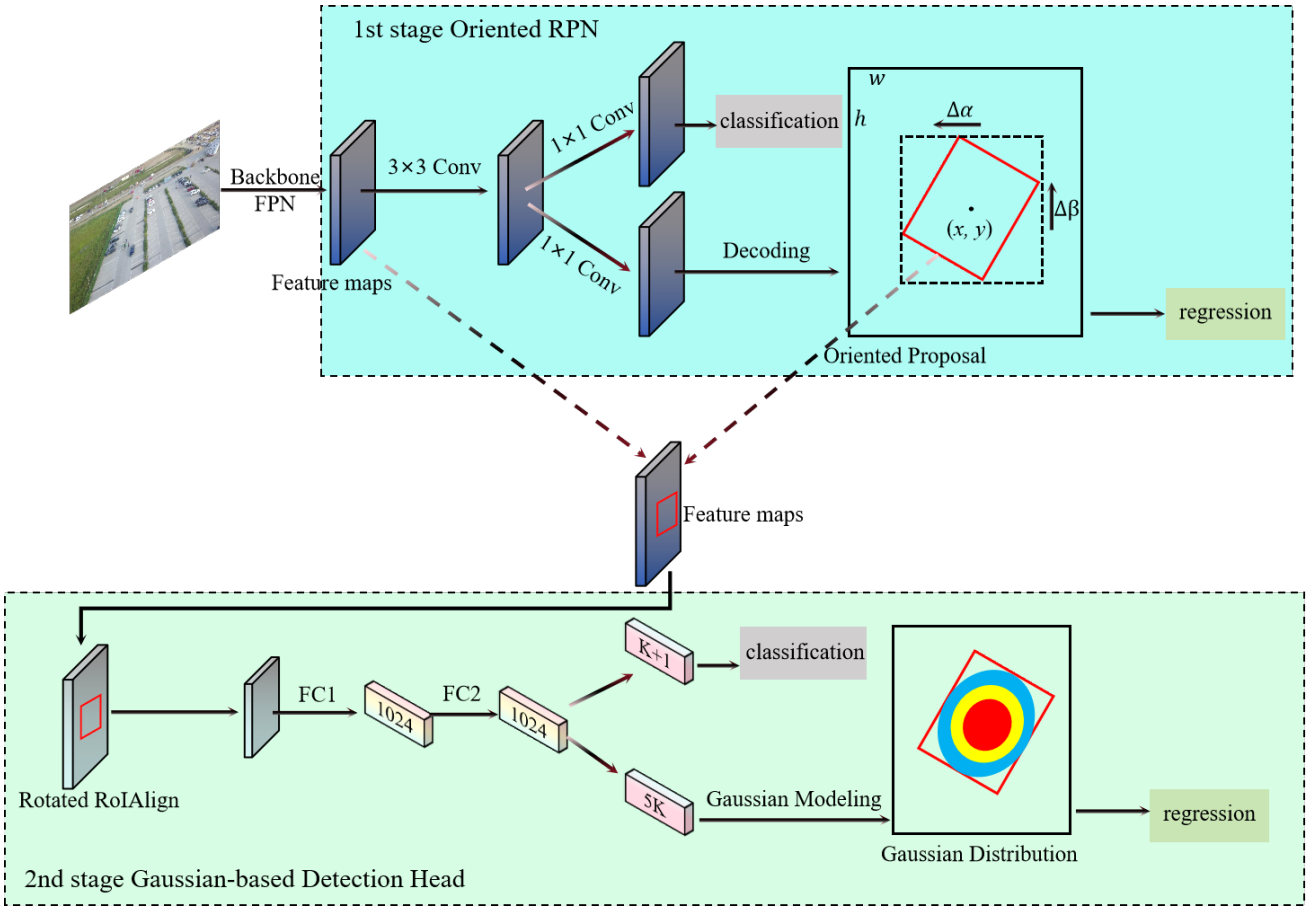


Figure 29. The general architecture diagram of Gaussian-based R-CNN.

In response to the challenge of detecting small targets in synthetic aperture radar images, Kamirul et al. [137] proposed the R-sparse R-CNN method. This method designs a dual context pooling module to enhance contextual feature connections regarding ship features. Then, the improved Transformer module is used to interact with the feature background of the ship and the proposed feature box.

Yang et al. [138] proposed a Gaussian-based R-CNN method for detecting small rotating targets, and its general framework is shown in Figure 29. This method addresses the interference of complex

backgrounds on rotating targets and proposes a strategy that integrates the alignment of rotating regions of interest and directional region proposal networks. This strategy will transform the bounding box into a Gaussian distribution, thereby improving the difficulty of angle parameter regression. Afterwards, the attention mechanism is introduced to enhance the model's ability to obtain information about the context surrounding the target object.

In order to strengthen the security inspection of airports, He et al. [139] proposed airport scene faster-RCNN (AS-Faster-RCNN). This method uses a network with stronger feature extraction capabilities as the backbone network to improve the detection of small objects. Then, variable convolutional network is used to better perceive deformed targets.

6.4. Other named two-stage small object detection methods

In Section 6.4, we will continue to introduce some excellent two-stage small object detection methods.

Aiming to better detect surface targets at low altitudes, Mittal et al. [140] improved the faster R-CNN model and proposed dilated convolution-based RCNN using feature fusion (DCRFF). The model mainly incorporates the dilated residual module (DRM) and feature fusion module (FFM). The DRM module can obtain feature information of different sizes or scales, and by setting the dilation convolution parameters, it can obtain more key feature information. The FFM module can obtain semantic information from different convolutional layers, and then aggregate these feature information to generate vector information with more feature information. The model demonstrated excellent performance in the VisDrone dataset from a drone perspective.

Yang et al. [141] proposed the ClusDet detection model to address issues such as small target objects and imbalanced distribution in aerial images. This model generates cluster regions about the target object in the image by introducing cluster proposal sub-network. Then use estimation sub-network to roughly estimate the size of the cluster. Through this strategy, ClusDet has achieved more accurate detection of small targets.

To better address the problem of difficult detection in small target images due to small sample sizes, Li et al. [142] proposed the method density-map guided object detection network (DMNet), which uses density maps in images to guide models for detection. This method draws on some modules of the R-CNN series for improvement. On the one hand, this method first analyzes the density of some target objects in the detected image, in order to determine whether there are target objects that need to be detected in that area, and guide the model to detect them. On the other hand, after the density map generation module takes effect, the image cropping module will crop these areas with a certain density, thereby prompting the detection model to perform more efficient detection on them.

Considering the presence of long-tailed phenomenon in small sample UAV datasets, the imbalance between samples in the dataset may lead to suboptimal model detection performance. Yu et al. [143] improved faster R-CNN and proposed a method called DSHNet (dual sampler and head detection network). In order to better address the challenge posed by the imbalanced number of head and tail classes in long-tailed datasets, this method specifically designs two sampling heads to handle this issue.

6.5. Summary of typical application scenarios

In Section 6, we provide many application cases of small object detection, including but not limited to UAV monitoring, traffic scene monitoring, agricultural monitoring, etc. In Section 6.5, we will briefly summarize some typical application scenarios.

UAV monitoring: In mountain rescue, Sambolek et al. [106] achieved more accurate detection using YOLOv4. In railway inspection scenarios, the multi-block SSD [123] method achieves more accurate and efficient inspections. In forest monitoring, LDS-YOLO [108] achieves more accurate monitoring and can detect smaller targets.

Traffic scene monitoring: Sun et al. [114] used a YOLOv3 based method for object detection in transportation systems. YOLOv8-QSD [116] achieves better detection of small target objects in intelligent driving scenarios by better integrating feature information at different scales.

Agricultural monitoring: Gallo et al. [110] used an improved YOLOv7 for precise detection of weeds in agricultural planting. Pest R-CNN [131] improves the faster R-CNN model to accurately detect pests in corn crops. Junos et al. [112] used an improved YOLOv3-tiny to detect small target oil palm trees.

7. Performance display of some methods

Overall, both general object detection technology and small object detection technology have achieved good development over the years. In order to demonstrate more specifically the performance improvement of some excellent models, we present in this section a series of performance demonstrations of advanced model methods on some classic datasets.

7.1. Performance display of object detection methods

We chose the classic MS-COCO dataset and demonstrated the performance of the most classic YOLO series methods, as shown in Table 3. The performance of mAP has increased from 28.2% of YOLOv3 to 54.4% of YOLOv10-X. In addition, we can also see in the performance demonstration of mAP50 that the performance has increased from 51.5% of YOLOv3 to 72.8% of YOLOv9-E, which is undoubtedly a significant improvement. From Table 3, we can still observe that the YOLO series has achieved good performance over time for models of different scales. Moreover, models of different sizes can adapt to the usage needs in different environments.

7.2. Performance display of small object detection methods

We chose the classic VisDrone dataset to demonstrate the performance of the small object detection model, and the experimental results are presented in Table 4. Due to the need for small object detection methods to be applicable to devices with average computing power, there are certain limitations on the parameter size of the model. From Table 4, we can see that the parameter size of YOLOv3-tiny is 14.3 M, and its mAP and mAP50 performance are 13.2% and 23.6%, respectively. The parameter count in YOLOv11n is only 6.3 M, but its mAP and mAP50 performance have improved significantly to 18.6% and 32.2%, respectively. Overall, with the development of technology, many lightweight detection models not only have lower parameter counts, but also have significant performance improvements.

Table 3. Performance demonstration of some classic and advanced methods in the YOLO series on the MS-COCO dataset. "a" indicates that the data is cited from paper [53], "b" indicates that the data is cited from paper [56], "c" indicates that the data is cited from paper [59], and "d" indicates that the data is cited from paper [60].

Method	Input Size	Publication(Year)	mAP(%)	mAP50(%)
YOLOv3 [a] [52]	320	arXiv (2018)	28.2	51.5
YOLOv3 [a] [52]	416	arXiv (2018)	31.0	55.3
YOLOv3 [a] [52]	608	arXiv (2018)	33.0	57.9
YOLOv4[a] [53]	416	arXiv (2020)	41.2	62.8
YOLOv4[a] [53]	512	arXiv (2020)	43.0	64.9
YOLOv4[a] [53]	608	arXiv (2020)	43.5	65.7
YOLOv5-N [b] [55]	640	– (2020)	28.0	45.7
YOLOv5-S [b] [55]	640	– (2020)	37.4	56.8
YOLOv5-M [b] [55]	640	– (2020)	45.4	64.1
YOLOv5-L [b] [55]	640	– (2020)	49.0	67.3
YOLOX-Tiny [b] [144]	416	arXiv (2021)	32.8	50.3
YOLOX-S [b] [144]	640	arXiv (2021)	40.5	59.3
YOLOX-M [b] [144]	640	arXiv (2021)	46.9	65.6
YOLOX-L [b] [144]	640	arXiv (2021)	49.7	68.0
PPYOLOE-S [b] [145]	640	arXiv (2022)	43.1	59.6
PPYOLOE-M [b] [145]	640	arXiv (2022)	49.0	65.9
PPYOLOE-L [b] [145]	640	arXiv (2022)	51.4	68.6
YOLOv6-T [b] [56]	640	arXiv (2022)	40.3	56.6
YOLOv6-M [b] [56]	640	arXiv (2022)	49.5	66.8
YOLOv6-L [b] [56]	640	arXiv (2022)	52.5	70.0
YOLOv7-Tiny [b] [57]	416	CVPR (2023)	33.3	49.9
YOLOv7-Tiny [b] [57]	640	CVPR (2023)	37.4	55.2
YOLOv7 [b] [57]	640	CVPR (2023)	51.2	69.7
YOLO-MS-N [c] [146]	–	TPAMI (2025)	43.4	60.4
YOLO-MS-S [c] [146]	–	TPAMI (2025)	46.2	63.7
YOLO-MS [c] [146]	–	TPAMI (2025)	51.0	68.6
YOLOv8-N [c] [58]	–	– (2023)	37.3	52.6
YOLOv8-M [c] [58]	–	– (2023)	50.2	67.2
YOLOv8-X [c] [58]	–	– (2023)	53.9	71.0
YOLOv9-S [c] [59]	–	ECCV (2024)	46.8	63.4
YOLOv9-M [c] [59]	–	ECCV (2024)	51.4	68.1
YOLOv9-E [c] [59]	–	ECCV (2024)	55.6	72.8
YOLOv10-N [d] [60]	–	NIPS (2024)	38.5	–
YOLOv10-S [d] [60]	–	NIPS (2024)	46.3	–
YOLOv10-M [d] [60]	–	NIPS (2024)	51.1	–
YOLOv10-L [d] [60]	–	NIPS (2024)	53.2	–
YOLOv10-X [d] [60]	–	NIPS (2024)	54.4	–

Table 4. Performance demonstration of some classic and advanced small object detection methods on the VisDrone dataset. "a" indicates that the data is cited from paper [151], "b" indicates that the data is cited from paper [152], and "c" indicates that the data is cited from paper [161].

Method [a]	Publication(Year)	mAP50(%)	mAP(%)	Parameters(M)
YOLOv3-tiny [a] [52]	arXiv (2018)	23.6	13.2	14.3
YOLOv5n [a] [55]	– (2020)	32.9	19.1	5.8
YOLOv5s [a] [55]	– (2020)	39.3	23.4	18.8
YOLOv5l [a] [55]	– (2020)	41.4	24.6	107.8
YOLOv6s [a] [56]	arXiv (2022)	17.7	11.5	4.15
YOLOv7-tiny [a] [57]	CVPR (2023)	35.8	18.8	13.3
YOLOv8n [a] [58]	– (2023)	33.1	19.2	6.8
YOLOv8s [a] [58]	– (2023)	39.1	23.4	23
YOLOv8m [a] [58]	– (2023)	42.5	26	67.5
YOLOv9s [a] [59]	ECCV (2024)	39.4	23.8	22.1
YOLOv10n [a] [60]	NIPS (2024)	34.5	19.9	6.5
YOLOv10s [a] [60]	NIPS (2024)	39.8	23.8	21.4
YOLOv10m [a] [60]	NIPS (2024)	44.2	26.9	58.9
YOLOv11n [a] [61]	arXiv (2024)	32.2	18.6	6.3
YOLOv11s [a] [61]	arXiv (2024)	39.4	23.6	21.3
YOLOv11m [a] [61]	arXiv (2024)	44.1	27.2	67.7
RT-DETR(r18) [a] [147]	CVPR (2024)	41.4	25.1	57
EL-YOLO [a] [148]	ESWA (2024)	42.9	24.8	6.7
EBC-YOLO [a] [149]	Earth Sci. Inf. (2025)	44.3	26.7	35.5
YOLOv8-QSD [a] [116]	TIM (2024)	34.6	16.8	–
EdgeYOLO [a] [150]	CCC (2023)	44.8	26.4	–
CF-YOLO [a] [151]	Sci. Rep. (2025)	44.9	27.5	23.9
AAPW-YOLO [b] [152]	Sci. Rep. (2025)	38.6	22.4	2.12
ClusDet [c] [141]	ICCV (2019)	56.2	32.4	–
DREN [c] [153]	ICCVW (2019)	–	30.3	–
DMNet [c] [142]	CVPRW (2020)	49.3	29.4	–
AMRNet [c] [154]	arXiv (2020)	–	32.1	
CDMNet [c] [155]	ICCV (2021)	51.3	30.7	–
GLSAN [c] [156]	TIP (2021)	55.8	32.5	–
HRDNet [c] [157]	arXiv (2020)	49.25	28.33	–
TOOD+SAHI [c] [158]	ICIP (2022)	43.5	–	–
QueryDet [c] [159]	CVPR (2022)	48.14	28.32	–
CEASC [c] [160]	CVPR (2023)	50.7	28.7	–
YOLC [c] [161]	TITS (2024)	63.7	39.6	–

Table 5. Quantitative experiments on representative methods for different structures in the VOC dataset. "e" indicates that the data is sourced from paper [113].

Method	Backbone	Train	Test	Size	mAP50 (%)	FPS
Faster R-CNN [e]	ResNet101	VOC2007+2012	VOC2007	600×1000	76.4	2.4
Faster R-CNN [e]	VGG16	VOC2007+2012	VOC2007	600×1000	73.2	7
SSD [e]	VGG16	VOC2007+2012	VOC2007	300×300	77.2	45.5
SSD [e]	VGG16	VOC2007+2012	VOC2007	512×512	78.7	38.6
FD-SSD [e]	VGG-16	VOC2007+2012	VOC2007	300×300	79.1	12.6
YOLO [e]	GoogleNet-9	VOC2007+2012	VOC2007	448×448	63.4	45
YOLOv5 [e]	CSPDarkNet-53	VOC2007+2012	VOC2007	512×512	85.1	163.9
SCA-YOLO [e]	CSPDarkNet-53	VOC2007+2012	VOC2007	512×512	85.4	232.5

7.3. Quantitative comparison of representative methods with different structures

In Table 5, based on the VOC dataset, we present performance comparisons of representative methods in some different structures. It can be observed that although the two-stage method faster R-CNN has better mAP50 performance, its FPS is extremely low, making it difficult to adapt to low-latency scenarios. Although the FPS of the SSD series has significantly improved compared to the two-stage faster R-CNN method, there is still a considerable gap compared to the classic YOLO series method.

8. Current achievements, challenges, and prospects for the future

In general, current object detection and small object detection methods have achieved good results and have significant applications in multiple fields. However, they also face many shortcomings and challenges. In future work, further improvements can be made to achieve better performance. In this section, we review the object detection and small object detection algorithms, as well as the challenges faced and prospects for future research directions.

8.1. Current achievements

Object detection: the current universal object detection technology has achieved considerable success after years of development.

The ability to represent the features of the target object continues to improve. From early feature extraction networks such as VGGNet and AlexNet as the backbone, to ResNet networks with residual components (mainly improving the gradient vanishing problem in model training), and then to various network models with attention mechanisms. The continuous improvement of the backbone network provides better front-end support for object detection models, enabling them to have better feature information extraction capabilities.

Significant improvement in detection accuracy and depth. The detection accuracy continues to improve in some mainstream datasets, such as PASCAL VOC and MS-COCO datasets. At the same time, there has been significant progress in tasks that require both real-time performance and accuracy, such as autonomous driving detection and security check detection.

Enhanced adaptability in detecting complex scenes. By introducing attention mechanisms, data

augmentation techniques, and fusion techniques for features at different scales, the performance of the model can be improved.

Small object detection: based on general object detection, significant progress has also been made in small object detection technology.

For specific small object scenarios, the model utilizes different strategies for improvement. For example, in scenes where the target object is extremely small and dense, different convolution blocks (such as Deformable Conv) can be used to capture more complex morphological changes about the target. In scenes with low image resolution, the low-level feature preservation ability of the model is strengthened to alleviate the severe loss of target object feature information in deep networks. In some devices with limited computing power, the detection model uses some improved lightweight networks (enhancing its feature extraction ability by adding attention mechanisms, etc.) as the backbone network. While ensuring the lightweighting of the model, it can also provide a certain level of detection performance.

The practical application of detection models in specific scenarios has been accelerated, with a significant increase in the number of actual application scenarios. For example, the application of agricultural pest monitoring, power inspection, security inspection and other scenarios has increased.

8.2. Current challenges

Object detection: although general object detection technology has made some progress, it also has some shortcomings.

Most models suffer from poor robustness in extremely complex environments. In general, the detection performance of the detection model for target objects will significantly decrease under conditions such as poor brightness and lighting, excessive occlusion, and adverse weather conditions.

In some datasets, there is a significant imbalance in the number of target objects between samples. This is manifested in the fact that some target objects have an extremely large number, while others have a very small number, exhibiting a long-tailed phenomenon. So using this dataset to train a model will result in higher performance detection for high-frequency objects and poorer performance for low-frequency objects.

Some detection models only pursue high performance, but ignore the adaptation to devices with lower computing power. Although the performance of some models is constantly improving, these methods are difficult to deploy on devices with lower computing power. This also limits the deployment of high-performance models in practical applications.

Most datasets typically use manual annotation to label target objects. This is not only time-consuming and laborious, but may also lead to the occurrence of incorrect labeling. Incorrect labeling not only misleads the model, but may also limit further improvements in model performance.

Small object detection: the development of small object detection has achieved good results so far, but there are still some shortcomings.

Some models have poor representation ability for small target objects. Small target objects often have low pixels, making it difficult for models to obtain accurate feature representations of the target object from a very small number of pixels. Due to this issue, it is difficult for the model to distinguish the target object from the background area, resulting in relatively high false detection and missed detection rates.

In images with small target objects, there are often a large number of small target objects present.

Very small target objects often have only a few pixels, while larger ones have over ten to twenty pixels, with a significant difference between the two. This also makes it difficult for the model to adaptively detect these small target objects. In addition, the distribution of small target objects between different images is also relatively uneven, which further limits the performance improvement of the model.

Due to the small size of small target objects in the image, the target objects are too dense, which may result in incorrect labeling (such as boundary box labeling errors) during annotation. The use of such incorrectly labeled images can also lead to poor generalization performance of the trained model.

In order to enhance the detection performance of the model, it is often necessary to add multi-scale feature aggregation and efficient attention mechanisms. Although these strategies enhance the performance of the model, the addition of additional modules also reduces the real-time detection capability of the model.

8.3. Prospects for the future

Object detection: in response to the challenges and shortcomings in general object detection technology, we have made some prospects for its future development.

In some current datasets, the number of target objects for certain categories is relatively uneven. This also leads to lower recognition performance of the model for target object categories with lower frequencies. To improve this issue, we may use diffusion models and generative adversarial networks (GANs) to generate images related to low-frequency categories.

The scalability and interpretability of some methods are still weak. In the future, the scalability and interpretability of the current model can be improved through more advanced end-to-end modules.

In some cases, a single modal interaction is difficult to meet the continuous improvement of model performance. In future work, it may be possible to combine different modal data such as images, point clouds, and text to improve the performance of detection models.

The current object detection models rely heavily on a large number of high-quality samples and accurately labeled datasets, but this is undoubtedly too time-consuming and laborious. In the following work, efficient few-shot or zero-shot learning methods can be explored to reduce the demand for a large number of images and high-quality annotated datasets.

In the following work, while pursuing model performance, it is also necessary to further reduce the number of parameters in the model, so that it can be deployed on devices with lower computing power.

Small object detection: small object detection technology has made good progress, but overall there are also some challenges and shortcomings. In order to better develop small object detection, we have made some expectations.

In some low pixel images, the features of some target objects are relatively weak, which can enhance the model's perception ability of weak features. Specifically, super-resolution technology can be combined to improve the perception ability of some weak features (such as extremely small target objects or blurry phenomena).

To address the issue of insufficient data for some target objects, image data synthesis techniques can be used to generate new samples of scarce samples, in order to compensate for the problem of insufficient data and images. However, simple data augmentation techniques can only generate raw images and are difficult to generate high-quality new samples. In the following work, more powerful data generators can be utilized to assist in generating higher quality image samples.

Although some current small object detection algorithms have achieved good detection results, their

parameter quantities are still large and cannot be applied to hardware devices with limited computing power. In the following work, specialized technologies such as neural network accelerators can be developed specifically for these devices.

The implementation of algorithms in different scenarios requires consideration of different scenarios and devices. For example, specialized equipment is required for unmanned aerial vehicle inspection, remote sensing detection, and medical imaging detection. In the future, customized small-scale object detection solutions can be developed to promote the deployment of models.

9. Conclusions

In this review, we give a systematic overview of general object detection and small object detection technologies. It should be noted that most of the references selected in this paper are deep learning, object detection, small object detection, and some inspiring articles. We focus on general object detection (using public data sets) and small object detection scenarios from the UAV perspective.

In addition, in this review, we have redrawn the general framework of some classic or advanced algorithms, so that readers can better understand these methods. In general, we first introduce the datasets and performance indicators that need to be used in object detection. Then, we introduce the general object detection algorithm systematically. These methods can be applied to subsequent small object detection with some improvements. Furthermore, we systematically introduce the application of small object detection in some practical scenarios. Finally, we review the achievements, shortcomings and possible future research directions of object detection and small object detection.

In the future, the development direction of general object detection and small object detection technology may start from the following points. The detection model can try to fuse data information between different modes such as vision, point cloud and text, which can not only enhance the model's ability to understand feature information in complex scenes, but also provide more information for small object objects. In addition, it may be considered to introduce domain adaptation technology, which can improve the diversity of data distribution in multiple scenarios, and thus improve the generalization performance of the model across different scenarios. Furthermore, the detection model can also use the semi-supervised learning technology to improve the dependence of the model on the large-scale annotated dataset. In addition, we believe that we can also explore more efficient end learning strategies, and further explore lighter architectures and more efficient reasoning strategies, so as to further promote the implementation and deployment of models.

Use of AI tools declaration

The authors declare they have not used artificial intelligence (AI) tools in the creation of this article.

Acknowledgments

This study is funded by the Key Project of Anhui Provincial Natural Science Foundation (2025AHGXZK10001) and the National Natural Science Foundation of China (61773415).

Conflict of interest

The authors declare there is no conflicts of interest.

References

1. G. Zhang, J. Chen, G. Gao, J. Li, S. Liu, X. Hu, SAFDNet: A simple and effective network for fully sparse 3d object detection, in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2024), 14477–14486. https://doi.org/10.1109/CVPR52733.2024.01372
2. C. M. Badgujar, A. Poulose, H. Gan, Agricultural object detection with you only look once (YOLO) algorithm: A bibliometric and systematic literature review, *Comput. Electron. Agric.*, **223** (2024), 109090. https://doi.org/10.1016/j.compag.2024.109090
3. Z. Liu, J. Hou, X. Wang, X. Ye, J. Wang, H. Zhao, et al., Lion: Linear group rnn for 3d object detection in point clouds, *Adv. Neural Inf. Process. Syst.*, **37** (2024), 13601–13626. https://doi.org/10.52202/079017-0435
4. W. Lim, K. S. C. Yong, B. T. Lau, C. C. L. Tan, Future of generative adversarial networks (GAN) for anomaly detection in network security: A review, *Comput. Secur.*, **139** (2024), 103733. https://doi.org/10.1016/j.cose.2024.103733
5. S. Islam, M. T. Aziz, H. R. Nabil, J. R. Jim, M. F. Mridha, M. M. Kabir, et al., Generative adversarial networks (GANs) in medical imaging: Advancements, applications, and challenges, *IEEE Access*, **12** (2024), 35728–35753. https://doi.org/10.1109/ACCESS.2024.3370848
6. W. Zeng, H. Zhu, C. Lin, Z. Xiao, A survey of generative adversarial networks and their application in text-to-image synthesis, *Electron. Res. Arch.*, **31** (2023), 7142–7181. https://doi.org/10.3934/era.2023362
7. R. Archana, P. S. E. Jeevaraj, Deep learning models for digital image processing: A review, *Artif. Intell. ReV.*, **57** (2024), 11. https://doi.org/10.1007/s10462-023-10631-z
8. F. Gao, X. Jin, X. Zhou, J. Dong, Q. Du, MSFMamba: Multi-scale feature fusion state space model for multi-source remote sensing image classification, *IEEE Trans. Geosci. Remote Sens.*, **63** (2025), 1–16. https://doi.org/10.1109/TGRS.2025.3535622
9. L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, Y. Miao, Review of image classification algorithms based on convolutional neural networks, *Remote Sens.*, **13** (2021), 4712. https://doi.org/10.3390/rs13224712
10. M. Wankhade, A. C. S. Rao, C. Kulkarni, A survey on sentiment analysis methods, applications, and challenges, *Artif. Intell. Revi.*, **55** (2022), 5731–5780. https://doi.org/10.1007/s10462-022-10144-1
11. K. L. Tan, C. P. Lee, K. M. Lim, A survey of sentiment analysis: Approaches, datasets, and future research, *Appl. Sci.*, **13** (2023), 4550. https://doi.org/10.3390/app13074550
12. A. S. Talaat, Sentiment analysis classification system using hybrid BERT models, *J. Big Data*, **10** (2023), 110. https://doi.org/10.1186/s40537-023-00781-w
13. A. Shamshiri, K. R. Ryu, J. Y. Park, Text mining and natural language processing in construction, *Autom. Constr.*, **158** (2024), 105200. https://doi.org/10.1016/j.autcon.2023.105200

14. S. Feuerriegel, A. Maarouf, D. Bär, D. Geissler, J. Schweisthal, N. Pröllochs, et al., Using natural language processing to analyse text data in behavioural science, *Nat. Rev. Psychol.*, **4** (2025), 96–111. https://doi.org/10.1038/s44159-024-00392-z

15. A. P. Wibawa, F. Kurniawan, Advancements in natural language processing: Implications, challenges, and future directions, *Telemat. Inf. Rep.*, **16** (2024), 100173. https://doi.org/10.1016/j.teler.2024.100173

16. Y. Zhang, Z. Shen, R. Jiao, Segment anything model for medical image segmentation: Current applications and future directions, *Comput. Biol. Med.*, **171** (2024), 108238. https://doi.org/10.1016/j.compbiomed.2024.108238

17. M. E. Rayed, S. M. S. Islam, S. I. Niha, J. R. Jim, M. M. Kabir, M. F. Mridha, Deep learning for medical image segmentation: State-of-the-art advancements and challenges, *Inf. Med. Unlocked*, **47** (2024), 101504. https://doi.org/10.1016/j.imu.2024.101504

18. F. Isensee, T. Wald, C. Ulrich, M. Baumgartner, S. Roy, K. Maier-Hein, et al., Nnu-net revisited: A call for rigorous validation in 3d medical image segmentation, in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (2024), 488–498. https://doi.org/10.1007/978-3-031-72114-4_47

19. W. Zeng, Z. Xiao, MinoritySalMix and adaptive semantic weight compensation for long-tailed classification, *Image Vision Comput.*, **152** (2024), 105307. https://doi.org/10.1016/j.imavis.2024.105307

20. W. Zeng, Z. Xiao, Enhancing long-tailed classification via multi-strategy weighted experts with hybrid distillation, *Multimedia Syst.*, **31** (2025), 38. https://doi.org/10.1007/s00530-024-01635-y

21. S. Fan, Z. Chai, Z. Fang, Y. Pan, H. Shen, X. Cheng, et al., MaxSwap-Enhanced Knowledge Consistency Learning for long-tailed recognition, *Image Vision Comput.*, **161** (2025), 105643. https://doi.org/10.1016/j.imavis.2025.105643

22. W. Zeng, M. Li, Leveraging multi-strategy labels for long-tailed classification, *Eng. Appl. Artif. Intell.*, **166** (2026), 113563. https://doi.org/10.1016/j.engappai.2025.113563

23. Y. Tian, C. Chen, K. Sagoe-Crentsil, J. Zhang, W. Duan, Intelligent robotic systems for structural health monitoring: Applications and future trends, *Automat. Constr.*, **139** (2022), 104273. https://doi.org/10.1016/j.autcon.2022.104273

24. C. Gao, G. Wang, W. Shi, Z. Wang, Y. Chen, Autonomous driving security: State of the art and challenges, *IEEE Internet Things J.*, **9** (2022), 7572–7595. 10.1109/JIOT.2021.3130054

25. S. A. H. Mohsan, M. A. Khan, F. Noor, I. Ullah, M. H. Alsharif, Towards the unmanned aerial vehicles (UAVs): A comprehensive review, *Drones*, **6** (2022), 147. https://doi.org/10.3390/drones6060147

26. A. A. Laghari, A. K. Jumani, R. A. Laghari, H. Nawaz, Unmanned aerial vehicles: A review, *Cognit. Rob.*, **3** (2023), 8–22. https://doi.org/10.1016/j.cogr.2022.12.004

27. S. A. H. Mohsan, N. Q. H. Othman, Y. Li, M. H. Alsharif, M. A. Khan, Unmanned aerial vehicles (UAVs): Practical aspects, applications, open challenges, security issues, and future trends, *Intell. Serv. Rob.*, **16** (2023), 109–137. https://doi.org/10.1007/s11370-022-00452-4

28. Z. Wu, Y. Peng, W. Wang, Deep learning-based unmanned aerial vehicle detection in the low altitude clutter background, *IET Signal Process.*, **16** (2022), 588–600. https://doi.org/10.1049/sil2.12133

29. Y. Luo, X. Yu, D. Yang, B. Zhou, A survey of intelligent transmission line inspection based on unmanned aerial vehicle, *Artif. Intell. Rev.*, **56** (2023), 173–201. https://doi.org/10.1007/s10462-022-10189-2

30. M. Lyu, Y. Zhao, C. Huang, H. Huang, Unmanned aerial vehicles for search and rescue: A survey, *Remote Sens.*, **15** (2023), 3266. https://doi.org/10.3390/rs15133266

31. G. Cheng, X. Yuan, X. Yao, K. Yan, Q. Zeng, X. Xie, et al., Towards large-scale small object detection: Survey and benchmarks, *IEEE Trans. Pattern Anal. Mach. Intell.*, **45** (2023), 13467–13488. https://doi.org/10.1109/TPAMI.2023.3290594

32. Z. Q. Zhao, P. Zheng, S. Xu, X. Wu, Object detection with deep learning: A review, *IEEE Trans. Neural Networks Learn. Syst.*, **30** (2019), 3212–3232. https://doi.org/10.1109/TNNLS.2018.2876865

33. Z. Zou, K. Chen, Z. Shi, Y. Guo, J. Ye, Object detection in 20 years: A survey, *Proc. IEEE*, **111** (2023), 257–276. https://doi.org/10.1109/JPROC.2023.3238524

34. A. B. Amjoud, M. Amrouch, Object detection using deep learning, CNNs and vision transformers: A review, *IEEE Access*, **11** (2023), 35479–35516. https://doi.org/10.1109/ACCESS.2023.3266093

35. Y. Liu, P. Sun, N. Wergeles, Y. Shang, A survey and performance evaluation of deep learning methods for small object detection, *Exp. Syst. Appl.*, **172** (2021), 114602. https://doi.org/10.1016/j.eswa.2021.114602

36. W. Wei, Y. Cheng, J. He, X. Zhu, A review of small object detection based on deep learning, *Neural Comput. Appl.*, **36** (2024), 6283–6303. https://doi.org/10.1007/s00521-024-09422-6

37. T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, et al., Microsoft coco: Common objects in context, in *European Conference on Computer Vision (ECCV)*, (2014), 740–755. https://doi.org/10.1007/978-3-319-10602-1_48

38. M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge: A retrospective, *Int. J. Comput. Vision*, **111** (2015), 98–136. https://doi.org/10.1007/s11263-014-0733-5

39. D. Du, P. Zhu, L. Wen, X. Bian, H. Lin, Q. Hu, et al., VisDrone-DET2019: The vision meets drone object detection in image challenge results, in *Proceedings of the IEEE/CVF international conference on computer vision workshops (ICCVW)*, (2019), 213–226. https://doi.org/10.1109/ICCVW.2019.00030

40. G. S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, et al., DOTA: A large-scale dataset for object detection in aerial images, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2018), 3974–3983. https://doi.org/10.1109/CVPR.2018.00418

41. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner,Gradient-based learning applied to document recognition, in *Proceedings of the IEEE*, **86** (1998), 2278–2324. https://doi.org/10.1109/5.726791

42. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, preprint, arxiv:1409.1556. https://doi.org/10.48550/arXiv.1409.1556
43. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016), 770–778. https://doi.org/10.1109/CVPR.2016.90
44. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, et al., Mobilenets: Efficient convolutional neural networks for mobile vision applications, preprint, arxiv:1704.04861. https://doi.org/10.48550/arXiv.1704.04861
45. J. He, L. Li, J. Xu, C. Zheng, ReLU deep neural networks and linear finite elements, *J. Comput. Math.*, **38** (2020), 502–527. https://doi.org/10.52202/079017-0435
46. D. Misra, Mish: A self regularized non-monotonic activation function, preprint, arxiv:1908.08681. https://doi.org/10.48550/arXiv.1908.08681
47. J. Xu, Z. Li, B. Du, M. Zhang, J. Liu, Reluplex made more practical: Leaky ReLU, in *2020 IEEE Symposium on Computers and communications (ISCC)*, (2020), 1–7. https://doi.org/10.1109/ISCC50000.2020.9219587
48. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, et al., Imagenet large scale visual recognition challenge, *Int. J. Comput. Vision*, **115** (2015), 211–252. https://doi.org/10.1007/s11263-015-0816-y
49. X. Yu, Y. Gong, N. Jiang, Q. Ye, Z. Han, Scale match for tiny person detection, in *2020 Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, (2020), 1246–1254. https://doi.org/10.1109/WACV45572.2020.9093394
50. J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in *2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016), 779–788. https://doi.org/10.1109/CVPR.2016.91
51. J. Redmon, A. Farhadi, YOLO9000: better, faster, stronger, in *2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017), 6517–6525. https://doi.org/10.1109/CVPR.2017.690
52. J. Redmon, A. Farhadi, Yolov3: An incremental improvement, preprint, arxiv:1804.02767.
53. A. Bochkovskiy, C. Y. Wang, H. Y. M. Liao, Yolov4: Optimal speed and accuracy of object detection, preprint, arXiv:2004.10934. https://doi.org/10.48550/arXiv.2004.10934
54. W. Zeng, Image data augmentation techniques based on deep learning: A survey, *Math. Biosci. Eng.*, **21** (2024), 6190–6224. https://doi.org/10.3934/mbe.2024272
55. G. Jocher, *Yolov5*, Available from: https://github.com/ultralytics/yolov5, 2020.
56. C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, et al., YOLOv6: A single-stage object detection framework for industrial applications, preprint, arXiv:2209.02976. https://doi.org/10.48550/arXiv.2209.02976
57. C. Y. Wang, A. Bochkovskiy, H. Y. Mark Liao, YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2023), 7464–7475. https://doi.org/10.1109/CVPR52729.2023.00721

58. G. Jocher, *Yolov8*, Available from: https://github.com/ultralytics/ultralytics, 2023.

59. C. Y. Wang, I. H. Yeh, H. Y. Mark Liao, Yolov9: Learning what you want to learn using programmable gradient information, in *European Conference on Computer Vision(ECCV)*, (2024), 1–21. https://doi.org/10.1007/978-3-031-72751-1_1

60. A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, G. Ding, Yolov10: Real-time end-to-end object detection, *Adv. Neural Inf. Process. Syst.*, **37** (2024), 107984–108011. https://doi.org/10.52202/079017-3429

61. R. Khanam, M. Hussain,Yolov11: An overview of the key architectural enhancements, preprint, arXiv:2410.17725. https://doi.org/10.48550/arXiv.2410.17725

62. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, et al., SSD: Single shot multibox detector, in *European Conference on Computer Vision(ECCV)*, (2016), 21–37. https://doi.org/10.1007/978-3-319-46448-0_2

63. C. Y. Fu, W. Liu, A. Ranga, A. Tyagi, A. C. Berg, DSSD: Deconvolutional single shot detector, preprint, arxiv:1701.06659. https://doi.org/10.48550/arXiv.1701.06659

64. Z. Li, L. Yang, F. Zhou, FSSD: feature fusion single shot multibox detector, preprint arXiv:1712.00960. https://doi.org/10.48550/arXiv.1712.00960

65. J. Jeong, H. Park, N. Kwak, Enhancement of SSD by concatenating feature maps for object detection, preprint, arXiv:1705.09587. https://doi.org/10.48550/arXiv.1705.09587

66. A. Chandio, G. Gui, T. Kumar, I. Ullah, R. Ranjbarzadeh, A. M. Roy, Precise single-stage detector, preprin, arxiv:2210.04252. https://doi.org/10.48550/arXiv.2210.04252

67. B. Huo, C. Li, J. Zhang, Y. Xue, Z. Lin, SAFF-SSD: Self-attention combined feature fusion-based SSD for small object detection in remote sensing. Remote Sensing, *Remote Sens.*, **15** (2023), 3027. https://doi.org/10.3390/rs15123027

68. X. Li, C. Wang, Z. Zeng, WS-SSD: Achieving faster 3D object detection for autonomous driving via weighted point cloud sampling, *Exp. Syst. Appl.*, **249** (2024), 123805. https://doi.org/10.1016/j.eswa.2024.123805

69. Y. Shao, A. Tan, Z. Sun, E. Zheng, T. Yan, P. Liao, PV-SSD: A multi-modal point cloud 3D object detector based on projection features and voxel features, *IEEE Trans. Emerging Top. Comput. Intell.*, **8** (2024), 3436–3449. https://doi.org/10.1109/TETCI.2024.3389710

70. L. Tan, H. Wu, Z. Xu, J. Xia, Multi-object garbage image detection algorithm based on SP-SSD, *Exp. Syst. Appl.*, **263** (2025), 125773. https://doi.org/10.1016/j.eswa.2024.125773

71. H. Law, J. Deng, Cornernet: Detecting objects as paired keypoints, in *2018 Proceedings of the European conference on computer vision (ECCV)*, (2018), 765–781. https://doi.org/10.1007/978-3-030-01264-9_45

72. X. Zhou, D. Wang, P. Krähenbühl, Objects as points, preprint, arXiv:1904.07850. https://doi.org/10.48550/arXiv.1904.07850

73. T. Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in *2017 Proceedings of the IEEE International Conference on Computer Vision(ICCV)*, (2017), 2999–3007. https://doi.org/10.1109/ICCV.2017.324

74. M. Tan, R. Pang, Q. V. Le, Efficientdet: Scalable and efficient object detection, in *2020 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR).*, (2020), 10778–10787. https://doi.org/10.1109/CVPR42600.2020.01079

75. N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in *European Conference on Computer Vision(ECCV)* , (2020), 213–229. https://doi.org/10.1007/978-3-030-58452-8_13

76. X. Zhu, W. Su, L. Lu, B. Li, X. Wang, J. Dai, Deformable detr: Deformable transformers for end-to-end object detection, preprint, arXiv:2010.04159. https://doi.org/10.48550/arXiv.2010.04159

77. D. Meng, X. Chen, Z. Fan, G. Zeng, H. Li, Y. Yuan, et al., Conditional detr for fast training convergence, in *2021 Proceedings of the IEEE International Conference on Computer Vision(ICCV)*, (2021), 3631–3640. https://doi.org/10.1109/ICCV48922.2021.00363

78. S. Liu, F. Li, H. Zhang, X. Yang, X. Qi, H. Su, et al., Dab-detr: Dynamic anchor boxes are better queries for detr, preprint, arXiv:2201.12329. https://doi.org/10.48550/arXiv.2201.12329

79. F. Li, H. Zhang, S. Liu, J. Guo, L. M. Ni, L. Zhang, Dn-detr: Accelerate detr training by introducing query denoising, in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2022), 13609–13617. https://doi.org/10.1109/CVPR52688.2022.01325

80. R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in *2014 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2014), 580–587. https://doi.org/10.1109/CVPR.2014.81

81. R. Girshick, Fast R-CNN, in *2015 Proceedings of the IEEE International Conference on Computer Vision(ICCV)*, (2015), 1440–1448. https://doi.org/10.1109/ICCV.2015.169

82. S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, *IEEE Trans. Pattern Analy. Mach. Intell.*, **39** (2016), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

83. K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in *2017 Proceedings of the IEEE international conference on computer vision(ICCV)*, (2017), 2980–2988. https://doi.org/10.1109/ICCV.2017.322

84. J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, D. Lin, Libra r-cnn: Towards balanced learning for object detection, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2019), 821–830. https://doi.org/10.1109/CVPR.2019.00091

85. X. Lu, B. Li, Y. Yue, Q. Li, J. Yan, Grid R-CNN, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2019), 7355–7364. https://doi.org/10.1109/CVPR.2019.00754

86. Z. Cai, N. Vasconcelos, Cascade r-cnn: Delving into high quality object detection, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2018), 6154–6162. https://doi.org/10.1109/CVPR.2018.00644

87. H. Zhang, H. Chang, B. Ma, N. Wang, X. Chen, Dynamic R-CNN: Towards high quality object detection via dynamic training, in *European conference on computer vision (ECCV)*, (2020), 260–275. https://doi.org/10.1007/978-3-030-58555-6_16

88. G. Gkioxari, J. Malik, J. Johnson, Mesh R-CNN, in *2019 Proceedings of the IEEE international conference on computer vision(ICCV)*, (2019), 9784–9794. https://doi.org/10.1109/ICCV.2019.00988

89. P. Voigtlaender, J. Luiten, P. H. S. Torr, B. Leibe, Siam R-CNN: Visual tracking by re-detection, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2020), 6577–6587. https://doi.org/10.1109/CVPR42600.2020.00661

90. P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, et al., Sparse r-cnn: End-to-end object detection with learnable proposals, in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2021), 14449–14458. https://doi.org/10.1109/CVPR46437.2021.01422

91. Q. Hong, F. Liu, D. Li, J. Liu, L. Tian, Y. Shan, Dynamic sparse R-CNN, in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2022), 4713–4722. https://doi.org/10.1109/CVPR52688.2022.00468

92. Z. Li, F. Wang, N. Wang, Lidar R-CNN: An efficient and universal 3d object detector, in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2021), 7542–7551. 10.1109/CVPR46437.2021.00746

93. P. Song, P. Li, L. Dai, T. Wang, Z. Chen, Boosting R-CNN: Reweighting R-CNN samples by RPN's error for underwater object detection, *Neurocomputing*, **530** (2023), 150–164. https://doi.org/10.1016/j.neucom.2023.01.088

94. J. Mao, M. Niu, H. Bai, X. Liang, H. Xu, C. Xu, Pyramid R-CNN: Towards better performance and adaptability for 3d object detection, in *2021 Proceedings of the IEEE international conference on computer vision(ICCV)*, (2021), 2703–2712. https://doi.org/10.1109/ICCV48922.2021.00272

95. X. Xie, G. Cheng, J. Wang, K. Li, X. Yao, J. Han, Oriented R-CNN and beyond, *Int. J. Comput. Vision*, **132** (2024), 2420–2442. https://doi.org/10.1007/s11263-024-01989-w

96. K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognitionn, *IEEE Trans. Pattern Anal. Mach. Intell.*, **37** (2015), 1904–1916. https://doi.org/10.1109/TPAMI.2015.2389824

97. J. Dai, Y. Li, K. He, J. Sun, R-FCN: Object detection via region-based fully convolutional networks, *Adv. Neural Inf. Process. Syst.*, **2016** (2016), 29.

98. Y. Li, Y. Chen, N. Wang, Z. Zhang, Scale-aware trident networks for object detection, in *2019 Proceedings of the IEEE International Conference on Computer Vision(ICCV)* , (2019), 6053–6062. https://doi.org/10.1109/ICCV.2019.00615

99. S. Zhang, L. Wen, X. Bian, Z. Lei, S. Z. Li, Single-shot refinement neural network for object detection, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2018), 4203–4212. https://doi.org/10.1109/CVPR.2018.00442

100. C. Yang, Y. Cao, X. Lu, Towards better small object detection in UAV scenes: Aggregating more object-oriented information, *Pattern Recognit. Lett.*, **182** (2024), 24–30. https://doi.org/10.1016/j.patrec.2024.04.002

101. X. Wang, N. He, C. Hong, Q. Wang, M. Chen, Improved YOLOX-X based UAV aerial photography object detection algorithm, *Image Vision Comput.*, **135** (2023), 104697. https://doi.org/10.1016/j.imavis.2023.104697

102. Y. Hui, J. Wang, B. Li, DSAA-YOLO: UAV remote sensing small target recognition algorithm for YOLOV7 based on dense residual super-resolution and anchor frame adaptive regression strategy, *J. King Saud Univ. Comput. Inf. Sci.*, **36** (2024), 101863. https://doi.org/10.1016/j.jksuci.2023.101863

103. C. Jiang, H. Ren, X. Ye, J. Zhu, H. Zeng, Y. Nan, et al., Object detection from UAV thermal infrared images and videos using YOLO models, *Int. J. Appl. Earth Obs. Geoinformation*, **112** (2022), 102912. https://doi.org/10.1016/j.jag.2022.102912

104. Y. Li, Y. Wang, Z. Ma, X. Wang, Y. Tang, Sod-uav: Small object detection for unmanned aerial vehicle images via improved yolov7, in *ICASSP 2024–2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* , (2024), 7610–7614. https://doi.org/10.1109/ICASSP48485.2024.10448458

105. Y. Chang, D. Li, Y. Gao, Y. Su, X. Jia, An improved YOLO model for UAV fuzzy small target image detection, *Appl. Sci.*, **13** (2023), 5409. https://doi.org/10.3390/app13095409

106. S. Sambolek, M. Ivasic-Kos, Automatic person detection in search and rescue operations using deep CNN detectors, *IEEE Access*, **9** (2021), 37905–37922. https://doi.org/10.1109/ACCESS.2021.3063681

107. H. Zhou, A. Ma, Y. Niu, Z. Ma, Small-object detection for UAV-based images using a distance metric method, *Drones*, **6** (2022), 308. https://doi.org/10.3390/drones6100308

108. X. Wang, Q. Zhao, P. Jiang, Y. Zheng, L. Yuan, P. Yuan, LDS-YOLO: A lightweight small object detection method for dead trees from shelter forest, *Comput. Electron. Agric.*, **198** (2022), 107035. https://doi.org/10.1016/j.compag.2022.107035

109. X. Luo, Y. Wu, L. Zhao,YOLOD: A target detection method for UAV aerial imagery, *Remote Sens.*, **14** (2022), 3240. https://doi.org/10.3390/rs14143240

110. I. Gallo, A. U. Rehman, R. H. Dehkordi, N. Landro, R. L. Grassa, M. Boschetti, Deep object detection of crop weeds: Performance of YOLOv7 on a real case dataset from UAV images, *Remote Sens.*, **15** (2023), 539. https://doi.org/10.3390/rs15020539

111. J. Cao, W. Bao, H. Shang, M. Yuan, Q. Cheng, GCL-YOLO: A GhostConv-based lightweight yolo network for UAV small object detection, *Remote Sens.*, **15** (2023), 4932. https://doi.org/10.3390/rs15204932

112. M. H. Junos, A. S. M. Khairuddin, S. Thannirmalai, M. Dahari, Automatic detection of oil palm fruits from UAV images using an improved YOLO model, *Visual Comput.*, **38** (2022), 2341–2355. https://doi.org/10.1007/s00371-021-02116-3

113. S. Zeng, W. Yang, Y. Jiao, L. Geng, X. Chen, SCA-YOLO: A new small object detection model for UAV images, *Visual Comput.*, **40** (2024), 1787–1803. https://doi.org/10.1007/s00371-023-02886-y

114. W. Sun, L. Dai, X. Zhang, P. Chang, X. He, RSOD: Real-time small object

detection algorithm in UAV-based traffic monitoring, *Appl. Intell.*, **52** (2022), 8448–8463. https://doi.org/10.1007/s10489-021-02893-3

115. M. Hu, Z. Li, J. Yu, X. Wan, H. Tan, Z. Lin, Efficient-lightweight YOLO: improving small object detection in YOLO for aerial images, *Sensors*, **23** (2023), 6423. https://doi.org/10.3390/s23146423
116. H. Wang, C. Liu, Y. Cai, L. Chen, Y. Li, YOLOv8-QSD: An improved small object detection algorithm for autonomous vehicles based on YOLOv8, *IEEE Trans. Instrum. Meas.*, **73** (2024), 1–16. https://doi.org/10.1109/TIM.2024.3379090
117. Y. Zhang, H. Zhang, Q. Huang, Y. Han, M. Zhao, DsP-YOLO: An anchor-free network with DsPAN for small object detection of multiscale defects, *Exp. Sys. Appl.*, **241** (2024), 122669. https://doi.org/10.1016/j.eswa.2023.122669
118. C. Bai, K. Zhang, H. Jin, P. Qian, R. Zhai, K. Lu, SFFEF-YOLO: Small object detection network based on fine-grained feature extraction and fusion for unmanned aerial images, *Image Vision Comput.*, **156** (2025), 105469. https://doi.org/10.1016/j.imavis.2025.105469
119. K. J. Giri, SO-YOLOv8: A novel deep learning-based approach for small object detection with YOLO beyond COCO, *Exp. Syst. Appl.*, **280** (2025), 127447. https://doi.org/10.1016/j.eswa.2025.127447
120. T. Hou, C. Leng, J. Wang, Z. Pei, J. Peng, I. Cheng, et al., MFEL-YOLO for small object detection in UAV aerial images, *Exp. Syst. Appl.*, **291** (2025), 128459. https://doi.org/10.1016/j.eswa.2025.128459
121. F. Özge-Ünel, B. O. Özkalayci, C. Çiğla, The power of tiling for small object detection, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition workshops (CVPRW)* , (2019), 582–591. https://doi.org/10.1109/CVPRW.2019.00084
122. X. Liang, J. Zhang, L. Zhuo, Y. Li, Q. Tian, Small object detection in unmanned aerial vehicle images using feature fusion and scaling-based single shot detector with spatial context analysis, *IEEE Trans. Circuits Syst. Video Technol.*, **30** (2019), 1758–1770. https://doi.org/10.1109/TCSVT.2019.2905881
123. Y. Li, H. Dong, H. Li, X. Zhang, B. Zhang, Z. Xiao, Multi-block SSD based on small object detection for UAV railway scene surveillance, *Chin. J. Aeronaut.*, **33** (2020), 1747–1755. https://doi.org/10.1016/j.cja.2020.02.024
124. J. Song, Z. Yu, G. Qi, Q. Su, J. Xie, W. Liu, UAV image small object detection based on RSAD algorithm, *Appl. Sci.*, **13** (2023), 11524. https://doi.org/10.3390/app132011524
125. V. N. Nguyen, R. Jenssen, D. Roverso, Intelligent monitoring and inspection of power line components powered by UAVs and deep learning, *IEEE Power Energy Technol. Syst. J.*, **6** (2019), 11–21. https://doi.org/10.1109/JPETS.2018.2881429
126. C. Sun, Y. Ai, S. Wang, W. Zhang, Mask-guided SSD for small-object detection, *Appl. Intell.*, **51** (2021), 3311–3322. https://doi.org/10.1007/s10489-020-01949-0
127. F. Yang, L. Huang, X. Tan, Y. Yuan, FasterNet-SSD: A small object detection method based on SSD model, *Signal Image Video Process.*, **18** (2024), 173–180. https://doi.org/10.1007/s11760-023-02726-5

128. L. Gong, X. Huang, Y. Chao, J. Chen, B. Lei, An enhanced SSD with feature cross-reinforcement for small-object detection, *Appl. Intell.*, **53** (2023), 19449–19465. https://doi.org/10.1007/s10489-023-04544-1

129. J. Xiao, S. Zhang, Y. Dai, Z. Jiang, B. Yi, C. Xu, Multiclass object detection in UAV images based on rotation region network, *IEEE J. Miniaturization Air Space Syst.*, **1** (2020), 188–196. https://doi.org/10.1109/JMASS.2020.3025970

130. J. Butler, H. Leung, A novel keypoint supplemented R-CNN for UAV object detection, *IEEE Sens. J.*, **23** (2023), 30883–30892. https://doi.org/10.1109/JSEN.2023.3330146

131. L. Du, Y. Sun, S. Chen, J. Feng, Y. Zhao, Z. Yan, et al., A novel object detection model based on faster R-CNN for spodoptera frugiperda according to feeding trace of corn leaves, *Agriculture*, **12** (2022), 248. https://doi.org/10.3390/agriculture12020248

132. M. Machefer, F. Lemarchand, V. Bonnefond, A. Hitchins, P. Sidiropoulos, Mask R-CNN refitting strategy for plant counting and sizing in UAV imagery, *Remote Sens.*, **12** (2020), 3015. https://doi.org/10.3390/rs12183015

133. D. Avola, L. Cinque, A. Diko, A. Fagioli, G. L. Foresti, A. Mecca, et al., MS-Faster R-CNN: Multi-stream backbone for improved Faster R-CNN object detection and aerial tracking from UAV images, *Remote Sens.*, **13** (2021), 1670. https://doi.org/10.3390/rs13091670

134. J. Liu, Z. Wang, Y. Wu, Y. Qin, X. Cao, Y. Huang, An improved faster R-CNN for UAV-based catenary support device inspection, *Int. J. Software Eng. Knowl. Eng.*, **30** (2020), 941–959. https://doi.org/10.1142/S0218194020400136

135. Y. Liu, F. Yang, P. Hu, Small-object detection in UAV-captured images via multi-branch parallel feature pyramid networks, *IEEE Access*, **8** (2020), 145740–145750. https://doi.org/10.1109/ACCESS.2020.3014910

136. X. Zhang, E. Izquierdo, K. Chandramouli, Dense and small object detection in UAV vision based on cascade network, in *2019 Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, (2019), 118–126. https://doi.org/10.1109/ICCVW.2019.00020

137. K. Kamirul, O. A. Pappas, A. M. Achim, R-sparse R-CNN: SAR ship detection based on background-aware sparse learnable proposals, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, **18** (2025), 14955–14973. https://doi.org/10.1109/JSTARS.2025.3577766

138. X. Yang, A. S. A. Mohamed, Gaussian-based R-CNN with large selective kernel for rotated object detection in remote sensing images, *Neurocomputing*, **620** (2025),129248. https://doi.org/10.1016/j.neucom.2024.129248

139. Z. He, Y. He, AS-Faster-RCNN: An Improved Object Detection Algorithm for Airport Scene Based on Faster R-CNN, *IEEE Access*, **13** (2025), 36050–36064. https://doi.org/10.1109/ACCESS.2025.3539930

140. P. Mittal, A. Sharma, R. Singh, V. Dhull, Dilated convolution based RCNN using feature fusion for Low-Altitude aerial objects, *Exp. Syst. Appl.*, **199** (2022), 117106. https://doi.org/10.1016/j.eswa.2022.117106

141. F. Yang, H. Fan, P. Chu, E. Blasch, H. Ling, Clustered object detection in aerial images, in *2019 Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, (2019), 8310–8319. https://doi.org/10.1109/ICCV.2019.00840

142. C. Li, T. Yang, S. Zhu, C. Chen, S. Guan, Density map guided object detection in aerial images, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition workshops (CVPRW)*, (2020), 737–746. https://doi.org/10.1109/CVPRW50498.2020.00103

143. W. Yu, T. Yang, C. Chen, Towards resolving the challenge of long-tail distribution in UAV images for object detection, in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision(WACV)*, (2021), 3257–3266. https://doi.org/10.1109/WACV48630.2021.00330

144. Z. Ge, S. Liu, F. Wang, Z. Li, J. Sun, Yolox: Exceeding yolo series in 2021, preprint, arxiv:2107.08430. https://doi.org/10.48550/arXiv.2107.08430

145. S. Xu, X. Wang, W. Lv, Q. Chang, C. Cui, K. Deng, et al., PP-YOLOE: An evolved version of YOLO, preprint, arxiv:2203.16250. https://doi.org/10.48550/arXiv.2203.16250

146. Y. Chen, X. Yuan, J. Wang, R. Wu, X. Li, Q, Hou, et al., YOLO-MS: Rethinking multi-scale representation learning for real-time object detection, *IEEE Trans. Pattern Anal. Mach. Intell.*, **47** (2025), 4240–4252. https://doi.org/10.1109/TPAMI.2025.3538473

147. Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, et al., Detrs beat yolos on real-time object detection, in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2024), 16965–16974. https://doi.org/10.1109/CVPR52733.2024.01605

148. C. Xue, Y. Xia, M. Wu, Z. Chen, F. Cheng, L. Yun, EL-YOLO: An efficient and lightweight low-altitude aerial objects detector for onboard applications, *Exp. Syst. Appl.*, **256** (2024), 124848. https://doi.org/10.1016/j.eswa.2024.124848

149. H. Luo, Y. Wang, Y. Chen, X. Li, J. Zhan, D. Zuo, Ebc-yolo: A remote sensing target recognition model adapted for complex environments, *Earth Sci. Inf.*, **18** (2025), 282. https://doi.org/10.1007/s12145-025-01808-x

150. S. Liu, J. Zha, J. Sun, Z. Li, G. Wang, EdgeYOLO: An edge-real-time object detector, in *2023 42nd Chinese Control Conference (CCC)*, (2023), 7507–7512. https://doi.org/10.23919/CCC58697.2023.10239786

151. C. Wang, Y. Han, C. Yang, M. Wu, Z. Chen, L. Yun, et al., CF-YOLO for small target detection in drone imagery based on YOLOv11 algorithm, *Sci. Rep.*, **15** (2025), 16741. https://doi.org/10.1038/s41598-025-99634-0

152. Y. Wu, X. Mu, H. Shi, M. Hou, An object detection model AAPW-YOLO for UAV remote sensing images based on adaptive convolution and reconstructed feature fusion, *Sci. Rep.*, **15** (2025), 16214. https://doi.org/10.1038/s41598-025-00239-4

153. J. Zhang, J. Huang, X. Chen, D. Zhang, How to fully exploit the abilities of aerial image detectors, in *2019 Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, (2019), 1–8. https://doi.org/10.1109/ICCVW.2019.00007

154. Z. Wei, C. Duan, X. Song, Y. Tian, H. Wang, Amrnet: Chips augmentation in aerial images object detection, preprint, arxiv:2009.07168. https://doi.org/10.48550/arXiv.2009.07168

155. C. Duan, Z. Wei, C. Zhang, S. Qu, H. Wang, Coarse-grained density map guided object detection in aerial images, in *2021 Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, (2021), 2789–2798. https://doi.org/10.1109/ICCVW54120.2021.00313

156. S. Deng, S. Li, K. Xie, W. Song, X. Liao, A. Hao, et al., A global-local self-adaptive network for drone-view object detection, *IEEE Trans. Image Process.*, **30** (2020), 1556–1569. https://doi.org/10.1109/TIP.2020.3045636

157. Z. Liu, G. Gao, L. Sun, Z. Fang, HRDNet: High-resolution detection network for small objects, preprint, arxiv:2006.07607. https://doi.org/10.48550/arXiv.2006.07607

158. F. C. Akyon, S. O. Altinuc, A. Temizel, Slicing aided hyper inference and fine-tuning for small object detection, in *2022 IEEE International Conference on Image Processing (ICIP)*, (2022), 966–970. https://doi.org/10.1109/ICIP46576.2022.9897990

159. C. Yang, Z. Huang, N. Wang, QueryDet: Cascaded sparse query for accelerating high-resolution small object detection, in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2022), 13658–13667. https://doi.org/1010.1109/CVPR52688.2022.01330

160. B. Du, Y. Huang, J. Chen, D. Huang, Adaptive sparse convolutional networks with global context enhancement for faster object detection on drone images, in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2023), 13435–13444. https://doi.org/10.1109/CVPR52729.2023.01291

161. C. Liu, G. Gao, Z. Huang, Z. Hu, Q. Liu, Y. Wang, Yolc: You only look clusters for tiny object detection in aerial images, *IEEE Trans. Intell. Transp. Syst.*, **25** (2024), 13863–13875. https://doi.org/10.1109/TITS.2024.3386928

AIMS Press