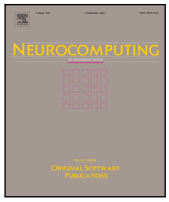
Neurocomputing 598 (2024) 128102

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Development and challenges of object detection: A survey

Zonghui Li [a], Yongsheng Dong [a,*], Longchao Shen [a], Yafeng Liu [a], Yuanhua Pei [a], Haotian Yang [a], Lintao Zheng [a], Jinwen Ma [b]

[a] School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China
[b] Department of Information and Computational Sciences, School of Mathematical Sciences and LMAM, Peking University, Beijing 100871, China

ARTICLE INFO

Keywords:
Object detection
Deep learning
Datasets
Evaluation metrics
Difficulties and challenges

ABSTRACT

Object detection is a basic vision task that accompanies people's daily lives all the time. The development of object detection technology has experienced an evolution from traditional-based algorithms to deep learning-based algorithms, which has made a qualitative leap in both detection accuracy and detection speed. With the advancement of deep learning, object detection techniques are increasingly becoming a part of everyday life, with the YOLO series of algorithms being extensively applied in various industries. In this paper, we initially present the frequently utilized datasets and evaluation criteria for object detection. Subsequently, we delve into the evolution of traditional object detection algorithms, highlighting two-stage and one-stage approaches through illustrative examples of classical methods. We also conduct a comprehensive summary and analysis of the detection results obtained by these methods. In addition, we introduce object detection applications in daily life, as well as the importance and some difficulties of these applications. Finally, we analyze and summarize the difficulties and challenges facing the task of object detection, and we look forward to the future development direction of object detection.

1. Introduction

The role of computer vision [1,2] is to give computers the ability to visualize like humans so that they can intuitively understand the content in an image or video like humans, thus meeting the needs of modern society. Object detection [3–5], as a core task in the field of computer vision, aims to explore possible objects in images and accurately identify their specific orientations, as shown in Fig. 1. In today's society, artificial intelligence [6–10] has come into our lives and brought us a lot of convenience. Similarly, object detection technology is also important in modern society. For example, object detection technology has made autonomous driving [11–16] possible, allowing cars to autonomously recognize objects on complex roads and safely avoid obstacles while traveling on the road. Object detection technology also has deep attainments in the medical [17–21] field, which can help doctors initially screen lesion images and improve the efficiency and accuracy of their diagnosis. In recent years, the flourishing development of deep learning [22–25] technology has greatly promoted the improvement and enhancement of object detection technology. Not only has significant progress been made in accuracy, but also in processing speed, fully meeting the diverse needs of daily life and technological applications. However, for some tasks that require high real-time accuracy, the object detection technology still needs to be further improved, which is also the direction of the future development of object detection technology.

In the past, when the development of deep learning techniques was not very mature, many traditional object detection techniques used hand-constructed features to capture the information in the input image more efficiently. Traditional object detection techniques cover methods such as histogram of oriented gradients (HOG [26]) and deformable part-based model (DPM [27]). These methods have shown impressive detection results in their respective periods. However, due to their reliance on manually constructing features, these techniques are relatively time-consuming and efficiency is limited when extracting image information. Generally, the hand-built features are used to segment the input image by sliding window [28,29] technique, and then image features are extracted, which is a time-consuming process compared to the subsequent convolutional neural network.

In 2012, the world witnessed the rise of convolutional neural networks [30]. Given that convolutional neural networks can easily capture high-level semantic information about an image, most object detection models now use convolutional neural networks as the backbone. Convolutional neural network-based object detection models can be categorized into two main groups: two-stage object detection and one-stage object detection. Typical algorithms for two-stage object detection

* Corresponding author.
E-mail address: ysdong@haust.edu.cn (Y. Dong).

https://doi.org/10.1016/j.neucom.2024.128102
Received 14 March 2024; Received in revised form 14 May 2024; Accepted 15 June 2024
Available online 22 June 2024

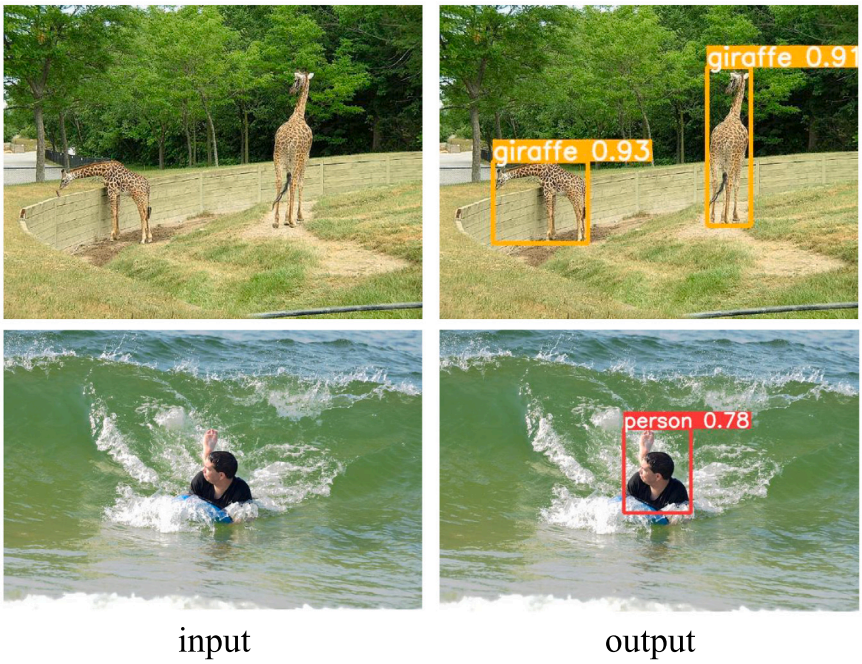


Fig. 1. Input and output images of object detection networks.

are the R-CNN [31–39] series, taking Faster R-CNN [33] as an example, which usually uses region proposal network (RPN) to screen the candidate box first and then classifies and localizes the objects in the candidate box. Since it is divided into a two-step strategy, the detection speed is usually slow, but the detection accuracy is relatively high. To enhance the speed of object detection, one-stage detection algorithms have emerged. Among the prominent examples of such one-stage techniques are the SSD [40] and YOLO [41–49] series.

Although the early one-stage detection models gained a great improvement in detection speed, the detection accuracy was not as good as the two-stage detection. Subsequently, researchers have focused their efforts on the trade-off between detection accuracy and speed, aiming to create a fast and accurate detection model. Based on the superior characteristics of convolutional neural networks, the current one-stage detection model has realized the trade-off between speed and accuracy, surpassing the two-stage detection model and becoming the current mainstream detection framework. This survey is an extended version of our previous conference paper about object detection [50].

2. Object detection dataset and evaluation metrics

The commonly used datasets for object detection tasks are PASCAL Visual Object Classes (VOC [51,52]), Common Objects in Context (COCO [53]), ImageNet Large Scale Visual Recognition Challenge (ILSVRC [54]), Open Images Detection (OID [55]), etc., and they have been a great help to the development of object detection techniques. A good dataset allows the object detection model to perform more fully. Using these high-quality datasets, researchers can evaluate the object detection models and test the robustness of the models in various complex scenarios, etc., thus advancing the development of object detection technology. In addition, high-quality datasets allow the detection model to learn richer image features, further improving the model's detection accuracy and generalization ability. In the following content, we will introduce these datasets in detail. Table 1 is a statistical table of common datasets for object detection, and Fig. 2 is a partial image of common datasets for object detection.

2.1. VOC

The PASCAL Visual Object Classes (VOC[1] [51,52]) dataset is a very classic dataset that can be used for a variety of different computer vision tasks, including object detection, semantic segmentation, image categorization, and so on. So far, the two commonly used versions of the VOC dataset are VOC2007 and VOC2012, and in order to expand the data volume of the dataset, researchers have merged these two versions to form some new datasets. VOC07+12 [33] uses the train and validation sets of VOC2007 and VOC2012 as the new train sets, and the test set of VOC2007 as the validation set. VOC07++12 [33] takes the train set, validation set, and test set of VOC2007, the train set and validation set of VOC2012 as the new train set, and the test set of VOC2007 as the validation set. The PASCAL VOC dataset contains a total of 20 common object categories, such as people, bicycles, dogs, and airplanes. For the train set images, each image has its own corresponding bounding box labeling information, such as the object category, the distance and size of the object, etc. However, the VOC dataset is released earlier, so some of its images may not reflect the current reality of the world well, and may lack some complex occlusion scenes.

2.2. COCO

The Common Objects in Context (COCO[2] [53]) dataset is the most widely used large-scale dataset for object detection technology, and it is also used for semantic segmentation and image classification tasks. The COCO dataset consists of two versions, COCO2014 and COCO2017, which contain a total of 80 common object categories for object detection tasks, covering a variety of complex scenarios in life. The COCO dataset is widely used and has higher image annotation information than other datasets, including object bounding box coordinates, strength segmentation annotation information, key point information, etc. This annotation information can provide more accurate object locations for the object detection model and allow the detection model

[1] http://host.robots.ox.ac.uk/pascal/VOC/
[2] https://cocodataset.org/

Table 1
The statistical table of common datasets for object detection tasks. The "/" left is the number of images, right is the number of objects, "–" indicates that the data is temporarily unknown.

Dataset	Train	Validation	Trainval	Test
VOC2007 [51]	2,501/6,301	2,510/6,307	5,011/12,608	4,952/14,976
VOC2012 [52]	5,717/13,609	5,823/13,841	11,540/27,450	10,991/–
ILSVRC2014 [54]	456,567/478,807	20,121/55,502	476,688/534,309	40,152/–
ILSVRC2017 [54]	456,567/478,807	20,121/55,502	476,688/534,309	65,500/–
COCO2014 [53]	82,783/604,907	40,504/291,875	123,287/896,782	81,434/–
COCO2017 [53]	118,287/860,001	5,000/36,781	123,287/896,782	40,670/–
OID2018 [55]	1,743,042/14,610,229	41,620/204,621	1,784,662/14,814,850	125,436/625,282

Fig. 2. Some images of common datasets for object detection tasks.

to learn richer image information so as to more accurately recognize the image species and object locations. However, the image resolution of the COCO dataset is relatively low, and there may be an imbalance in the number of different object categories, which may affect the training and evaluation of the model.

2.3. ILSVRC

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC[3] [54]) dataset is a large image classification dataset. It has been used in the competition every year from 2010 to 2017, with the aim of better advancing computer vision technology. The commonly used versions of the dataset are ILSVRC-2014 and ILSVRC-2017. Both versions have the same train set and validation set images. The train set data is about 456K images, and the validation set data is about 20K images. The number of images in the test sets of ILSVRC-2014 and ILSVRC-2017 is 40K and 65K, respectively. There are more than 20,000 image categories, and there are a large number of images in each category. Also, this dataset can be used for object detection tasks, where 200 object categories can be used. The images in this dataset cover various scenes, including natural scenes from life, web images, etc., which makes the dataset images more diverse. For each image in the dataset, a primary category label is provided, and several other possible secondary category labels are labeled. However, the ILSVRC dataset may still have insufficient category coverage for some fine-grained object recognition tasks. Meanwhile, the image quality of the dataset may be inconsistent, thus affecting the performance of the model.

2.4. OID

The Open Images Detection (OID[4] [55]) is a large visual dataset created by the Google team in 2018. That is mainly used for object detection tasks and visual relationship detection. For the object detection task, the dataset contains 600 common object categories, including a variety of objects, animals, scenes in daily life, etc. The OID-2018 dataset has a train set of about 1,743K images, a validation set of 40K images, and a test set of 125K images. Each image in the dataset is rigorously labeled with information, including the category of the object in the picture, the location and size of the object bounding box, etc. With a large number of data images, OID-2018 plays a key role in the development of the object detection task and, at the same time, promotes the development of computer vision tasks. However, the OID dataset may lack some unconventional scene images, and some newly emerged object categories may not be found in the dataset, thereby affecting the model's generalization ability.

2.5. Evaluation indicators

Evaluation metrics for object detection are a set of measurements used to assess the effectiveness of object detection models. These metrics encompass precision (P), recall (R), average precision (AP), mean average precision (mAP), and frames per second (FPS), etc.

Prior to discussing these evaluation metrics, it is essential to define four key terms: true positive (TP), false positive (FP), false negative (FN), and true negative (TN). In object detection, predicting objects within an image leads to four possible outcomes: TP occurs when both the predicted and actual boxes are positive; FP arises when the predicted box is positive but the actual box is negative; FN happens when the predicted box is negative while the actual box is positive; TN occurs when both the predicted and actual boxes are negative. Understanding these terms allows us to calculate precision and recall. Precision, sometimes referred to as accuracy, represents the proportion of correct predictions among all detected objects. Recall measures the proportion of accurately detected objects out of all actual objects. These concepts are quantified using the following equations:

$$P_{(precision)} = \frac{TP}{TP + FP} \tag{1}$$

$$R_{(recall)} = \frac{TP}{TP + FN} \tag{2}$$

[3] http://image-net.org/challenges/LSVRC/
[4] https://storage.googleapis.com/openimages/web/index.html

The average precision (AP) represents the mean accuracy for a single category, typically calculated as the area under the precision–recall (PR) curve. This curve plots recall (R) on the horizontal axis against precision (P) on the vertical axis, according to the following formula:

$$AP = \int_0^1 P(R) dR \tag{3}$$

The mean average precision (mAP) determines the average precision (AP) for each individual category by analyzing the accuracy–recall curve. Subsequently, it computes the overall average of these AP values across all categories, serving as the ultimate metric for evaluation. In the context of the COCO dataset, AP is mAP, where AP has been calculated as the mean value under all categories.

The frames per second (FPS) is an important parameter. It is often used to measure the detection speed of a object detection task. Assuming that the preprocessing time is denoted by P, the inference time by I, and the Non-Maximum Suppression (NMS) time by N, the FPS is expressed by the following equation:

$$FPS = \frac{1000}{P + I + N} \tag{4}$$

The intersection over union (IoU) serves as a fundamental metric in object detection for quantifying the overlap extent between predicted outcomes by the model and actual labels. In object detection tasks, we adopt a supervised training approach, where the dataset typically contains manually annotated labels to accurately indicate the true position of objects in the image, which are referred to as ground truth (GT). Upon the input image being processed by the object detection network, it outputs the anticipated location of the object within the image, referred to as the prediction box. The greater the overlap between this prediction box and the ground truth, the more effective the performance of the detection network is considered. But nowadays, the object detection network often cannot detect the position of the object perfectly, so the IoU usually sets a threshold of 0.5, and if the IoU threshold is greater than 0.5, the detection is considered successful. However, in the COCO dataset, in order to pursue higher accuracy, indicators such as AP, AP_{50}, and AP_{75} are set. AP stands for the average accuracy of the IoU threshold between 0.5 and 0.95; AP_{50} stands for the indicators with an IoU threshold greater than 0.5; and AP_{75} stands for the indicators with an IoU threshold greater than 0.75. Fig. 3 shows the prediction box and ground truth of the object in object detection, and the IoU is calculated as follows:

$$IoU = \frac{Prediction\ Box \cap Ground\ Truth}{Prediction\ Box \cup Ground\ Truth} \tag{5}$$

3. Object detection methods

Object detection tasks are generally categorized into traditional methods and deep learning methods. Traditional object detection methods generally require researchers to use hand-designed feature extractors, which are difficult to adapt to the scale and diversity of different objects and are inefficient. The rapid development of deep learning has brought a qualitative leap to the task of object detection and also laid the foundation for technological progress in modern society. Deep learning-based object detection methods can be summarized into two main categories: two-stage object detection algorithms and one-stage object detection algorithms. The two-stage object detection algorithm achieves detection tasks through two independent networks. It performs preliminary screening on candidate boxes, resulting in higher detection accuracy. However, the one-stage object detection algorithm uses only one network to complete detection tasks, without the need for additional segmentation or region proposal steps, so the detection speed is faster. Fig. 4 shows partial methods in the development process of object detection.

3.1. Traditional object detection algorithms

Viola Jones: The Viola–Jones [56] detector is a machine learning algorithm for face recognition and human detection. It was first proposed in 2001. The algorithm utilizes Haar features [57], integral images [57], and the Adaboost algorithm [58] to construct an efficient cascade classifier for the fast detection of objects, especially faces, in images. The Viola–Jones detector has been widely used in the fields of face recognition [59–62] and real-time video processing [63–66] and lays the foundation for the development of subsequent object detection algorithms.

Although the Viola–Jones detector has made significant progress in real-time object detection, it has some limitations. First, the Viola–Jones algorithm is less robust [56] to situations such as illumination changes, attitude changes, and occlusions, which may lead to a decrease in detection accuracy. Second, the algorithm performs poorly when dealing with complex backgrounds and non-frontal [67] objects, which can easily lead to false detections. Finally, although the Viola–Jones algorithm is able to detect objects quickly, it may have certain computational resource requirements when dealing with large-scale datasets and is not applicable to certain resource-constrained environments.

Histogram of Oriented Grades: The histogram of oriented gradients (HOG [26]) detector is a classical object detection algorithm originally proposed in 2005. The algorithm describes the structural and textural features of an image by calculating histograms of gradient orientations in local regions of the image and uses these features to detect objects. The HOG detector is primarily applied in visual tasks like pedestrian detection, excelling in capturing the object's edge and texture features through gradient information extraction from the image. This capability facilitates efficient object recognition and localization. Although the HOG detector performs well in some scenarios, it is less robust to factors such as illumination changes and pose changes and also requires higher computational resources for feature extraction and object detection.

Deformable Parts Model: The deformable parts model (DPM [27]) detector is a classical object detection algorithm proposed in 2010. The algorithm is based on the local parts model of an image and achieves object detection by learning the morphology and compositional structure of the object. The DPM detector takes into account the deformations of the object and the interrelationships between different parts in the detection process and has strong robustness and accuracy. It performs well in object detection tasks such as human bodies and automobiles and achieves better performance in object localization and detection accuracy. However, the DPM detector may suffer from performance degradation when dealing with situations such as complex backgrounds and occlusions, and it requires high computational resources.

3.2. Two-stage object detection algorithms

R-CNN: The region with convolutional neural network (R-CNN [31]) was proposed in 2014. This algorithm is the first time that deep learning has been used for an object detection task and has achieved excellent results. R-CNN abandons the traditional manual feature extraction method and uses a selective search [68] algorithm to divide the candidate regions on the input image, and then each region is classified and regressed by a convolutional neural network. In paper [31], R-CNN achieved admirable results on the VOC2007 test set, with the mAP reaching 58.5% when using the T-Net [69] structure and 66.0% when using the O-Net [70] structure. Compared to the previous more popular detector, DPM, there is a high improvement, with mAP increasing from 34.3% in DPM HSC [71] to 58.5% in R-CNN. The experimental results clearly demonstrate that the R-CNN detector was considered cutting-edge at its time, utilizing a two-stage method to segment the detection task into distinct phases, significantly enhancing the accuracy

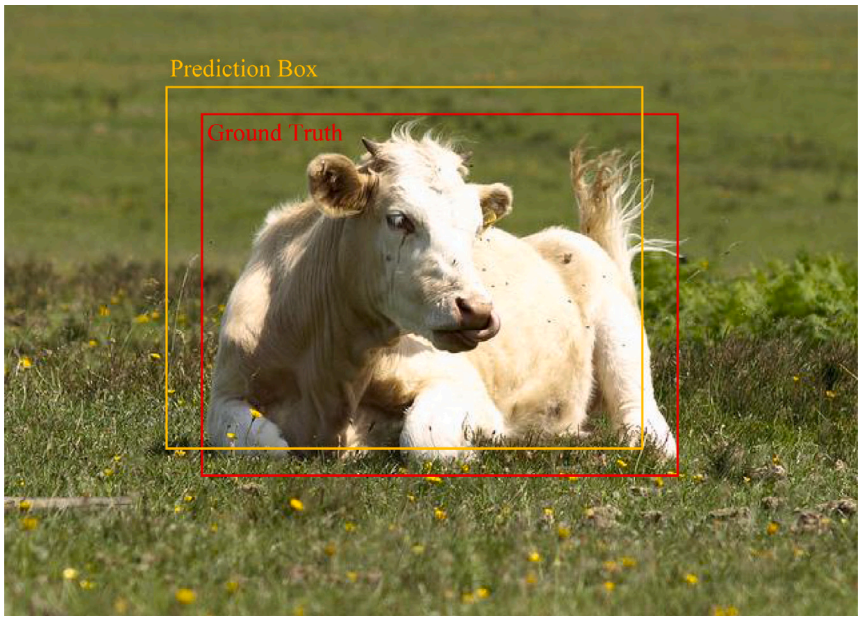


Fig. 3. Ground truth and prediction box of object in image.

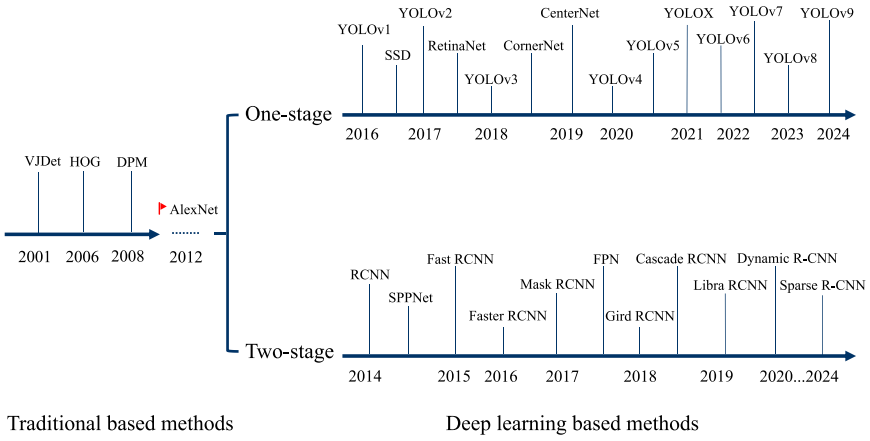


Fig. 4. Partial methods in the development process of object detection.

of object detection. Nonetheless, the inference speed of R-CNN is noted for being sluggish, failing to fulfill real-time detection demands. And the training process is divided into stages, which require independent training of convolutional neural networks and support vector machines (SVM [72–74]), which increases the complexity of training.

Fast R-CNN: The fast region-based convolutional network (Fast R-CNN [32]) was proposed in 2015 to alleviate the problem of the computational inefficiency of R-CNN. Fast R-CNN draws on the strengths of spatial pyramid pooling network (SPPNet [75]) to share convolutional parameters, alleviating the disadvantage of convolutional neural networks for each candidate region in R-CNN and integrating classification and regression tasks into one network. In paper [32], on the VOC2007 test set, Fast R-CNN outperforms R-CNN (58.5%) in detection accuracy, with a mAP of 66.9%. On the VOC2012 dataset, similarly, there is an improvement compared to R-CNN, with mAP improving from 62.4% for R-CNN to 65.7% for Fast R-CNN. When trained on the VOC07+12 dataset, the mAP of Fast R-CNN reaches 70.0%, which is a great improvement compared to R-CNN. In summary, Fast R-CNN achieved advanced detection results on the VOC dataset, stood out among the detection models at that time, and mitigated the shortcomings of the time-consuming training of the R-CNN detection algorithm, becoming a relatively mainstream detection model at that time. However, Fast R-CNN does not discard the selective search algorithm and still needs to spend some computational cost to extract candidate regions, which is not enough for real-time target detection.

Faster R-CNN: In 2015, Ren et al. introduced Faster R-CNN [33], moving away from the use of the selective search algorithm for generating candidate regions. Instead, Faster R-CNN introduces a Region Proposal Network (RPN) that streamlines the training process and lowers the complexity of training. RPN efficiently selects candidate regions with high scores and fine-tunes them through regression, narrowing down the search space for accurate object detection. Faster R-CNN is an improved version of the Fast R-CNN algorithm and has achieved some results. In paper [33], on the VOC2007 test set, different training sets have different detection results. When trained on the VOC2007 dataset, the mAP is 69.9%. When trained on the VOC07+12 dataset, the mAP is 73.2%. When tested on the COCO test-dev dataset, when the IoU threshold is 0.5, the mAP is 42.7%. When the IoU threshold is between 0.5 and 0.95, the mapping is 21.9%. Faster R-CNN, building upon the foundations of R-CNN and Fast R-CNN, significantly enhances detection performance. However, faster R-CNN still suffers from the disadvantages of long training times and a relatively complex computational process, which may pose certain challenges for real-time application scenarios.

Table 2
The detection results of common two-stage object detection methods on different training and testing sets.

Method	Author	Train set	Test set	mAP
R-CNN [31]	Ross Girshick et al.	VOC2010	VOC2010 test	50.2
R-CNN BB [31]	Ross Girshick et al.	VOC2010	VOC2010 test	53.7
R-CNN T-Net [31]	Ross Girshick et al.	VOC2007	VOC2007 test	54.2
R-CNN T-Net BB [31]	Ross Girshick et al.	VOC2007	VOC2007 test	58.5
R-CNN O-Net [31]	Ross Girshick et al.	VOC2007	VOC2007 test	62.2
R-CNN O-Net BB [31]	Ross Girshick et al.	VOC2007	VOC2007 test	66.0
Fast R-CNN [32]	Ross Girshick et al.	VOC2007	VOC2007 test	66.9
Fast R-CNN [32]	Ross Girshick et al.	VOC07+12	VOC2007 test	70.0
Fast R-CNN [32]	Ross Girshick et al.	VOC2012	VOC2010 test	66.1
Fast R-CNN [32]	Ross Girshick et al.	VOC07++12	VOC2010 test	68.8
Fast R-CNN [32]	Ross Girshick et al.	VOC2012	VOC2012 test	65.7
Fast R-CNN [32]	Ross Girshick et al.	VOC07++12	VOC2012 test	68.4
Faster R-CNN [33]	Kaiming He et al.	VOC2007	VOC2007 test	69.9
Faster R-CNN [33]	Kaiming He et al.	VOC07+12	VOC2007 test	73.2
Faster R-CNN [33]	Kaiming He et al.	VOC2007	VOC2012 test	67.0
Faster R-CNN [33]	Kaiming He et al.	VOC07++12	VOC2012 test	70.4
Faster R-CNN [33]	Kaiming He et al.	COCO	VOC2007 test	76.1
Faster R-CNN [33]	Kaiming He et al.	COCO	VOC2012 test	73.0
Faster R-CNN [33]	Kaiming He et al.	COCO+VOC07+12	VOC2007 test	78.8
Faster R-CNN [33]	Kaiming He et al.	COCO+VOC07++12	VOC2012 test	75.9

Table 2 shows the detection results of common two-stage object detection methods on different train and test sets, where VOC07+12 denotes the train sets and validation sets of the VOC2007 and VOC2012 datasets, and VOC07++12 denotes the train, validation, and test sets of VOC2007 and VOC2012. BB denotes the bounding-box regression stage that reduces localization errors. As can be seen from Table 2, the R-CNN O-Net BB method mAP reaches up to 66.0%. The Fast R-CNN technique achieves a mAP of up to 70.0%, marking a 4.0% increase over R-CNN (66.0%). Meanwhile, Faster R-CNN's mAP peaks at 78.8%, showcasing an 8.8% enhancement relative to Fast R-CNN (70.0%). From these data, it can be seen that the R-CNN series gradually improves in terms of accuracy, which is enough to reflect the superiority of the two-stage object detection methods. However, the two-stage method still needs to be improved in terms of speed, which is not enough for some tasks that require real-time performance.

3.3. One-stage object detection algorithms

YOLO: The you only look once (YOLO [41]) is a popular real-time object detection algorithm proposed in 2015. The algorithm achieves fast and accurate object detection by transforming the object detection task into a regression problem for a single neural network and making predictions directly over the entire image. YOLO employs a convolutional network architecture to divide the input image into S × S grids. In every grid, it predicts B bounding boxes and estimates the probability that each box corresponds to K different categories. The main advantage of the YOLO detector is its fast detection speed. In paper [41], the FPS of Fast YOLO reaches an amazing 155, and the FPS of YOLO is 45. Though the YOLO series has a very fast detection speed, there are some drawbacks as well. The YOLO algorithm is a one-stage network and does not have a candidate region filtering phase like the two-stage network, so the localization accuracy is relatively low, and it is easy to have inaccurate localization for small objects. In addition, the YOLO algorithm is not effective in detecting difficult samples and occlusions and is easy to missed detection.

SSD: The single shot multiBox detector (SSD [40]) was proposed in 2016, realizing an end-to-end single model to accomplish the object detection task and simplifying the entire detection process. SSD proposes the concept of a default box, which eliminates proposal generation and the subsequent resampling phase, simplifying the computation of the network. In addition, SSD adopts for the first time a multilayer feature map for the detection task, which is performed on three different sizes of feature maps. Different sizes of default boxes are represented on different sizes of feature maps, and the detection results of multiple different sizes of feature maps are combined to achieve the detection of objects of different sizes. In paper [40], the authors validated it on multiple datasets. SSD can be categorized into SSD300 and SSD512 based on the input image size. On the PASCAL VOC07+12 dataset, the map of SSD512 is 81.6%. On the PASCAL VOC07++12 dataset, the map of SSD512 is 74.9%, which is 17% higher compared to YOLO (57.9%). Overall, the SSD algorithm demonstrates a significant enhancement in detection accuracy over YOLO, with the added benefit of being able to detect objects across multiple scales. However, it lacks speed compared to YOLO.

YOLOv2: YOLOv2 [42] is an improved version of YOLO, proposed in 2016, which has improved both in accuracy and speed. YOLOv2 introduces the concept of anchor boxes based on YOLO, which presets some anchor boxes of possible objects to determine the position of the object, alleviating the disadvantage of inaccurate localization in the YOLO algorithm. In addition, YOLOv2 replaces its backbone network with Darknet-19, which is more accurate and efficient than YOLO, further improving the detection performance and speed of the model. YOLOv2 uses batch normalization (BN [76]) to standardize the feature maps, which speeds up the training speed of the model and its generalization ability. To enhance the detection of objects of varying sizes, YOLOv2 employs multi-scale prediction, allowing it to detect objects of different dimensions on feature maps of different scales. This approach significantly improves the model's capability in detecting small objects. In paper [42], when detecting on the VOC2007 dataset, the mAP of YOLOv2 can reach 78.6%, which is a 15.2% improvement compared to YOLOv1 (63.4%). When testing on the VOC07++12 dataset, YOLOv2's mAP can reach 73.4%, which is a 15.5% improvement compared to YOLO (57.9%). Overall, YOLOv2 has improved a lot in detection accuracy compared to YOLO, and the speed has also improved. However, in terms of small object detection, YOLOv2's detection is still not good enough and needs to be further improved.

Table 3 shows the test results of the YOLO [41], YOLOv2 [42], and SSD [40] models on the VOC dataset. It can be seen from Table 3 that both methods, YOLO and SSD, meet the criteria for real-time detection. When trained on the VOC07+12 dataset and tested on the VOC07 dataset, Fast YOLO achieves an amazing FPS of 155, while SSD300 can achieve an FPS of 46. However, we can also see that the two methods have different test results on different datasets. When the number of train datasets is large enough, the network model can show better detection results on the same test set. In addition, from the table, we can see that when the input image size of the network model is larger, the detection accuracy is also higher. SSD512 has an mAP of 81.6% when trained on the VOC07+12+COCO dataset. Under the same condition, the mAP of SSD300 is 79.6%. YOLOv2 also has a gradual increase in mAP with the increase in the size of the input image, from

Table 3
The test results of YOLOv1 and SSD models on the VOC dataset. "–" indicates that the data is temporarily unknown.

Method	Author	Train set	Test set	mAP	FPS
YOLO [41]	Joseph Redmon et al.	VOC07+12	VOC2007 test	63.4	45
Fast YOLO [41]	Joseph Redmon et al.	VOC07+12	VOC2007 test	52.7	155
YOLO VGG-16 [41]	Joseph Redmon et al.	VOC07+12	VOC2007 test	66.4	21
YOLO [41]	Joseph Redmon et al.	VOC07++12	VOC2012 test	57.9	–
SSD300 [40]	Wei Liu et al.	VOC2007	VOC2007 test	68.0	–
SSD300 [40]	Wei Liu et al.	VOC07+12	VOC2007 test	74.3	46
SSD300 [40]	Wei Liu et al.	VOC07+12+COCO	VOC2007 test	79.6	–
SSD512 [40]	Wei Liu et al.	VOC2007	VOC2007 test	71.6	–
SSD512 [40]	Wei Liu et al.	VOC07+12	VOC2007 test	76.8	19
SSD512 [40]	Wei Liu et al.	VOC07+12+COCO	VOC2007 test	81.6	–
SSD300 [40]	Wei Liu et al.	VOC07++12	VOC2012 test	72.4	–
SSD300 [40]	Wei Liu et al.	VOC07++12+COCO	VOC2012 test	77.5	–
SSD512 [40]	Wei Liu et al.	VOC07++12	VOC2012 test	74.9	–
SSD512 [40]	Wei Liu et al.	VOC07++12+COCO	VOC2012 test	80.0	–
YOLOv2 288*288 [42]	Joseph Redmon et al.	VOC07+12	VOC2007 test	69.0	91
YOLOv2 352*352 [42]	Joseph Redmon et al.	VOC07+12	VOC2007 test	73.7	81
YOLOv2 416*416 [42]	Joseph Redmon et al.	VOC07+12	VOC2007 test	76.8	67
YOLOv2 480*480 [42]	Joseph Redmon et al.	VOC07+12	VOC2007 test	77.8	59
YOLOv2 544*544 [42]	Joseph Redmon et al.	VOC07+12	VOC2007 test	78.6	40
YOLOv2 544*544 [42]	Joseph Redmon et al.	VOC07++12	VOC2012 test	73.4	–

69.1% to 78.6%. However, an increase in the input image size also puts a certain computational burden on the network, resulting in a decrease in FPS. When the input image of YOLOv2 is 288*288, the FPS is 91. When the input image is increased to 544*544, the FPS drops to 40. Therefore, from the data in Table 3, we can conclude that the number of train datasets for the network model is as large as possible, which can to some extent improve the detection accuracy of the network model. In addition, the input image size can be appropriately increased if it meets the requirements, which has a certain improvement in detection accuracy. SSD and YOLO are both one-stage object detection methods that perfectly trade off the accuracy and speed of the network model and, at the same time, lay the foundation for the subsequent real-time object detection algorithms to be proposed.

RetinaNet: RetinaNet [77] was proposed in 2017, which distinguished itself not through its network architecture but via the introduction of a novel loss function called focal loss. RetinaNet employs ResNet [78] as its foundational network and incorporates a feature pyramid network (FPN [79]) for multi-scale feature fusion. The authors highlight that single-stage object detection methods face a significant challenge with category imbalance, specifically, the disparity in sample numbers between foreground and background categories. To address this issue, they recommend modifying the conventional cross-entropy loss function to introduce focal loss. The method reduces the learning weight of simple background samples to a weighted form and focuses on difficult samples. In paper [77], RetinaNet can achieve 39.1% AP on the COCO dataset when using the ResNet-101-FPN structure, which is an improvement of 5.9% compared to the DSSD [80] (33.2%) using the ResNet-101 backbone network and compared to the two-stage network Faster R-CNN with TDM [81] (36.8%), which is an improvement of 2.3%. The introduction of focal loss has been significantly beneficial for one-stage object detection methods, mitigating the issue of category imbalance often found within the dense anchor boxes of such detection frameworks. This innovation has established a crucial groundwork for the advancement of object detection technology.

YOLOv3: YOLOv3 [43] is an improved version of YOLOv2, proposed in 2018. Based on YOLOv2, YOLOv3 makes its backbone deeper and uses Darknet-53 to extract richer features to further improve the detection performance of the model. In addition, YOLOv3 adds cross-layer connectivity to fuse the deep and shallow feature maps, which more fully utilizes the semantic information of different layers and further improves the detection of small object objects. YOLOv3 introduces an adaptive anchor box setting method, which automatically adjusts the size and aspect ratio of the anchor box according to the dataset, making the model more adaptive to the object features of different datasets and scenarios. In paper [43], the AP of YOLOv3 reaches 33.0% when detected on the COCO dataset, which is an improvement of 11.4% compared to YOLOv2 (21.6%). On a large dataset such as COCO, this improvement is a surprising result that speaks volumes about YOLOv3's improvement. Moreover, YOLOv3 can still achieve real-time detection with an FPS of 78 using a deep backbone like Darknet-53.

CornerNet: CornerNet [82] was proposed in 2018, which abandons the anchor box design in the unified stage object detection algorithm and instead accurately calibrates the position of objects through diagonal points (i.e. upper left and lower right corners). This end-to-end corner representation not only simplifies the model structure, but also reduces the complexity of the regression process, achieving more efficient detection performance. To better determine the location of corner points, the authors propose corner pooling, a pooling layer that determines the location of corner points based on the information of two edges near a pixel point. However, there may be more than one object in the image, so the corner points need to be grouped. The authors borrowed the method of multi-person human body pose estimation [83,84]. Each corner point generates an embedded vector. If the distance between two vectors is the smallest, then it will be used as a group of corner points. After grouping the corner points, the detected corner points are positionally fine-tuned to get a more accurate bounding box. In paper [82], CornerNet uses the Hourglass-104 [85] backbone network, and the AP can reach 42.2%, surpassing other one-stage object detection networks of the same period.

CenterNet: CenterNet [86] was proposed in 2019, which improves on CornerNet. The authors argue that CornerNet uses corner points, and its ability to refer to global objects is limited. Corner points are more sensitive to boundaries and do not know which object that pair of corner points belongs to, thus generating many incorrect bounding boxes and adding difficulty to the detection network. CenterNet adds a center point to CornerNet's two corner points to identify an object. With the addition of a center point, the network can perceive the visual patterns of each candidate region and thus better locate the object. In paper [86], when using the Hourglass-104 backbone network, CenterNet511 achieves 47.0% AP on the COCO dataset, which is a 4.9% improvement compared to CornerNet (42.1%). In 2023, the authors updated CenterNet with some of the latest backbone networks. In the new paper [87], the AP can reach 57.1% on the COCO dataset when using the Swin-L [88] backbone network. And a real-time object detector was designed with both accuracy and speed, with an AP of 43.6% and an FPS of 30.5.

FCOS: The fully convolutional one-stage (FCOS [89]) object detection was proposed in 2019, which is a fully convolutional one-stage anchor-free detection network. The core idea of FCOS is similar to

semantic segmentation [90–92] with pixel-by-pixel prediction. Specifically, all pixels are treated as positive samples, and each pixel is allowed to regress four edges to form a prediction frame. However, if each pixel regresses a prediction box, then the network will generate a large number of low-quality prediction boxes, increasing the network training burden. In order to obtain higher-quality prediction boxes, FCOS adopts a new sample selection strategy (center-ness) to limit positive and negative samples. The approach involves designating pixels with high scores that are near the object's center as positive samples, while those located further away are considered negative samples. FCOS avoids the definition of an anchor box by not using the anchor-based method, which reduces the computational effort of the network. More importantly, the hyperparameters associated with the anchor box are avoided, which in turn simplifies the design of the network. In paper [89], when using the ResNeXt-101-FPN [93] backbone network, FCOS achieves 44.7% AP on the COCO dataset, which is an improvement of 4.2% compared to CornerNet (40.5%) using Hourglass-104.

YOLOv4: YOLOv4 [44] is an improved version of YOLOv3, proposed in 2020. Based on YOLOv3, YOLOv4 employs a deeper backbone network, CSPDarknet-53, which combines the features of the cross-stage partial network (CSPNet [94]) with the Darknet network to enrich the gradient flow information and improve the performance of the network. In terms of feature fusion, YOLOv4 introduces spatial pyramid pooling (SPP [75]) and path aggregation network (PANet [95]), which perform top-down and bottom-up feature fusion of feature maps at different scales and improve the detection of objects at different scales. In addition, YOLOv4 uses some techniques called "Bag of Freebies" and "Bag of Specials", including MixUp [96] data enhancement, CutMix [97] data enhancement, Cross mini-Batch Normalization (CmBN, reference CBN [98]), etc., and uses these techniques to further enhance the performance of the network model and generalization ability. In paper [44], when tested on the COCO2017 test-dev dataset with the input image set to 608*608, YOLOv4 can achieve an AP of 43.5% and an FPS of about 65 on a Tesla V100.

YOLOv5: YOLOv5 [45] is an important improved version of the YOLO family, proposed in 2020, which laid the foundation for the development of the YOLO family. YOLOv5 utilizes a more lightweight model architecture with higher speed and smaller model sizes, which improves the speed of inference and the deployment efficiency of the model while maintaining good detection accuracy. YOLOv5 categorizes models into different versions, including YOLOv5-N, YOLOv5-S, YOLOv5-M, YOLOv5-L, and YOLOv5-X, etc., which perfectly trade-off model speed and accuracy. Among them, the YOLOv5-S model can reach 140 FPS and 37.4% AP, and the parameter count is only 7.2M, which plays a key role in real-time object detection tasks by maintaining high accuracy while also having higher inference speed and a smaller model size. In addition, YOLOv5 introduces a new data augmentation strategy and training strategy, including automatic data augmentation [99–101] and cross-scale training. These strategies enhances the model's generalizability and robustness, enabling it to handle a wide range of complex situations and variations. In order to present the detection effect of the YOLOv5 model more intuitively, Fig. 5 shows the detection results of the input images on three models of different sizes. The detection results are tested on the COCO2017 validation set using the officially provided pre-training weights.

YOLOX: YOLOX [102] is a new detection model proposed after YOLOv5, which was proposed in 2021. The YOLOX paper points out that the coupled detection head may impair the detection performance of the network and draws on the idea of anchor-free to change the YOLO model to an anchor-free approach. Its original coupled detection head is decoupled, and two branches are used to realize the classification and regression tasks. In terms of label assignment strategy, the SimOTA strategy is proposed by borrowing the optimal transport assignment (OTA [103]) strategy to consider it an optimal transport problem. In paper [102], the AP of YOLOX-S is 39.6%, which is 2.2% higher than that of YOLOv5-S (37.4%), and the number of parameters is 9.0 M. YOLOX has enhanced the model's detection capabilities with only a minimal rise in parameter count. The model architecture of YOLOX is notably straightforward and clear, facilitating easier understanding and implementation. This simplicity streamlines the model's deployment and application, making it more user-friendly and accessible.

YOLOv6: YOLOv6 [46] was proposed in 2022. Inspired by the article RepVGG [104], YOLOv6 changed its backbone network to EfficientRep, where the small model has a simple single-path backbone and the large model is built on efficient multi-branch blocks. And it draws heavily on recent ideas in network design, training strategies, testing techniques, quantization, and optimization methods, making YOLOv6 a SOTA approach. YOLOv6 adds a self-distillation [105] strategy to the network, performing both classification tasks and regression tasks. Dynamically adjusting knowledge from the teacher's network and labeling helps the student model learn knowledge more efficiently in all training phases. In paper [46], the AP of YOLOv6-S is 43.5%, which is a 3.9% improvement compared to YOLOX-S (39.6%). The FPS is 358 when the batch size is 1 and 495 when the batch size is 32. The YOLOv6 model, after absorbing a large number of advanced techniques, has been improved in accuracy and speed and has become a mainstream real-time object detection model. In 2023, the Meituan team proposed YOLOv6 v3.0 [106] version, which can achieve an AP of 57.2%. In order to present the detection effect of the YOLOv6 model more intuitively, Fig. 6 shows the detection results of the input images on three models of different sizes. The detection results are tested on the COCO2017 validation set using the officially provided pre-training weights.

YOLOv7: YOLOv7 [47] was proposed in 2022, and this paper improves on an efficient layer aggregation network (ELAN [107]) by proposing an Extended-ELAN (E-ELAN) structure. The architecture employs group convolution to widen the channel base and computational blocks. It also incorporates strategies like expansion, shuffling, and merging cardinality to enhance the network's learning capacity while ensuring the network's original gradient remains intact. The YOLOv7 paper also uses structural reparameterization [104] ideas to make changes to the backbone network and introduces a model scaling strategy [108,109] that primarily generates models with different scales to fulfill the needs of different inference speeds. In terms of the detection head, YOLOv7 added an auxiliary head in the training phase to improve the detection accuracy by increasing the training cost without affecting the inference speed. At that time, the YOLOv7 detection network exceeded the detection accuracy of the network between 5 FPS and 160 FPS, and the AP for YOLOv7-tiny-SiLU on the COCO2017 test set reached 38.7%, and the AP for YOLOv7-E6E was 56.8%, which was quite a good result at that time. In order to present the detection effect of the YOLOv7 model more intuitively, Fig. 7 shows the detection results of the input images on two models of different sizes. The detection results are tested on the COCO2017 validation set using the officially provided pre-training weights.

ObjectBox: ObjectBox [110] was proposed in 2022. The method is a one-stage anchor-free detection network. ObjectBox uses only the object center position (a point) as a positive sample and treats all objects equally at different feature levels, regardless of the size or shape of the object. It means that ObjectBox treats the object's center position as a shape-agnostic and size-agnostic anchor. However, the ablation experiments in the paper [110] found that the accuracy of regression using only the center point is very low, so this paper maps the center point of the GT (Ground Truth) box to the upper-left and lower-right corners of this pixel cell, using these two corner points to regress (the upper-left point is responsible for regressing the right border and lower border, and the lower-right point is responsible for regressing the left border and upper border), and the four edges form a prediction box, and the experiment proves that the method is very effective. In paper [110], when using the CSPDarknet backbone network, an AP of

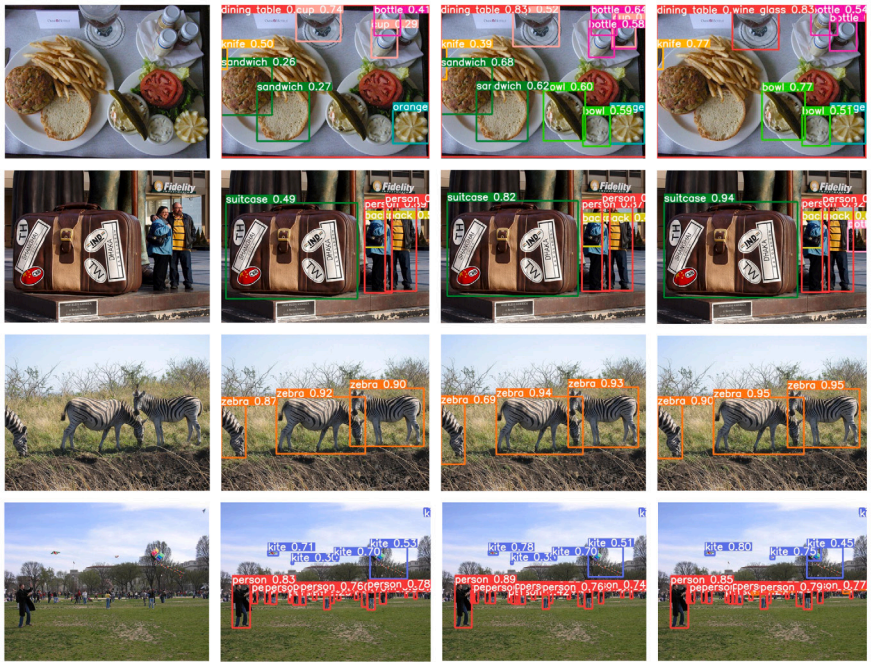


Fig. 5. The visual results of YOLOv5 on the COCO2017 validation set.

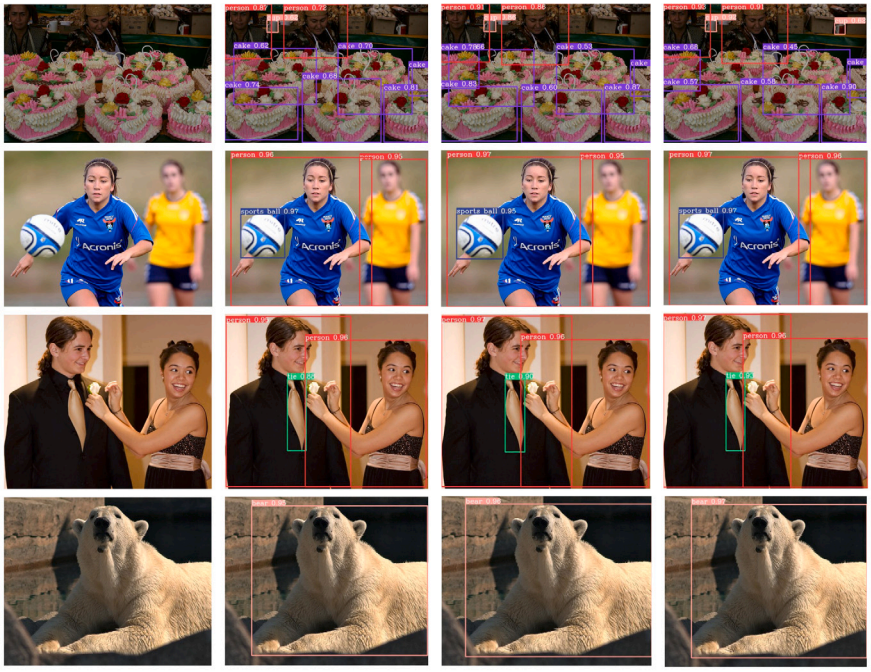


Fig. 6. The visual results of YOLOv6 on the COCO2017 validation set.

46.8% can be achieved on the COCO dataset. Compared to CenterNet (41.6%) with Hourglass-52, an improvement of 5.2% is achieved. From the results, the ObjectBox detector brings some accuracy improvement, which contributes to the development of anchor-free detectors.

GiraffeDet: GiraffeDet [111] was proposed in 2022, which breaks the paradigm of traditional object detection algorithms of "body-heavy, neck-light". General object detection algorithms have a very deep backbone network, which is used to extract deep potential features from the body. The neck is a shallow module that fuses these potential features to capture information at different scales. The method hypothesizes that the neck is more important than the backbone network in object detection algorithms from another perspective, and experiments are done to verify that the hypothesis holds. So the authors designed a lightweight backbone to extract features, paired with a heavy-duty feature fusion module with a structure similar to a giraffe. The traditional pyramid just fuses the feature maps of the same and previous levels. The authors argue that this approach does not fully capitalize on the semantic information from deep features and the spatial information

Fig. 7. The visual results of YOLOv7 on the COCO2017 validation set.

from shallow features. Therefore, they propose a novel feature fusion technique termed "queen fusion". This fusion takes into account features of the same level and neighboring levels, just like playing queen pieces in chess, and adds rich jump-layer connections to the pyramid structure to ensure that the original feature map information is not lost. From their experiments, the authors found that the network detection accuracy rises significantly as the depth of the pyramid deepens. In paper [111], the highest accuracy is achieved when the neck depth is stacked to 29 layers, and GiraffeDet-D29 can achieve 54.1% AP on the COCO dataset.

DAMO-YOLO: DAMO-YOLO [112] was proposed in 2022, which is a faster and more accurate detector. DAMO-YOLO gathers a lot of advanced techniques, and the backbone network uses MAE-NAS [113] to get the optimal backbone network to extract richer feature map information. In terms of neck selection, the authors believe that the queen fusion approach of Generalized-FPN [111] cannot meet the real-time requirements, and the additional up-and-down sampling is time-consuming. Therefore, the authors proposed Efficient RepGFPN, which uses CSP [94] structure, RepConv [104], and Efficient Layer Aggregation Networks (ELAN [107]), respectively, and the structure can well meet the real-time requirement of the YOLO series. Furthermore, the authors discovered through their experiments that a larger neck and a smaller head configuration are more optimal. As a result, they introduced a concept called "zero head", which specifically employs a linear layer to separately accomplish classification and regression tasks. Of course, the paper also combines some other advanced techniques to improve the performance of the network, such as AlignOTA and knowledge distillation [114]. In paper [112], DAMO-YOLO achieves relatively good accuracy and speed, while DAMO-YOLO-M has an AP of 50.0% on the COCO dataset with the distillation technique.

YOLOv8: YOLOv8 [48] was proposed in 2023 and is an extended version of the YOLO series. YOLOv8 cannot only realize object detection tasks. It can also accomplish tasks such as image classification and semantic segmentation. In the detection head part, the coupling head of YOLOv5 is abandoned and replaced by a decoupling head using an anchor-free strategy. In addition, YOLOv8 combines the ideas of C3 and Efficient Layer Aggregation Network (ELAN [107]) to design a C2f module instead of the original C3 module, which further realizes the lightweight nature of the model and can be used to obtain richer gradient flow information. The classification loss function uses VariFocal loss (VFL [115]), the regression loss uses CIOU loss, and the IOU matching assignment method is discarded, and the task-aligned assigner matching method is used instead. YOLOv8 has a great improvement in accuracy. The AP of YOLOv8-S on the COCO dataset is 44.9%, which is 1.4% higher than YOLOv6-S (43.5%). In order to present the detection effect of the YOLOv8 model more intuitively, Fig. 8 shows the detection results of the input images on three models of different sizes. The detection results are tested on the COCO2017 validation set using the officially provided pre-training weights.

CFPNet: CFPNet [116] was proposed in 2023, and this paper focuses on doing some work on FPN [79]. The authors believe that the existing methods are overly focused on inter-layer feature interactions while ignoring intra-layer feature rules. In order to solve this problem, a centralized feature pyramid (CFP) is proposed, which is structured to process the last feature layer, and an explicit visual center (EVC) structure is designed to process the deep feature maps. The EVC consists of two parts: the first part is a lightweight MLP to capture global long-range dependencies, the structure is introduced from the transformer [117]. The authors argue that traditional transformers [118–120] are based on multi-attention feature interactions to

Fig. 8. The visual results of YOLOv8 on the COCO2017 validation set.

capture global dependency correlations, but an obvious drawback is their large computational complexity. And the second part is a parallel learnable visual center (LVC) mechanism, which is used to capture the local corner regions of the input image. The EVC-processed feature maps are then fused with the shallow feature maps so that the explicit visual center information obtained from the deeper features can be used to modulate the shallow features in order to achieve better detection results. In paper [116], the authors use YOLOv5 and YOLOX as the baseline and add their own innovations, and the accuracy is greatly improved. In particular, $CFP_{YOLOv5-L}$ has an AP of 46.0% on the COCO dataset, which is an improvement of 0.8% compared to YOLOv5-L (45.2%). $CFP_{YOLOv5-L}$ has an AP of 49.4% on the COCO dataset, which is an improvement of 1.6% compared to YOLOX-L (47.8%).

InternImage: InternImage [121] was proposed in 2023. This paper compares ViTs [118] and CNNs and finds that CNNs based on large models are still at an early stage. Based on this problem, the authors' team proposed a large-scale CNN-based basic network model. InternImage established a new record for accuracy in object detection on the COCO dataset and for semantic segmentation on the ADE20K [122] dataset. Transformer [123–127] is a hot large-scale model recently, which has the advantages of long-range dependency and adaptive spatial aggregation, but its computational volume is huge. Therefore, the authors improve DCNv2 [128] (deformable convolution) to propose DCNv3 and design the base module with reference to the structure of the transformer. Firstly, they utilize separable convolution [129] with shared weights to decrease the number of parameters and complexity. Secondly, a grouping mechanism is introduced, where different groups on the convolution can have different spatial aggregation patterns. Finally, the modulation scalar is standardized along the sampling points, using softmax instead of sigmoid, and the gradient is more stable. With these three improvements, the convolutional model can have the advantages of the transformer model, and since DCNv3 is a dynamically sparse convolution, it is not very computationally intensive. In paper [121], InternImage has an AP of 65.4% on the COCO dataset.

1DSNet: 1DSNet [50] was proposed in 2023. This paper finds that existing self-attention mechanisms [130–132] and their variants focus on spatial and channel dimensions, ignoring the width and height dimensions. The authors use one-dimensional convolution to weight the width and height dimensions of the feature maps by compressing the height and width dimensions separately and then performing a weighted summation with the input feature maps to obtain the feature map information in the width and height dimensions. In addition, the authors added the receptive field dilated branch (RFD) to spatial pyramid pooling (SPP) to form the receptive field dilated spatial pyramid pooling (RFD-SPP). This is done by expanding the receptive field of the network through dilated convolution [133], thus obtaining richer feature map information. In paper [50], 1DSNet achieves relatively good accuracy on the COCO dataset, with an AP of 37.9%.

Gold-YOLO: Gold-YOLO [134] was proposed in 2024, and this paper breaks the traditional way of fusing multi-scale features. The traditional feature pyramid tends [79,95,135–138] to lose information when fusing multi-scale feature information across layers, so this paper proposes a brand new mechanism: the gather-and-distribution (GD) mechanism. The gather mechanism collects the feature map information at different scales in the last layers of the backbone network in a unified way and fuses the multi-scale feature map information. Then the distribution mechanism is utilized to issue global information to deeper feature maps, which avoids the disadvantage of traditional feature fusion in that information is lost when fusing across layers. In addition, Gold-YOLO implements MAE-style [139] pre-training for the first time in the YOLO family, allowing the network to benefit from unsupervised pre-training. In paper [134], Gold-YOLO achieves relatively good accuracy. On the COCO dataset, Gold-YOLO-L has an AP of 53.3% with 116 FPS, realizing the requirement of real-time detection. Gold-YOLO-N has an AP of 39.9% with 1030 FPS, which is a 2.4% improvement compared to YOLOv6-3.0-N (37.5%).

YOLOv9: YOLOv9 [49] was proposed in 2024. This author was also involved in the research on YOLOv7 and YOLOv4. In order to prevent the problem of data loss that is easily caused when data is transmitted through deep networks and undergoes layer-by-layer feature extraction and spatial transformation, this paper analyzes the existing deep neural network architectures from the perspective of reversible functions. On this basis, YOLOv9 proposes the concept of programmable gradient information (PGI), which can provide complete

Fig. 9. The visual results of YOLOv9 on the COCO2017 validation set.

input information for the target task to compute the objective function, so as to obtain reliable gradient information for updating the network weights. YOLOv9 designs a new lightweight network architecture on the basis of efficient layer aggregation network (ELAN [139]), named generalized efficient layer aggregation network (GELAN). The design of GELAN takes into account the number of parameters, computational complexity, accuracy, and inference speed simultaneously. This design achieves higher parameter utilization than deep convolutional designs based on state-of-the-art technology by using only standard convolution, showing the great advantages of lightness, speed, and accuracy. In paper [49], YOLOv9-S is used as a lightweight network with an AP of 46.8%, which outperforms existing real-time object detectors. In order to present the detection effect of the YOLOv9 model more intuitively, Fig. 9 shows the detection results of the input images on two models of different sizes. The detection results are tested on the COCO2017 validation set using the officially provided pre-training weights.

Table 4 shows the test results of the one-stage object detection algorithms we mainly discussed above on the COCO dataset. The table mainly contains method, backbone, AP, AP_{50}, AP_{75}, AP_S, AP_M, and AP_L. From the data in the table, it can be seen that the accuracy of the one-stage object detection methods is continuously improving. YOLOv2 [42] and SSD [40] are the early object detection methods, which are not very good at small object detection accuracy, resulting in low overall accuracy. SSD300 and SSD512 have small object accuracy AP_S of 5.3% and 9.0%, respectively, and YOLOv2 has an AP_S of 5.0%, which is due to the fact that the object localization of the early one-stage methods is more rough. In addition, the multi-scale fusion structure was not introduced in the early stages, the resolution of the small objects was low, and the extracted feature information was limited.

In 2017, He et al. proposed a feature pyramid structure (FPN [79]) to obtain multi-scale contextual information by top-down fusion of feature maps at different scales. Inspired by FPN, many new multi-scale fusion structures have been proposed subsequently, such as PANet [95], BiFPN [135], Generalized-FPN [111], etc. These structures can adequately fuse deep and shallow feature information to improve the accuracy of object detection for multi-scale objects. From the experimental data of RetinaNet [115], it can be seen that the detection accuracy of small objects is effectively improved after the introduction of FPN in the backbone network. Among them, the AP_S of RetinaNet using the ResNet-101-FPN backbone network is 21.8%. YOLOv3 [43] also adopts multi-scale fusion for prediction and updates the backbone network depth, with an AP_S of 18.3%, which is improved by 13.3% compared to YOLOv2 [42] (5.0%).

The backbone network used by CornerNet [82] and CenterNet [86] is Hourglass, which was initially used for human pose estimation tasks and introduced by the authors for object detection tasks. Hourglass network [85] is composed of multiple blocks of hourglass by stacking them, and there are corresponding up-sampling and down-sampling operations in each block, which allows to capture the feature map information at different levels and output heat maps matching the input size. This design makes the hourglass network more flexible in dealing with images of different sizes, but the computational volume is also larger, which affects the inference speed of the model, and the resource requirement is higher. The overfitting phenomenon is likely to occur if the amount of train data is insufficient or the distribution of the data is not uniform. Hourglass is divided into Hourglass-52 and Hourglass-104, with AP of 40.6% and 44.9% when both CornerNet and CenterNet use Hourglass-104. CenterNet adds a center point to

Table 4
The test results of common one-stage object detection algorithms on the COCO dataset. "*" represents multi-scale detection result. "–" indicates that the data is temporarily unknown.

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
SSD300 [40]	VGG-16	23.2	41.2	23.4	5.3	23.2	39.6
SSD512 [40]	VGG-16	26.8	46.5	27.8	9.0	28.9	41.9
YOLOv2 [42]	Darknet-19	21.6	44.0	19.2	5.0	22.4	35.5
RetinaNet [77]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet [77]	ResNeXt-101-FPN	40.8	61.1	44.1	24.1	44.2	51.2
YOLOv3 [43]	Darknet-53	33.0	57.9	34.4	18.3	35.4	41.9
CornerNet511 [82]	Hourglass-104	40.6	56.4	43.2	19.1	42.8	54.3
CornerNet511* [82]	Hourglass-104	42.2	57.8	45.2	20.7	44.8	56.6
CenterNet511 [86]	Hourglass-52	41.6	59.4	44.2	22.5	43.1	54.1
CenterNet511 [86]	Hourglass-104	44.9	62.4	48.1	25.6	47.4	57.4
CenterNet511* [86]	Hourglass-52	43.5	61.3	46.7	25.3	45.3	55.0
CenterNet511* [86]	Hourglass-104	47.0	64.5	50.7	28.9	49.9	58.9
FCOS [89]	ResNet-101-FPN	41.5	60.7	45.5	24.4	44.8	51.6
FCOS [89]	HRNet-W32–51	42.0	60.4	45.3	25.4	45.0	51.0
FCOS [89]	ResNeXt-32 × 8d-101-FPN	42.7	62.2	46.1	26.0	45.6	52.6
FCOS [89]	ResNeXt-64 × 4d-101-FPN	44.7	64.1	48.4	27.6	47.5	55.6
YOLOv4 416 [44]	CSPDarknet-53	41.2	62.8	44.3	20.4	44.4	56.0
YOLOv4 512 [44]	CSPDarknet-53	43.0	64.9	46.5	24.3	46.1	55.2
YOLOv4 608 [44]	CSPDarknet-53	43.5	65.7	47.3	26.7	46.7	53.3
YOLOv5-S [45]	Modified CSP v6.1	37.4	56.8	–	–	–	–
YOLOv5-M [45]	Modified CSP v6.1	45.4	64.1	–	–	–	–
YOLOv5-L [45]	Modified CSP v6.1	49.0	67.3	–	–	–	–
YOLOv5-X [45]	Modified CSP v6.1	50.7	68.9	–	–	–	–
YOLOX-S [102]	Modified CSP v5	39.6	–	–	–	–	–
YOLOX-M [102]	Modified CSP v5	46.4	65.4	50.6	26.3	51.0	59.9
YOLOX-L [102]	Modified CSP v5	50.0	68.5	54.5	29.8	54.5	64.4
YOLOX-X [102]	Modified CSP v5	51.2	69.6	55.7	31.2	56.1	66.1
YOLOv6-S [106]	EfficientRep	43.5	60.4	–	–	–	–
YOLOv6-M [106]	EfficientRep	49.5	66.8	–	–	–	–
YOLOv6-L [106]	EfficientRep	52.5	70.0	–	–	–	–
YOLOv7-tiny [47]	E-ELAN	35.2	52.8	37.3	15.7	38.0	53.4
YOLOv7 [47]	E-ELAN	51.2	69.7	55.5	35.2	56.0	66.7
YOLOv7-X [47]	E-ELAN	52.9	71.1	57.5	36.9	57.7	68.6
ObjectBox [110]	ResNet-101	46.1	65.0	48.3	26.0	48.7	57.3
ObjectBox [110]	CSPDarknet	46.8	65.9	49.5	26.8	49.5	57.6
GiraffeDet-D7 [111]	S2D-Chain	45.6	–	–	28.8	48.7	55.6
GiraffeDet-D11 [111]	S2D-Chain	46.9	–	–	29.9	51.1	58.4
GiraffeDet-D14 [111]	S2D-Chain	47.7	–	–	30.9	51.6	60.3
GiraffeDet-D16 [111]	S2D-Chain	48.7	–	–	31.7	52.4	61.3
GiraffeDet-D25 [111]	S2D-Chain	50.5	–	–	32.2	54.2	63.5
GiraffeDet-D29 [111]	S2D-Chain	51.3	–	–	33.1	54.9	64.9
DAMO-YOLO-T [112]	MAE-NAS	43.0	59.4	46.6	23.3	47.4	61.0
DAMO-YOLO-S [112]	MAE-NAS	46.8	63.5	51.1	26.9	51.7	64.9
DAMO-YOLO-M [112]	MAE-NAS	50.0	66.8	54.6	30.4	54.8	67.6
$CFP_{YOLOv5-S}$ [116]	Modified CSP v5	36.0	56.2	47.8	22.8	42.9	51.6
$CFP_{YOLOv5-M}$ [116]	Modified CSP v5	43.2	62.9	48.5	29.1	49.4	53.3
$CFP_{YOLOv5-L}$ [116]	Modified CSP v5	46.6	64.9	50.0	30.4	51.7	59.5
$CFP_{YOLOX-S}$ [116]	Modified CSP v5	41.1	60.0	44.5	24.2	45.4	54.5
$CFP_{YOLOX-M}$ [116]	Modified CSP v5	46.4	65.1	50.3	29.4	51.2	60.5
$CFP_{YOLOX-L}$ [116]	Modified CSP v5	49.4	67.8	53.6	32.4	54.3	64.0
InternImage-H [121]	InternImage	65.4	–	–	–	–	–
1DSNet [50]	DSPDarknet-53	37.9	55.7	41.4	23.9	43.6	47.5
Gold-YOLO-S [134]	MAE	46.4	63.4	–	–	–	–
Gold-YOLO-M [134]	MAE	51.1	68.5	–	–	–	–
Gold-YOLO-L [134]	MAE	53.3	70.9	–	–	–	–
YOLOv8-N [48]	Modified CSP v8	37.3	–	–	–	–	–
YOLOv8-S [48]	Modified CSP v8	44.9	–	–	–	–	–
YOLOv8-M [48]	Modified CSP v8	50.2	–	–	–	–	–
YOLOv8-L [48]	Modified CSP v8	52.9	–	–	–	–	–
YOLOv8-X [48]	Modified CSP v8	53.9	–	–	–	–	–
YOLOv9-S [49]	Modified CSP v9	46.8	63.4	50.7	26.6	56.0	64.5
YOLOv9-M [49]	Modified CSP v9	51.4	68.1	56.1	33.6	57.0	68.0
YOLOv9-C [49]	Modified CSP v9	53.0	70.2	57.8	36.2	58.5	69.3
YOLOv9-E [49]	Modified CSP v9	55.6	72.8	60.6	40.2	61.0	71.4

CornerNet to determine the position of the object, which can more accurately locate the object in the image, and therefore the accuracy is higher.

FCOS [89] uses four different backbone networks: ResNet-101-FPN [78], HRNet-W32-51 [140], ResNeXt-32x8d-101-FPN [93], and ResNeXt-64x4d-101-FPN [93], respectively. From the results of FCOS, it can be seen that the different backbones have different results on the detection accuracy of the network model. When using ResNeXt-64x4d-101-FPN, the AP is the highest at 44.7%, which is 3.2% higher compared to using ResNet-101-FPN (41.5%). ResNet introduces residual connections in deep neural networks to address gradient vanishing and explosion issues during training. While ResNeXt maintains a similar basic structure, it introduces the concept of cardinality to simplify the network design. This not only enhances network performance but also reduces the number of parameters. However, ResNeXt widens the network, leading to increased computational cost and a greater number of hyper-parameters compared to ResNet, necessitating careful parameter adjustment. HRNet is a commonly used backbone network for computer vision recognition tasks that can compute feature maps of different resolutions and retain the feature information of high-resolution images so that the detail information can be better captured. However, multiple-resolution images need to be processed, and the computational complexity will be relatively high. FCOS uses HRNet, with an AP of 42.0%.

YOLOv4 [44] has updated the backbone network to include a cross stage partial (CSP [94]) structure, named CSPDarknet53. The CSP connection allows feature maps from different layers to interact with each other, which improves the representation of the network and helps to capture more detailed feature information. YOLOv4 uses 416, 512, and 608. There are three different sizes of input images with different resolutions, with AP of 41.2%, 43.0%, and 43.5% and AP_S of 20.4%, 24.3%, and 26.7%, respectively. The findings from this experiment indicate that larger input images enable the network to extract more detailed information from small objects, thereby increasing the accuracy. However, an increase in the input image also means that the computational complexity becomes larger and the inference speed of the network is affected, so we have to weigh the relationship between speed and accuracy when designing the network.

YOLOv5 [45] divides the network into YOLOv5-S, YOLOv5-M, YOLOv5-L and YOLOv5-X, etc. based on the depth and width of the network and also provides a reference for the subsequent YOLO series. Through this version of the division, it allows scholars with limited hardware resources to complete the experiment. YOLOv5 not only improves accuracy but also lightens the model and greatly reduces the training time. Influenced by YOLOv5, researchers subsequently proposed high-quality papers such as YOLOX [102], YOLOv6 [46], YOLOv7 [47], and so on. YOLOX has improved its detection head based on YOLOv5 by replacing the coupling head with a decoupling head and using anchor-free. As can be seen from Table 4, the accuracy of YOLOX has some improvement compared to YOLOv5. The AP of YOLOX-S is 39.6%, which is improved by 2.2% compared to YOLOv5-S (37.4%). Similarly, YOLOX-M, YOLOX-L, and YOLOX-X are improved by 1.0%, 1.0%, and 0.5%, respectively. YOLOv6 is affected by the idea of structural reparameterization [104] influence; a new backbone network, EfficientRep, is proposed. Multi-branch structure is used to obtain rich feature map information during training, and single-branch structure is used for inference. This not only improves the network detection accuracy but also does not affect the inference speed of the network. The AP of YOLOv6-S (43.5%), YOLOv6-M (49.5%), and YOLOv6-L (52.5%) are improved by 6.1%, 4.1%, and 3.5% compared to YOLOv5-S (37.4%), YOLOv5-M (45.4%), and YOLOv5-L (49.0%), respectively. The anchor-based approach is still adopted in YOLOv7. It still meets the requirement of real-time detection while maintaining high accuracy. The AP of both YOLOv7 and YOLOX-X is 51.2%, but the small object detection accuracy of YOLOv7 is higher than that of YOLOX. The AP_S of YOLOv7 is 35.2%, compared with that of YOLOX-X (31.2%), which is 4.0%, which indicates that YOLOv7 has improved a lot in small object detection. However, YOLOv7 is sensitive to hyper-parameters and requires careful parameter tuning. YOLOv8 [48] has improved compared to YOLOv7. YOLOv8-N has an AP of 37.3%, which is 2.1% higher compared to YOLOv7-tiny (35.2%). YOLOv8-X has an AP of 53.9%, which is 1.0% higher compared to YOLOv7-X (52.9%). Moreover, YOLOv8 still maintains its lightweight and high inference speed. ObjectBox [110] has still changed the network model to anchor-free based on the YOLO series, and the result is very good. ObjectBox has an AP of 46.8% when using the CSPDarkNet backbone. The difference between the AP and YOLOv8-M (49.5%) using the EfficientRep backbone is only 2.7%, indicating that the point-based detection method is an improvement for the YOLO series. YOLOv9 [49] is the newest method in the YOLO series, and it outperforms other YOLO detectors. The AP of YOLOv9-S is 46.8%, which is an improvement of 1.9% compared to YOLOv8-S (44.9%). The AP of YOLOv9-M is 51.4%, which is an improvement of 1.2% compared to YOLOv8-M (50.2%).

GiraffeDet [111] divides six networks according to the neck depth. The backbone networks all use S2D-Chain, where space-to-depth (S2D) converts spatial-dimensional information into depth-dimensional information. GiraffeDet increases the AP as the neck depth increases, confirming the conjecture in the paper that the neck is more important than the backbone network. DAMO-YOLO [112], CFPNet [116], and Gold-YOLO [134] are all extensions of the YOLO family. DAMO-YOLO uses the MAE-NAS [113] technique to get the optimal backbone network, and the process of searching for the optimal network is not time-consuming. DAMO-YOLO divides the network into three, namely, DAMO-YOLO-T, DAMO-YOLO-S, and DAMO-YOLO-M, with AP of 43.0%, 46.8%, and 50.0%, respectively. Gold-YOLO is an extended version of the YOLO series, updating the neck of the network to overcome the problem of information loss during feature fusion. Gold-YOLO has some improvement compared to DAMO-YOLO, and Gold-YOLO-M (51.1%) has an improvement of 1.1% compared to DAMO-YOLO-M (50.0%). InternImage [121] designed DCNv3 and constructed a new network that set a new accuracy record for object detection with an AP of 65.4% on the COCO dataset.

4. Applications of object detection

In this section, we present an overview of various common object detection applications, encompassing face detection, defect detection, remote sensing image detection, text detection, pedestrian detection, and numerous other domains.

4.1. Face detection

Face detection [141–143] is a very classical object detection application that has gone through several stages of development. So far, face detection technology has become very mature and relevant to our daily lives. The following are the stages that face detection has gone through.

Early stage: In the early stages, face detection usually utilizes simple image processing techniques such as edge detection, color segmentation, and geometric feature extraction. Early face detection can achieve high recognition accuracy in some simple scenes and improve the efficiency of face recognition. However, early face detection techniques are not robust and generalizable and are limited by computer performance. In the face of occlusion and expression changes, the face detection ability is limited, which may lead to detection failure.

Feature engineering stage: In 2001, the Viola–Jones face detector was introduced, marking the beginning of the feature engineering era for face detection [57] technology, where detection was achieved through manually designed features and classifiers. Typical feature representations in the feature engineering era are Haar features [57] and HOG [71] features, and the classifiers are AdaBoost [58] and SVM [74]. The Viola–Jones face detector mainly uses Haar features and AdaBoost classifiers, which realized real-time face detection with relatively good

detection accuracy at that time. The Viola–Jones face detector greatly contribute to the advancement of face detection technology and still plays an extremely important role today.

Deep learning stage: With the rapid development of deep learning technology, face detection technology has entered the deep learning era. The swift advancement of convolutional neural networks has led to the replacement of manual feature extraction, with deep learning models now capable of automatically learning the feature representation of face images. This development has significantly propelled the progress of face detection technology. Representative models of convolutional neural networks include Faster R-CNN [33], SSD [40], MTCNN [144], etc. And with the improvement of hardware capability, face detection technology has been extended and promoted, such as face recognition [145,146], face verification [147,148], face tracking [149,150], and so on.

Although facial detection technology is already very mature, there are still some difficulties and challenges, summarized as follows.

- **Occlusion and complex background:** The background in daily life is often complex, and there may be some objects that resemble human faces and produce false detections. Factors such as occlusion, wearing glasses, or hats can pose challenges to the feature extraction process in face detection techniques, consequently impacting the performance of face detectors.
- **Age difference:** The appearance characteristics of the face will change with increasing age, which requires the face detector to have strong generalization ability.
- **Real time detector:** In today's daily life, the requirement for real-time performance is relatively common, and the detector needs to meet the requirement of real-time performance. For example, in scenarios such as video surveillance and mobile applications, face detection technology needs to complete facial detection tasks in a relatively short amount of time.
- **Privacy and security:** Face detection technology often involves the personal privacy of customers in daily life applications, requiring reasonable means to obtain user information and ensure that user information is not leaked or abused.

In summary, we need to pay more attention to the above issues when designing face detection applications. As much as possible, we need to ensure that the detector has strong enough generalization ability and real-time performance and can protect customer privacy. In terms of algorithms, they need to be designed to meet the needs of different scenarios and improve detector accuracy.

4.2. Defect detection

Defect detection [151–153] is often used for quality inspection in manufacturing and industry and is a very important method in industrial development. Defect detection aims to identify and locate flaws in products during industrial production to ensure product quality meets standard requirements, enhance production efficiency, and minimize production costs. Defect detection has undergone the evolution of many methods, the following are some of the classic defect detection methods.

Threshold-based methods: Threshold-based defect detection methods are relatively simple and intuitive. A threshold is usually set to determine whether there is a defect in an image. If the pixel value in the image exceeds a specified threshold, it is identified as a defect. Threshold-based methods can be subdivided into global thresholding [154], adaptive thresholding [155], dual thresholding [156], Otsu thresholding [157], etc.

Texture analysis-based methods: Texture analysis-based defect detection methods typically examine an image's texture characteristics to pinpoint defect locations. Numerous prevalent texture analysis techniques exist, such as the Gray Level Co-occurrence Matrix (GLCM [158]) and Local Binary Pattern (LBP [159]), among others.

Deep learning-based methods: Deep learning-based methods extract image information through convolutional neural networks, design networks for defect detection tasks, and automatically learn image feature information. Common networks include U-Net [160], Mask R-CNN [34], etc.

The progress in computer vision technology has resulted in enhanced accuracy for defect detection and the maturation of its industrial applications. However, there are also certain difficulties and challenges in defect detection, as summarized below.

- **Product diversity:** In industrial production, there are various types of product defects. There are significant differences in shape, color, size, and other aspects. Therefore, designing a detector that can adapt to various types of defects is difficult and also a challenge for the future.
- **Sample imbalance problem:** In a normal defect detection dataset, the number of defect samples in a product is often less than the number of normal samples, which can easily lead to positive and negative sample imbalances during detection, thereby affecting the performance of the defect detection model.
- **Complex light conditions:** Under strong-light conditions, the surface of the product may produce reflections and shadows, and the same defect may present different appearances under different lighting conditions, thereby affecting defect detection performance. In low-light conditions, the image contrast diminishes, making defects less discernible. Additionally, images under such conditions are more susceptible to noise interference.
- **Real time:** In industrial production, real-time is very important to ensure the normal operation of the production line. So defect detection not only requires high accuracy but also fast detection speed. Traditional defect detectors often fail to meet the requirements of real-time detection speed, and pursuing real-time performance is a challenge.

In summary, defect detection is indispensable for industrial production, but designing a defect detection model with high robustness and strong generalization is a challenge. Designers need to design detectors suitable for different environmental conditions.

4.3. Remote sensing image detection

Remote sensing image detection [161–163] is a very important application in object detection tasks, which has extremely important applications in many fields such as urban planning, military exploration, and architectural exploration. The following are the two development stages of remote sensing image detection.

Early stage: Early remote sensing image detection was limited by remote sensing technology, with low image resolution and a relatively limited detection distance, making it difficult to extract targets. The remote sensing image detection technology during this period was mainly based on pixel change detection, performing algebraic operations on pixels to generate difference maps between pixels, and then obtaining the results of changes through set thresholds. Common methods for monitoring changes include algebraic methods, transformation methods, classification methods, etc.

Deep learning stage: The onset of the deep learning era has also propelled advancements in remote sensing image detection. Classification and regression trees, convolutional neural networks, backpropagation algorithms, etc. have all improved the accuracy of remote sensing image detection. Specifically, recurrent neural networks (RNNs [164]) are commonly used to process sequential data, such as time and spatial sequences. In remote sensing image detection, RNN can process satellite images of time series, capture temporal dependencies in satellite images, and use them for image detection and analysis. A notable example of such networks is the Long Short-Term Memory Network (LSTM [165]). The Generative Adversarial Network (GAN [166]) is

comprised of two components: a generator and a discriminator. In remote sensing image detection tasks, the generator can generate as realistic remote sensing images as possible, and the discriminator is used to determine whether the generated remote sensing images are input into the real dataset. GAN can perform super-resolution reconstruction of remote sensing images, remove clouds and fog, etc.

With the continuous advancement of technology, the detection effect of remote sensing images is also getting better and better. However, remote sensing image detection technology still faces certain difficulties and challenges, summarized as follows.

- **Diversity of remote sensing images:** Obtaining remote sensing images typically involves methods like aerial and drone photography. There are various types of surface objects, each with complex shapes, textures, and colors. In addition, images from different sources may have different levels of noise and resolution, which puts higher demands on the detection performance and generalization ability of remote sensing images.
- **Big data processing:** Remote sensing image datasets usually contain a large amount of spatiotemporal information, which needs to be processed on a large scale before detection. However, traditional object detection algorithms often cannot efficiently process the temporal information of remote sensing images. Therefore, how to efficiently complete data processing and design detectors that can meet real-time and efficient requirements is a challenge.
- **Land change:** Remote sensing images are different from other datasets and have a certain degree of temporal variability. Surface objects will undergo changes over time, such as crop growth and harvesting, house renovations, etc. These changes require remote sensing image detection models to have strong robustness.
- **Data quality and noise:** When obtaining remote sensing images in the air, they are subject to various environmental conditions, such as cloud cover, surface reflection, atmospheric scattering, etc. These objective factors will directly affect the quality of remote sensing image detection, posing certain challenges to remote sensing image detection.

In summary, remote sensing image detection has value in multiple fields and provides convenience for daily life. However, there are still some difficulties and challenges that need to be overcome in remote sensing image detection.

4.4. Text detection

Text detection [167–169] is a relatively ancient application in computer vision tasks, mainly identifying text from videos or images and locating its position. Text detection can be used in many application scenarios in daily life, including document recognition [170], licence plate recognition [171], and so on. The development of text detection has gone through multiple stages, and the following are some classic text detection methods.

Edge detection-based methods: One of the initial approaches employed for text detection is edge detection. This technique typically leverages edge detection operators to identify the boundaries of preprocessed images. Common examples of edge detection operators include Sobel and Canny. These operators determine the position of text in a graph based on extracted geometric attributes such as corners, edges, and strokes.

Connectivity region-based methods: Text detection techniques that utilize connected regions typically divide the image into various connected regions. Widely used algorithms for extracting these regions include Maximally Stable Extremal Regions (MSER [172]) and Stroke Width Transform (SWT [173]), among others. Subsequently, the text feature information of each connected region is extracted, including morphological features, color features, etc., and they are used to realize the text region and non-text region.

Sliding Window-based methods: The text detection methods that use a sliding window approach typically utilize several sliding windows of different sizes to scan the image. These windows capture feature information, which is subsequently examined to assess if it includes text. Each sliding window undergoes classification using a pre-trained classifier. In contrast, deep learning-based methods commonly leverage convolutional neural networks to automatically extract pertinent feature information from images for text detection.

Deep learning-based methods: Text detection methods based on deep learning typically use convolutional neural networks to extract image feature information. At the same time, attention mechanisms and other methods will be used to weight channels or spaces to obtain more important information. Subsequently, perform text detection tasks.

Clustering-based methods: Clustering-based approaches are widely used in unsupervised learning within machine learning to separate image pixels into text and non-text areas. Text detection is performed by identifying the distinctive features of text. Among the clustering methods used are K-means [174], hierarchical [175], and spectral clustering [176], among others.

With the advancement of technology, the effectiveness of text detection is also getting better and better. However, text detection still faces some difficulties and challenges, summarized as follows.

- **The diversity of text:** Text usually comes in many different fonts, sizes, colors, etc. and even undergoes deformation, rotation, and other situations in some advertising campaigns. In addition, the background of the text is not static, and in real-life scenarios, the text may appear in backgrounds such as buildings and trees. These situations all pose challenges to text detection.
- **Multilingual:** Text detection may be designed for languages from multiple different countries, and there may be scenarios where multiple languages are used interchangeably. The language writing rules and recognition difficulty vary in different countries, which also poses certain challenges for text detection.
- **Text density and overlap:** Handwritten text is particularly prone to text density and overlap, which may result in individual characters being difficult to recognize. In addition, handwriting is prone to smudging marks, which may affect the accuracy of image text detection tasks.
- **Low contrast and noise:** In text image datasets, there may be some text images with low contrast, which can make it difficult for the text detector to recognize, thereby reducing the accuracy of the text detector. In addition, noise can also interfere with the text detector.

In summary, text detection faces various difficulties and challenges in real-life scenarios, and researchers need to choose appropriate technical means to continuously improve the accuracy and robustness of text detectors. Similarly, real-time requirements are also required for text detection.

4.5. Pedestrian detection

Pedestrian detection [177–179] is a key task in computer vision, aimed at determining whether an image contains pedestrians and detecting their positions. The success of pedestrian detection technology has provided assistance in many fields, such as autonomous driving [15], intelligent monitoring [180] and human–machine interaction [181], and so on. Pedestrian detection methods have gone through multiple stages of development, summarized as follows.

Traditional machine learning methods: Early pedestrian detection algorithms primarily relied on traditional machine learning techniques, which involved manually designing features and classifiers to achieve the desired detection outcomes. The commonly used feature description methods include Haar features [57], HOG [26] features, and sifi features [182], while classifiers include AdaBoost classifiers [58]

and SVM classifiers [183]. Combining feature description methods and classifiers can achieve pedestrian detection methods, each with its own advantages and disadvantages, which need to be selected according to specific situations. Common traditional pedestrian detectors include HOG detectors [26].

Deep learning-based methods: The proposal for Transformer has prompted researchers to focus on the role of attention and apply it to pedestrian detection tasks. In paper [184], attention mechanisms are utilized to enable the model to better extract pedestrian features. In addition, traditional pedestrian detection datasets typically require a large amount of manual annotation. In order to alleviate this problem, the paper [185] uses unsupervised learning methods to complete pedestrian detection tasks and uses methods such as Generative Adversarial Networks (GANs [166]) to train without data annotation.

Point cloud data-based methods: Pedestrian detection methods based on point cloud data typically use 3D sensors to obtain point cloud data and then analyze this data to detect pedestrians. Point cloud data has been a research hotspot in recent years, providing three-dimensional shape and structural information about objects. The paper [186] directly utilizes point cloud data for pedestrian detection and achieves good results.

The continuous progress of pedestrian detection cannot be separated from the development of technology, but it also faces some difficulties and challenges, summarized as follows.

- **Occlusion problem:** In practical life scenarios, pedestrians on the street are likely to be obstructed by pedestrians, vehicles, buildings, etc., which will result in only partial parts of the pedestrian's body, thereby increasing the difficulty of pedestrian detection. In addition, when there are many pedestrians on the street, congestion can occur, which also poses certain challenges to pedestrian detection.
- **Dynamic environment:** In real-life applications, pedestrian detectors are often needed to dynamically detect pedestrians, such as pedestrians in moving vehicles, moving pedestrians, etc. These dynamic factor detectors may not be able to accurately track.
- **Complex background:** Scenes in daily life are often extremely complex, with many objects that may resemble pedestrians, such as trees, sculptures, billboards, etc. These objects are interfering factors for the detector and are prone to false positives and missed detections.
- **Small pedestrians:** There are some difficult to detect small pedestrians in the pedestrian dataset, which occupy very few pixels. Pedestrian detectors are prone to losing their feature information when extracted, resulting in inefficient detection and thus reducing the performance of pedestrian detection.

In summary, there are still some difficulties in pedestrian detection that are difficult to solve, and researchers need to continuously propose new algorithms and technologies.

4.6. Other applications

There are many other applications for object detection besides those described above. For example, autonomous driving [14] realizes the detection of road vehicles and pedestrians, completes the driverless function, helps drivers reduce the burden of driving, and ensures traffic safety. Medical imaging detection [17] initially helps doctors review lesion images, reduces the burden on hospitals, and promotes the improvement of medical standards. In terms of environmental protection [187], object detection can be used to monitor water pollution, wildlife behavior, forest fires, etc., timely detect environmental problems, and solve them in advance, playing a role in environmental protection. In terms of entertainment [188], object detection can be used for game entertainment, recognizing and tracking the actions, gestures, etc. of characters in virtual games, helping users achieve interactive experiences and enjoyable physical and mental experiences. In agriculture, object detection can detect and identify whether crops have pathological changes, achieving precise agricultural production.

5. Challenges and prospects of object detection

In today's era, the development of object detection technology has become relatively mature and has been used in various scenarios in daily life. However, there are still some difficulties and challenges in object detection technology that need improvement. In this section, we will introduce some of the challenges faced by object detection technology and its future prospects.

5.1. Difficulties and challenges

Object detection plays a critical role in the advancement of deep learning and has seen significant progress with the evolution of deep learning technologies. Now, object detection methods are sufficiently advanced to cater to everyday requirements. Yet, when it comes to specialized, high-tech applications, object detection encounters numerous challenges that demand further enhancements. Here is an overview of these challenges.

Background interference: In specific life scenarios, the background of the object is intricate and complex. For example, in natural scenes, objects may be affected by factors such as weather, lighting, shadows, and occlusion, resulting in low discrimination between the object and the background, which brings detection difficulties to object detection technology. Present deep learning approaches can still struggle with complex backgrounds, particularly when the background and object features closely resemble each other. In addition, how to effectively combine contextual information for object detection remains a research hotspot and challenge.

Generalization ability: In commonly used datasets [51,53] for object detection, the categories and shapes of objects are diverse, and the same object may be classified into different categories in different detectors. In dynamic scenes, objects in the same category may exhibit different poses. This requires object detectors to have strong generalization ability, but existing object detection still has certain limitations when dealing with objects of different categories and shapes. And when facing objects of different shapes, existing algorithms are difficult to handle in extreme situations.

Multi scale detection: In real life, the scale of the object is often polymorphic, and the object may appear on different scales, with a very large range of changes in scale. Therefore, designing a detection algorithm that can perfectly adapt to multi-scale changes is very challenging, usually requiring methods such as multi-scale inputs. Although the existing pyramid structure [31,95] can handle multi-scale object detection problems, the performance of object detectors is relatively limited under the conditions of small object detection. Therefore, designing a system that can maintain consistency and high accuracy in detection results across different scales remains a challenge.

Occlusion problem: Object occlusion [3] in object detection technology is a common phenomenon. The occlusion problem includes complete occlusion and partial occlusion. Complete occlusion poses certain difficulties for object detection tasks, as the detector cannot fully extract the feature map information of the object, thereby affecting the detection accuracy of the detector. How to extract visible object information from detectors under partial occlusion is an important research direction and also a huge challenge.

Computational efficiency and real-time performance: Computational efficiency is often related to the computational complexity of object detection, which typically involves designing a large number of parameters, resulting in high computational complexity. Although some lightweight object detection [45,48] algorithms have been proposed, improving detection speed while ensuring accuracy remains a challenge. Real-time is essential in daily life scenarios, such as video surveillance and autonomous driving tasks. Therefore, how to improve the computational efficiency and real-time performance of the model remains a challenge for future object detection tasks.

Dataset bias: There are many commonly used datasets for object detection, and there is often some bias between different datasets, such as background, category, etc. With the development of deep learning technology, the problem of dataset bias is becoming increasingly apparent. How to reduce the dependence on the dataset during the training process is a research hotspot. Therefore, designing a detection algorithm that can generalize to different datasets and complex scenes is a challenge.

Video object detection: In video object detection [189] tasks, objects often change dynamically, and the background also changes accordingly. A video is composed of images with continuous temporal sequences, and the temporal nature of the video can lead to situations where adjacent image frames have continuous content and small object changes. This requires the detector to maintain the temporal and spatial consistency of the video sequence, as well as model lightweighting. Therefore, how to effectively utilize the temporal information of videos and reasonably solve complex background changes remains a challenge and difficulty in video object detection technology.

Small object detection: Small object detection [190] has always been a challenge in object detection technology. Small objects occupy very few pixels in the image, making it difficult to extract feature information. Moreover, the size of small objects is relatively small, and high accuracy is required for the localization of bounding boxes. Even small deviations may result in undetectable small objects. The enhancement of pyramid and multi-scale detection technologies has somewhat improved the detection accuracy of small objects. Despite these advancements, the accuracy levels for detecting small objects with existing detectors remain relatively low and require further enhancements. Accurately extracting information from small objects is essential, making the detection of small objects a significant area for future research.

In summary, object detection technology remains a field full of challenges and opportunities. Through continuous research and innovation, we believe that more advanced and practical object detectors will emerge in the future, providing more accurate and efficient object detection services for practical applications.

5.2. Future and prospects

The future development of object detection technology is full of infinite possibilities and development space. This section mainly discusses several possible directions for the future development of object detection technology, summarized as follows.

Multimodal fusion: In theory, object detection may see significant development in multimodal aspects. With the integration of computer vision and natural language processing, object detection systems will gradually integrate multi-source information such as images, texts, and speech to achieve more comprehensive and in-depth scene understanding. Multimodal object detection [191] will not only focus on recognizing the appearance of objects but also pay attention to the semantic and contextual information of objects, improving the system's perception ability for complex scenes. This trend will promote intelligent systems to respond more intelligently and flexibly to diverse information inputs in practical applications, expand the applicable fields of object detection, such as autonomous driving, smart homes, and medical assistance, and provide richer and more comprehensive technical support for future multimodal human–computer interaction.

3D object detection: 3D object detection [192] is crucial in machine learning, pattern recognition, and computer vision, aiming to precisely identify and locate objects in a three-dimensional space. 3D object detection is widely used in various fields, such as autonomous driving and robotics, and has great potential for development in the future. 3D object detectors have a strong dependence on sensors and are classified based on the type of sensor used. In the future, it may rely more on multi-sensor fusion technology to fuse various types of sensor data to obtain more accurate object information, thereby improving the robustness and accuracy of 3D object detection.

Self-supervised learning and semi-supervised learning: Object detection, a form of supervised learning, necessitates manual data annotation, which can be a time-intensive process. Self-supervised learning typically does not require manual annotation of data and can be used as a pre-trained model for object detection. In addition, self-supervised learning can learn low-level and high-level features of images and use contextual information to learn more robust information. These are beneficial for object detection tasks, and there is a certain prospect for combining self-supervised learning with object detection in the future. Semi-supervised learning typically uses limited data annotation, which can also address the problem of insufficient data annotation. If object detection could integrate self-supervised and semi-supervised learning techniques to fully leverage the information from unlabeled data and enhance detection performance, it would represent a significant breakthrough in the field of computer vision.

Large-scale models: In recent years, the development of large-scale models has been very rapid, playing a positive role in classical computer vision tasks and also greatly promoting the development of society. For object detection, large-scale models typically have more learnable parameters, enabling them to learn more complex features and better capture semantic and spatial information in images. In addition, the development of large-scale models has enabled classical object detection tasks to have faster inference speeds. After the training of the large-scale models is completed, further optimization processing can improve the inference speed, which is very important for real-time applications in real life and improves the speed of object detection. Although large-scale models have greatly improved in performance and speed, they typically require more computing resources and storage space to train and deploy. This is a huge challenge for devices with limited hardware resources, leading to an increase in training costs and making it difficult to achieve widespread adoption and application of large-scale models. Overall, the development of large-scale models is a crucial driving force for classical object detection tasks, while also facing challenges in terms of computational resources and deployment costs that need to be addressed in the future.

Diffusion models: In recent years, object detection has gradually emerged in the direction of diffusion models. DiffusionDet [193] successfully applied the diffusion model to object detection for the first time and demonstrated good performance on the COCO dataset. The success of DiffusionDet demonstrates the potential of diffusion models in object detection tasks, which requires further exploration by future researchers. With the continuous progress of diffusion model technology, the focus of attention in the future may be on how to make diffusion models more suitable for object detection tasks, how to improve diffusion model architecture, train optimization strategies, and enhance efficiency. The diffusion model, as a new direction in deep generative models, has high flexibility and scalability. As technology continues to advance and application scenarios expand, the use of diffusion models in object detection is expected to become more varied in the future.

Future applications: Object detection has many applications in daily life, and there will be more beneficial applications in the future. Autonomous driving, medical image analysis, environmental monitoring, and other fields will become increasingly mature in the future, becoming the important application scenarios in the field of object detection. However, different practical application scenarios require the use of different object detection algorithms to achieve. One-stage object detection algorithms usually have faster detection speed and are more suitable for applications that pursue real-time performance, such as environmental monitoring, intelligent transportation systems, etc. Two-stage object detection algorithms typically have higher accuracy and are more suitable for applications that pursue accuracy, such as autonomous driving and intelligent robots, etc. End-to-end object detection algorithms typically have stronger object localization

capabilities and are more suitable for applications that pursue fine-grained detection, such as medical image analysis and environmental perception, etc. With the development of computer vision technology and artificial intelligence, practical applications in the field of object detection will become more intelligent and user-friendly in the near future.

In summary, there are many development directions for object detection technology in the future, especially in the direction of diffusion models and multimodal fusion. With the joint efforts of a large number of researchers, future object detection technologies will become increasingly advanced and meet the high requirements of real life.

6. Conclusion

In this paper, the development and challenges of object detection are mainly introduced. Firstly, the development process of object detection is introduced, from early traditional algorithms to the evolution based on deep learning algorithms. Traditional algorithms such as the Viola Jones detector, although they had some effectiveness at the time, also had certain limitations. The object detection methods based on deep learning are mainly divided into two-stage detection algorithms and one-stage detection methods, such as the Faster R-CNN and YOLO series, which have improved accuracy and speed to a certain extent. Then, a detailed introduction is given to the commonly used datasets and evaluation metrics for object detection tasks, including VOC, COCO, ILSVRC, and OID. The evaluation metrics include precision (P), recall (R), average accuracy (AP), mean average accuracy (mAP), frames per second (FPS), and intersection over union (IoU). In addition, this paper discusses the application of object detection technology, highlighting its importance in real life, including face detection, defect detection, remote sensing image detection, text detection, pedestrian detection, etc. Finally, the paper discusses some challenges encountered by object detection technology and looks forward to the future development direction of object detection. Overall, object detection has developed relatively maturely and can basically meet the needs of real life. However, for high demand application scenarios such as autonomous driving, detectors still need to have strong real-time performance and accuracy. In the near future, object detection technology will continue to improve to meet these requirements.

CRediT authorship contribution statement

Zonghui Li: Writing – original draft, Visualization, Methodology, Investigation, Conceptualization. **Yongsheng Dong:** Writing – review & editing, Methodology, Investigation, Formal analysis, Conceptualization. **Longchao Shen:** Writing – review & editing, Methodology, Investigation, Conceptualization. **Yafeng Liu:** Writing – review & editing, Methodology, Investigation, Conceptualization. **Yuanhua Pei:** Writing – review & editing, Methodology, Investigation, Conceptualization. **Haotian Yang:** Writing – review & editing, Methodology, Investigation, Conceptualization. **Lintao Zheng:** Writing – review & editing, Methodology, Investigation, Conceptualization. **Jinwen Ma:** Writing – review & editing, Methodology, Investigation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

This work was supported by the Natural Science Foundation of Henan, China under Grant 232300421023.

References

[1] J. Wang, L. Jiang, H. Yu, Z. Feng, R. Castaño-Rosa, S.-j. Cao, Computer vision to advance the sensing and control of built environment towards occupant-centric sustainable development: A critical review, Renew. Sustain. Energy Rev. 192 (2024) 114165.
[2] P. Fraternali, F. Milani, R.N. Torres, N. Zangrando, Black-box error diagnosis in deep neural networks for computer vision: A survey of tools, Neural Comput. Appl. 35 (4) (2023) 3041–3062.
[3] Z. Zou, K. Chen, Z. Shi, Y. Guo, J. Ye, Object detection in 20 years: A survey, Proc. IEEE (2023).
[4] G. Cheng, X. Yuan, X. Yao, K. Yan, Q. Zeng, X. Xie, J. Han, Towards large-scale small object detection: Survey and benchmarks, IEEE Trans. Pattern Anal. Mach. Intell. (2023).
[5] D. Cazzato, C. Cimarelli, J.L. Sanchez-Lopez, H. Voos, M. Leo, A survey of computer vision methods for 2D object detection from unmanned aerial vehicles, J. Imaging 6 (8) (2020) 78.
[6] D. Vrontis, M. Christofi, V. Pereira, S. Tarba, A. Makrides, E. Trichina, Artificial intelligence, robotics, advanced technologies and human resource management: A systematic review, Int. J. Hum. Res. Manag. 33 (6) (2022) 1237–1266.
[7] M. Krenn, R. Pollice, S.Y. Guo, M. Aldeghi, A. Cervera-Lierta, P. Friederich, G. dos Passos Gomes, F. Häse, A. Jinich, A. Nigam, et al., On scientific understanding with artificial intelligence, Nat. Rev. Phys. 4 (12) (2022) 761–769.
[8] Y.K. Dwivedi, N. Pandey, W. Currie, A. Micu, Leveraging ChatGPT and other generative artificial intelligence (AI)-based applications in the hospitality and tourism industry: Practices, challenges and research agenda, Int. J. Contemp. Hosp. Manag. 36 (1) (2024) 1–12.
[9] A. Belhadi, V. Mani, S.S. Kamble, S.A.R. Khan, S. Verma, Artificial intelligence-driven innovation for enhancing supply chain resilience and performance under the effect of supply chain dynamism: An empirical investigation, Ann. Oper. Res. 333 (2) (2024) 627–652.
[10] H. Yaacob, F. Hossain, S. Shari, S.K. Khare, C.P. Ooi, U.R. Acharya, Application of artificial intelligence techniques for brain-computer interface in mental fatigue detection: A systematic review (2011–2022), IEEE Access (2023).
[11] E. Yurtsever, J. Lambert, A. Carballo, K. Takeda, A survey of autonomous driving: Common practices and emerging technologies, IEEE Access 8 (2020) 58443–58469.
[12] B.R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A.A. Al Sallab, S. Yogamani, P. Pérez, Deep reinforcement learning for autonomous driving: A survey, IEEE Trans. Intell. Transp. Syst. 23 (6) (2021) 4909–4926.
[13] W. Wang, L. Wang, C. Zhang, C. Liu, L. Sun, et al., Social interactions for autonomous driving: A review and perspectives, Found. Trends® Robot. 10 (3–4) (2022) 198–376.
[14] J. Zhao, W. Zhao, B. Deng, Z. Wang, F. Zhang, W. Zheng, W. Cao, J. Nan, Y. Lian, A.F. Burke, Autonomous driving system: A comprehensive survey, Expert Syst. Appl. (2023) 122836.
[15] B. Kaltenhäuser, K. Werdich, F. Dandl, K. Bogenberger, Market development of autonomous driving in Germany, Transp. Res. Part A: Policy Pract. 132 (2020) 882–910.
[16] X. Tian, T. Jiang, L. Yun, Y. Mao, H. Yang, Y. Wang, Y. Wang, H. Zhao, Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving, Adv. Neural Inf. Process. Syst. 36 (2024).
[17] M. Baumgartner, P.F. Jäger, F. Isensee, K.H. Maier-Hein, nnDetection: A self-configuring method for medical object detection, Med. Image Comput. Comput. Assist. Intervent. (2021) 530–539.
[18] Y. Shou, T. Meng, W. Ai, C. Xie, H. Liu, Y. Wang, Object detection in medical images based on hierarchical transformer and mask mechanism, Comput. Intell. Neurosci. 2022 (2022).
[19] R. Qureshi, M.G. Ragab, S.J. Abdulkader, A. Alqushaib, E.H. Sumiea, H. Alhussian, et al., A comprehensive systematic review of YOLO for medical object detection (2018 to 2023), 2023, Authorea Preprints.
[20] X. Zeng, Y. Liu, J. Zhang, Y. Guo, Medical object detector jointly driven by knowledge and data, Neural Netw. 172 (2024) 106084.
[21] Q. Wang, F. Liu, R. Zou, Y. Wang, C. Zheng, Z. Tian, S. Du, W. Zeng, Enhancing medical image object detection with collaborative multi-agent deep Q-networks and multi-scale representation, EURASIP J. Adv. Signal Process. 2023 (1) (2023) 132.
[22] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, D. Terzopoulos, Image segmentation using deep learning: A survey, IEEE Trans. Pattern Anal. Mach. Intell. 44 (7) (2021) 3523–3542.
[23] L.V. Jospin, H. Laga, F. Boussaid, W. Buntine, M. Bennamoun, Hands-on Bayesian neural networks—A tutorial for deep learning users, IEEE Comput. Intell. Mag. 17 (2) (2022) 29–48.

[24] S. Minaee, A. Abdolrashidi, H. Su, M. Bennamoun, D. Zhang, Biometrics recognition using deep learning: A survey, Artif. Intell. Rev. (2023) 1–49.

[25] M. Raissi, Forward–backward stochastic neural networks: Deep learning of high-dimensional partial differential equations, in: Peter Carr Gedenkschrift: Research Advances in Mathematical Finance, 2024, pp. 637–655.

[26] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, 2005, pp. 886–893.

[27] P. Ott, M. Everingham, Shared parts for deformable part-based models, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 1513–1520.

[28] C.-H. Lee, C.-R. Lin, M.-S. Chen, Sliding-window filtering: An efficient algorithm for incremental mining, in: Proceedings of the Tenth International Conference on Information and Knowledge Management, 2001, pp. 263–270.

[29] V. Braverman, R. Ostrovsky, C. Zaniolo, Optimal sampling from sliding windows, in: Proceedings of the Twenty-eighth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, 2009, pp. 147–156.

[30] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al., Recent advances in convolutional neural networks, Pattern Recognit. 77 (2018) 354–377.

[31] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.

[32] R. Girshick, Fast R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.

[33] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, Adv. Neural Inf. Process. Syst. 28 (2015).

[34] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.

[35] X. Lu, B. Li, Y. Yue, Q. Li, J. Yan, Grid R-CNN, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 7363–7372.

[36] Z. Cai, N. Vasconcelos, Cascade R-CNN: Delving into high quality object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6154–6162.

[37] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, D. Lin, Libra R-CNN: Towards balanced learning for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 821–830.

[38] H. Zhang, H. Chang, B. Ma, N. Wang, X. Chen, Dynamic R-CNN: Towards high quality object detection via dynamic training, in: Proceedings of the European Conference on Computer Vision, 2020, pp. 260–275.

[39] P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, M. Tomizuka, L. Li, Z. Yuan, C. Wang, et al., Sparse R-CNN: End-to-end object detection with learnable proposals, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021, pp. 14454–14463.

[40] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: Single shot multibox detector, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 21–37.

[41] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.

[42] J. Redmon, A. Farhadi, YOLO9000: Better, faster, stronger, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7263–7271.

[43] A. Farhadi, J. Redmon, Yolov3: An incremental improvement, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 1804, 2018, pp. 1–6.

[44] A. Bochkovskiy, C.-Y. Wang, H.-Y.M. Liao, Yolov4: Optimal speed and accuracy of object detection, 2020, arXiv preprint arXiv:2004.10934.

[45] J. Glenn, YOLOv5 release v6.1, 2022, https://github.com/ultralytics/yolov5/releases/tag/v6.1.

[46] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, et al., YOLOv6: A single-stage object detection framework for industrial applications, 2022, arXiv preprint arXiv:2209.02976.

[47] C.-Y. Wang, A. Bochkovskiy, H.-Y.M. Liao, YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2023, pp. 7464–7475.

[48] J. Glenn, YOLOv8 release v8.1.0, 2023, https://github.com/ultralytics/ultralytics/tree/v8.1.0.

[49] C.-Y. Wang, I.-H. Yeh, H.-Y.M. Liao, YOLOv9: Learning what you want to learn using programmable gradient information, 2024, arXiv preprint arXiv:2402.13616.

[50] L. Shen, Y. Dong, Y. Pei, H. Yang, L. Zheng, J. Ma, One-dimensional feature supervision network for object detection, in: International Conference on Intelligent Computing, 2023, pp. 147–156.

[51] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, Int. J. Comput. Vis. 88 (2010) 303–338.

[52] M. Everingham, S.A. Eslami, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge: A retrospective, Int. J. Comput. Vis. 111 (2015) 98–136.

[53] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft COCO: Common objects in context, in: Proceedings of the European Conference on Computer Vision, 2014, pp. 740–755.

[54] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (2015) 211–252.

[55] I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, A. Veit, et al., Openimages: A public dataset for large-scale multi-label and multi-class image classification, vol. 2(3), 2017, p. 18, Dataset available from https://github.com/openimages.

[56] P. Viola, M.J. Jones, Robust real-time face detection, Int. J. Comput. Vis. 57 (2004) 137–154.

[57] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, 2001, p. I.

[58] Y. Freund, R. Schapire, N. Abe, A short introduction to boosting, J.-Jap. Soc. Artif. Intell. 14 (771–780) (1999) 1612.

[59] T. Ephraim, T. Himmelman, K. Siddiqi, Real-time Viola-Jones face detection in a web browser, in: Canadian Conference on Computer and Robot Vision, 2009, pp. 321–328.

[60] A.W.Y. Wai, S.M. Tahir, Y.C. Chang, GPU acceleration of real time Viola-Jones face detection, in: IEEE International Conference on Control System, Computing and Engineering, 2015, pp. 183–188.

[61] S. Suma, S. Raga, Real time face recognition of human faces by using LBPH and Viola Jones algorithm, Int. J. Sci. Res. Comput. Sci. Eng. 6 (5) (2018) 6–10.

[62] H. Joseph, B.K. Rajan, Real time drowsiness detection using Viola Jones & KLT, in: International Conference on Smart Electronics and Communication, 2020, pp. 583–588.

[63] T.H. Obaida, A.S. Jamil, N.F. Hassan, Real-time face detection in digital video-based on Viola-Jones supported by convolutional neural networks, Int. J. Electr. Comput. Eng. 12 (3) (2022).

[64] V.K. Gurrala, S. Talasila, P. Madhuri, S.N. Varma, L. Puneeth, P. Koppireddi, Enhancing safety and security: Face tracking and detection in dehazed video frames using KLT and Viola-Jones algorithms, Int. J. Saf. Secur. Eng. 13 (4) (2023).

[65] O.M. Demidenko, N.A. Aksionova, A.V. Varuyeu, Identification of students' faces in a video stream using the Viola-Jones method, in: International Conference on Information, Control, and Communication Technologies, 2022, pp. 1–5.

[66] B. Edwiranda, B.C. Purba, Y. Bandung, Design and implementation of real-time object tracking system based on Viola-Jones algorithm for supporting video conference, in: International Conference on Telecommunication Systems, Services, and Applications, 2018, pp. 1–6.

[67] T. Paul, U.A. Shammi, M.U. Ahmed, R. Rahman, S. Kobashi, M.A.R. Ahad, A study on face detection using viola-jones algorithm in various backgrounds, angles and distances, Int. J. Biomed. Soft Comput. Hum. Sci. 23 (1) (2018) 27–36.

[68] J.R. Uijlings, K.E. Van De Sande, T. Gevers, A.W. Smeulders, Selective search for object recognition, Int. J. Comput. Vis. 104 (2013) 154–171.

[69] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012).

[70] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.

[71] X. Ren, D. Ramanan, Histograms of sparse codes for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3246–3253.

[72] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (1995) 273–297.

[73] H. Wang, D. Hu, Comparison of SVM and LS-SVM for regression, in: International Conference on Neural Networks and Brain, vol. 1, 2005, pp. 279–283.

[74] S. Huang, N. Cai, P.P. Pacheco, S. Narrandes, Y. Wang, W. Xu, Applications of support vector machine (SVM) learning in cancer genomics, Cancer Genom. Proteomics 15 (1) (2018) 41–51.

[75] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1904–1916.

[76] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, 2015, pp. 448–456.

[77] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2980–2988.

[78] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[79] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125.

[80] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, A.C. Berg, DSSD: Deconvolutional single shot detector, 2017, arXiv preprint arXiv:1701.06659.

[81] A. Shrivastava, R. Sukthankar, J. Malik, A. Gupta, Beyond skip connections: Top-down modulation for object detection, 2016, arXiv preprint arXiv:1612.06851.

[82] H. Law, J. Deng, Cornernet: Detecting objects as paired keypoints, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 734–750.

[83] A. Newell, J. Deng, Pixels to graphs by associative embedding, Adv. Neural Inf. Process. Syst. 30 (2017).

[84] A. Newell, Z. Huang, J. Deng, Associative embedding: End-to-end learning for joint detection and grouping, Adv. Neural Inf. Process. Syst. 30 (2017).

[85] A. Newell, K. Yang, J. Deng, Stacked hourglass networks for human pose estimation, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 483–499.

[86] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, Q. Tian, Centernet: Keypoint triplets for object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 6569–6578.

[87] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, Q. Tian, CenterNet++ for object detection, IEEE Trans. Pattern Anal. Mach. Intell. (2023).

[88] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE International Conference on Computer Vision, 2021, pp. 10012–10022.

[89] Z. Tian, C. Shen, H. Chen, T. He, FCOS: Fully convolutional one-stage object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 9627–9636.

[90] I. Ulku, E. Akagündüz, A survey on deep learning-based architectures for semantic segmentation on 2d images, Appl. Artif. Intell. 36 (1) (2022) 2032924.

[91] H. Thisanke, C. Deshan, K. Chamith, S. Seneviratne, R. Vidanaarachchi, D. Herath, Semantic segmentation using vision transformers: A survey, Eng. Appl. Artif. Intell. 126 (2023) 106669.

[92] S. Hao, Y. Zhou, Y. Guo, A brief survey on semantic segmentation with deep learning, Neurocomputing 406 (2020) 302–321.

[93] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1492–1500.

[94] C.-Y. Wang, H.-Y.M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, I.-H. Yeh, CSPNet: A new backbone that can enhance learning capability of CNN, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 390–391.

[95] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8759–8768.

[96] H. Zhang, M. Cisse, Y.N. Dauphin, D. Lopez-Paz, Mixup: Beyond empirical risk minimization, 2017, arXiv preprint arXiv:1710.09412.

[97] S. Yun, D. Han, S.J. Oh, S. Chun, J. Choe, Y. Yoo, Cutmix: Regularization strategy to train strong classifiers with localizable features, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 6023–6032.

[98] Z. Yao, Y. Cao, S. Zheng, G. Huang, S. Lin, Cross-iteration batch normalization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021, pp. 12331–12340.

[99] J. Xu, M. Li, Z. Zhu, Automatic data augmentation for 3D medical image segmentation, Med. Image Comput. Comput. Assist. Intervent. (2020) 378–387.

[100] R. Raileanu, M. Goldstein, D. Yarats, I. Kostrikov, R. Fergus, Automatic data augmentation for generalization in reinforcement learning, Adv. Neural Inf. Process. Syst. 34 (2021) 5402–5415.

[101] Y. Li, G. Hu, Y. Wang, T. Hospedales, N.M. Robertson, Y. Yang, Differentiable automatic data augmentation, in: Proceedings of the European Conference on Computer Vision, 2020, pp. 580–595.

[102] Z. Ge, S. Liu, F. Wang, Z. Li, J. Sun, YOLOX: Exceeding yolo series in 2021, 2021, arXiv preprint arXiv:2107.08430.

[103] Z. Ge, S. Liu, Z. Li, O. Yoshie, J. Sun, OTA: Optimal transport assignment for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021, pp. 303–312.

[104] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, J. Sun, RepVGG: Making vgg-style convnets great again, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021, pp. 13733–13742.

[105] L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, K. Ma, Be your own teacher: Improve the performance of convolutional neural networks via self distillation, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 3713–3722.

[106] C. Li, L. Li, Y. Geng, H. Jiang, M. Cheng, B. Zhang, Z. Ke, X. Xu, X. Chu, Yolov6 v3. 0: A full-scale reloading, 2023, arXiv preprint arXiv:2301.05586.

[107] C.-Y. Wang, H.-Y.M. Liao, I.-H. Yeh, Designing network design strategies through gradient path analysis, 2022, arXiv preprint arXiv:2211.04800.

[108] C.-Y. Wang, A. Bochkovskiy, H.-Y.M. Liao, Scaled-YOLOv4: Scaling cross stage partial network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021, pp. 13029–13038.

[109] P. Dollár, M. Singh, R. Girshick, Fast and accurate model scaling, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021, pp. 924–932.

[110] M. Zand, A. Etemad, M. Greenspan, ObjectBox: From centers to boxes for anchor-free object detection, in: Proceedings of the European Conference on Computer Vision, 2022, pp. 390–406.

[111] Y. Jiang, Z. Tan, J. Wang, X. Sun, M. Lin, H. Li, GiraffeDet: A heavy-neck paradigm for object detection, 2022, arXiv preprint arXiv:2202.04256.

[112] X. Xu, Y. Jiang, W. Chen, Y. Huang, Y. Zhang, X. Sun, DAMO-YOLO: A report on real-time object detection design, 2022, arXiv preprint arXiv:2211.15444.

[113] Z. Sun, M. Lin, X. Sun, Z. Tan, H. Li, R. Jin, MAE-Det: Revisiting maximum entropy principle in zero-shot nas for efficient object detection, 2021, arXiv preprint arXiv:2111.13336.

[114] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, 2015, arXiv preprint arXiv:1503.02531.

[115] H. Zhang, Y. Wang, F. Dayoub, N. Sunderhauf, Varifocalnet: An iou-aware dense object detector, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021, pp. 8514–8523.

[116] Y. Quan, D. Zhang, L. Zhang, J. Tang, Centralized feature pyramid for object detection, 2022, arXiv preprint arXiv:2210.02093.

[117] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).

[118] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, 2020, arXiv preprint arXiv:2010.11929.

[119] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: Proceedings of the European Conference on Computer Vision, 2020, pp. 213–229.

[120] J. Beal, E. Kim, E. Tzeng, D.H. Park, A. Zhai, D. Kislyuk, Toward transformer-based object detection, 2020, arXiv preprint arXiv:2012.09958.

[121] W. Wang, J. Dai, Z. Chen, Z. Huang, Z. Li, X. Zhu, X. Hu, T. Lu, L. Lu, H. Li, et al., Internimage: Exploring large-scale vision foundation models with deformable convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2023, pp. 14408–14419.

[122] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, A. Torralba, Scene parsing through ade20k dataset, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 633–641.

[123] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, et al., A survey on vision transformer, IEEE Trans. Pattern Anal. Mach. Intell. 45 (1) (2022) 87–110.

[124] K.S. Kalyan, A. Rajasekharan, S. Sangeetha, Ammus: A survey of transformer-based pretrained models in natural language processing, 2021, arXiv preprint arXiv:2108.05542.

[125] Y. Liu, Y. Zhang, Y. Wang, F. Hou, J. Yuan, J. Tian, Y. Zhang, Z. Shi, J. Fan, Z. He, A survey of visual transformers, IEEE Trans. Neural Netw. Learn. Syst. (2023).

[126] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, L. Sun, Transformers in time series: A survey, 2022, arXiv preprint arXiv:2202.07125.

[127] W. Fedus, B. Zoph, N. Shazeer, Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, J. Mach. Learn. Res. 23 (1) (2022) 5232–5270.

[128] X. Zhu, H. Hu, S. Lin, J. Dai, Deformable convnets v2: More deformable, better results, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 9308–9316.

[129] F. Chollet, Xception: Deep learning with depthwise separable convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1251–1258.

[130] S. Woo, J. Park, J.-Y. Lee, I.S. Kweon, CBAM: Convolutional block attention module, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 3–19.

[131] S. Deng, Z. Liang, L. Sun, K. Jia, Vista: Boosting 3d object detection via dual cross-view spatial attention, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 8448–8457.

[132] M.-H. Guo, C.-Z. Lu, Q. Hou, Z. Liu, M.-M. Cheng, S.-M. Hu, Segnext: Rethinking convolutional attention design for semantic segmentation, Adv. Neural Inf. Process. Syst. 35 (2022) 1140–1156.

[133] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, 2015, arXiv preprint arXiv:1511.07122.

[134] C. Wang, W. He, Y. Nie, J. Guo, C. Liu, Y. Wang, K. Han, Gold-YOLO: Efficient object detector via gather-and-distribute mechanism, Adv. Neural Inf. Process. Syst. 36 (2024).

[135] M. Tan, R. Pang, Q.V. Le, Efficientdet: Scalable and efficient object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 10781–10790.

[136] G. Ghiasi, T.-Y. Lin, Q.V. Le, NAS-FPN: Learning scalable feature pyramid architecture for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 7036–7045.

[137] M. Hu, Y. Li, L. Fang, S. Wang, A2-FPN: Attention aggregation based feature pyramid network for instance segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021, pp. 15343–15352.

[138] Y. Luo, X. Cao, J. Zhang, J. Guo, H. Shen, T. Wang, Q. Feng, CE-FPN: Enhancing channel information for object detection, Multimedia Tools Appl. 81 (21) (2022) 30685–30704.
[139] K. Tian, Y. Jiang, Q. Diao, C. Lin, L. Wang, Z. Yuan, Designing bert for convolutional networks: Sparse and hierarchical masked modeling, 2023, arXiv preprint arXiv:2301.03580.
[140] K. Sun, B. Xiao, D. Liu, J. Wang, Deep high-resolution representation learning for human pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5693–5703.
[141] A. Kumar, A. Kaur, M. Kumar, Face detection techniques: A review, Artif. Intell. Rev. 52 (2019) 927–948.
[142] S. Hangaragi, T. Singh, N. Neelima, Face detection and recognition using face mesh and deep neural network, Procedia Comput. Sci. 218 (2023) 741–749.
[143] P. Melzi, R. Tolosana, R. Vera-Rodriguez, M. Kim, C. Rathgeb, X. Liu, I. DeAndres-Tame, A. Morales, J. Fierrez, J. Ortega-Garcia, et al., FRCSyn challenge at WACV 2024: Face recognition challenge in the era of synthetic data, in: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, 2024, pp. 892–901.
[144] K. Zhang, Z. Zhang, Z. Li, Y. Qiao, Joint face detection and alignment using multitask cascaded convolutional networks, IEEE Signal Process. Lett. 23 (10) (2016) 1499–1503.
[145] M. Wang, W. Deng, Deep face recognition: A survey, Neurocomputing 429 (2021) 215–244.
[146] J.N. Kolf, F. Boutros, J. Elliesen, M. Theuerkauf, N. Damer, M. Alansari, O.A. Hay, S. Alansari, S. Javed, N. Werghi, et al., Efar 2023: Efficient face recognition competition, 2023, arXiv preprint arXiv:2308.04168.
[147] M. Huber, A.T. Luu, P. Terhörst, N. Damer, Efficient explainable face verification based on similarity score argument backpropagation, in: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, 2024, pp. 4736–4745.
[148] D. Mery, B. Morris, On black-box explanation for face verification, in: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, 2022, pp. 3418–3427.
[149] J. Hartbrich, F. Weidner, C. Kunert, A. Raake, W. Broll, S. Arévalo Arboleda, Eye and face tracking in VR: Avatar embodiment and enfacement with realistic and cartoon avatars, in: Proceedings of the International Conference on Mobile and Ubiquitous Multimedia, 2023, pp. 270–278.
[150] S. Ranganatha, T.G. MT, S. Shivashankara, P. Ravi, A pragamatic approach for real time face tracking, Int. J. Intell. Syst. Appl. Eng. 12 (9s) (2024) 205–214.
[151] Y. Chen, Y. Ding, F. Zhao, E. Zhang, Z. Wu, L. Shao, Surface defect detection methods for industrial products: A review, Appl. Sci. 11 (16) (2021) 7657.
[152] A. Saberironaghi, J. Ren, M. El-Gindy, Defect detection methods for industrial products using deep learning techniques: A review, Algorithms 16 (2) (2023) 95.
[153] J. Tang, Z. Wang, H. Zhang, H. Li, P. Wu, N. Zeng, A lightweight surface defect detection framework combined with dual-domain attention mechanism, Expert Syst. Appl. 238 (2024) 121726.
[154] S.U. Lee, S.Y. Chung, R.H. Park, A comparative performance study of several global thresholding techniques for segmentation, Comput. Vis. Graph. Image Process. 52 (2) (1990) 171–190.
[155] P. Roy, S. Dutta, N. Dey, G. Dey, S. Chakraborty, R. Ray, Adaptive thresholding: A comparative study, in: International Conference on Control, Instrumentation, Communication and Computational Technologies, 2014, pp. 1182–1186.
[156] Y. Wan, L. Yao, B. Xu, Automatic segmentation of fiber cross sections by dual thresholding, J. Eng. Fibers Fabrics 7 (1) (2012) 155892501200700113.
[157] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9 (1) (1979) 62–66.
[158] R.M. Haralick, K. Shanmugam, I.H. Dinstein, Textural features for image classification, IEEE Trans. Syst. Man Cybern. (6) (1973) 610–621.
[159] A. Satpathy, X. Jiang, H.-L. Eng, LBP-based edge-texture features for object recognition, IEEE Trans. Image Process. 23 (5) (2014) 1953–1964.
[160] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, Med. Image Comput. Comput.-Assist. Intervent. (2015) 234–241.
[161] A. Shafique, G. Cao, Z. Khan, M. Asad, M. Aslam, Deep learning-based change detection in remote sensing images: A review, Remote Sens. 14 (4) (2022) 871.
[162] E. Duraklı, E. Aptoula, Domain generalized object detection for remote sensing images, in: Signal Processing and Communications Applications Conference, 2023, pp. 1–4.
[163] A.S. Sagar, Y. Chen, Y. Xie, H.S. Kim, MSA R-CNN: A comprehensive approach to remote sensing object detection and scene understanding, Expert Syst. Appl. 241 (2024) 122788.
[164] S. Grossberg, Recurrent neural networks, Scholarpedia 8 (2) (2013) 1888.
[165] L.S.-T. Memory, Long short-term memory, Neural Comput. 9 (8) (2010) 1735–1780.
[166] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Adv. Neural Inf. Process. Syst. 27 (2014).
[167] Q. Ye, D. Doermann, Text detection and recognition in imagery: A survey, IEEE Trans. Pattern Anal. Mach. Intell. 37 (7) (2014) 1480–1500.
[168] Y. Baek, B. Lee, D. Han, S. Yun, H. Lee, Character region awareness for text detection, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 9365–9374.
[169] X. Hu, P.-Y. Chen, T.-Y. Ho, Radar: Robust ai-text detection via adversarial learning, Adv. Neural Inf. Process. Syst. 36 (2024).
[170] T. Ghazal, Convolutional neural network based intelligent handwritten document recognition, Comput. Mater. Continua 70 (3) (2022) 4563–4581.
[171] M.S.H. Onim, H. Nyeem, K. Roy, M. Hasan, A. Ishmam, M.A.H. Akif, T.B. Ovi, Blpnet: A new dnn model and bengali ocr engine for automatic licence plate recognition, Array 15 (2022) 100244.
[172] M. Donoser, H. Bischof, Efficient maximally stable extremal region (MSER) tracking, in: IEEE Computer Society conference on Computer Vision and Pattern Recognition, vol. 1, 2006, pp. 553–560.
[173] B. Epshtein, E. Ofek, Y. Wexler, Detecting text in natural scenes with stroke width transform, in: IEEE Computer Society conference on Computer Vision and Pattern Recognition, 2010, pp. 2963–2970.
[174] S. Na, L. Xumin, G. Yong, Research on k-means clustering algorithm: An improved k-means clustering algorithm, in: International Symposium on Intelligent Information Technology and Security Informatics, 2010, pp. 63–67.
[175] F. Murtagh, P. Contreras, Algorithms for hierarchical clustering: An overview, Wiley Interdisc. Rev.: Data Min. Knowl. Discov. 2 (1) (2012) 86–97.
[176] U. Von Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (2007) 395–416.
[177] P. Dollar, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: An evaluation of the state of the art, IEEE Trans. Pattern Anal. Mach. Intell. 34 (4) (2011) 743–761.
[178] X. Huang, Z. Ge, Z. Jie, O. Yoshie, Nms by representative region: Towards crowded pedestrian detection by proposal pairing, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 10750–10759.
[179] S. Mallick, S. Ghosal, A. Balakrishnan, J. Deshmukh, Safety monitoring for pedestrian detection in adverse conditions, in: International Conference on Runtime Verification, 2023, pp. 389–399.
[180] K. Li, Z. Li, X. Jia, L. Liu, M. Chen, A domain adversarial graph convolutional network for intelligent monitoring of tool wear in machine tools, Comput. Ind. Eng. 187 (2024) 109795.
[181] J. Peng, A. Kimmig, D. Wang, Z. Niu, X. Tao, J. Ovtcharova, Intention recognition-based human–machine interaction for mixed flow assembly, J. Manuf. Syst. 72 (2024) 229–244.
[182] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2004) 91–110.
[183] Y. Yang, J. Li, Y. Yang, The research of the fast SVM classifier method, in: International Computer Conference on Wavelet Active Media Technology and Information Processing, 2015, pp. 121–124.
[184] J. Yuan, P. Barmpoutis, T. Stathaki, Effectiveness of vision transformer for fast and accurate single-stage pedestrian detection, Adv. Neural Inf. Process. Syst. 35 (2022) 27427–27440.
[185] C. Lyu, P. Heyer, B. Goossens, W. Philips, An unsupervised transfer learning framework for visible-thermal pedestrian detection, Sensors 22 (12) (2022) 4416.
[186] A.H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, O. Beijbom, Pointpillars: Fast encoders for object detection from point clouds, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 12697–12705.
[187] T. Sasilatha, D.H.M.S. Amala, R.S. Santhammal, Č. Lačnjevac, G. Singh, Deep learning-based underwater metal object detection using input image data and corrosion protection of mild steel used in underwater study: A case study: Part a: Deep learning-based underwater metal object detection using input image data, Mater. Protect. 63 (1) (2022) 5–14.
[188] Z. Mahmood, T. Ali, S. Khattak, L. Hasan, S.U. Khan, Automatic player detection and identification for sports entertainment applications, Pattern Anal. Appl. 18 (2015) 971–982.
[189] L. Jiao, R. Zhang, F. Liu, S. Yang, B. Hou, L. Li, X. Tang, New generation deep learning for video object detection: A survey, IEEE Trans. Neural Netw. Learn. Syst. 33 (8) (2021) 3195–3215.
[190] Y. Liu, P. Sun, N. Wergeles, Y. Shang, A survey and performance evaluation of deep learning methods for small object detection, Expert Syst. Appl. 172 (2021) 114602.
[191] Y.-T. Chen, J. Shi, Z. Ye, C. Mertz, D. Ramanan, S. Kong, Multimodal object detection via probabilistic ensembling, in: Proceedings of the European Conference on Computer Vision, 2022, pp. 139–158.
[192] R. Qian, X. Lai, X. Li, 3D object detection for autonomous driving: A survey, Pattern Recognit. 130 (2022) 108796.
[193] S. Chen, P. Sun, Y. Song, P. Luo, Diffusiondet: Diffusion model for object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2023, pp. 19830–19843.

Zonghui Li is currently pursuing M.S. degree with the School of Information Engineering, Henan University of Science and Technology, China. His current research interests include deep learning and computer vision.

Yongsheng Dong received his Ph.D. degree in applied mathematics from Peking University in 2012. He was a postdoctoral research fellow with the Center for Optical Imagery Analysis and Learning, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China from 2013 to 2016. From 2016 to 2017, he was a visiting research fellow at the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He is currently a full professor with the School of Information Engineering, Henan University of Science and Technology, China. His current research interests include pattern recognition, machine learning, and computer vision.

He has authored and co-authored over 60 papers at famous journals and conferences, including IEEE TIP, IEEE TNNLS, IEEE TCYB, IEEE TCSVT, IEEE TGRS and IEEE TIE. He has served as a reviewer for over 40 international prestigious journals and conferences, such as IEEE TPAMI, IEEE TIP, IEEE TKDE, IEEE TNNLS, IEEE TCYB, IEEE TIE, IEEE TSP, IEEE TCSVT, IEEE TMM, IEEE TCDS, IEEE TEICI, IEEE TGRS and ACM TIST. He has also served as a program committee member for more than 10 international conferences. He is an editorial board member of Neurocomputing.

Longchao Shen received his M.S. degree from Henan University of Science and Technology in 2023. His current research interests include deep learning and computer vision.

Yafeng Liu is currently pursuing M.S. degree with the School of Information Engineering, Henan University of Science and Technology, China. His current research interests include deep learning and computer vision.

Yuanhua Pei received his M.S. degree from Henan University of Science and Technology in 2023. He currently works in the Information Engineering Department of Luoyang Railway Information Engineering School. His current research interests include deep learning and computer vision. In recent years, he has written and co-authored over 10 papers in the above-mentioned fields.

Haotian Yang received his M.S. degree from Henan University of Science and Technology in 2023. His current research interests include deep learning and computer vision.

Lintao Zheng received his Ph.D. degree in Computer Science from Zhejiang University in 2012. He is currently an associate professor with the school of Information Engineering, Henan University of Science and Technology, Luoyang, China. His current research interests include image processing and computer vision.

Jinwen Ma received the M.S. degree in applied mathematics from Xi'an Jiaotong University in 1988 and the Ph.D. degree in probability theory and statistics from Nankai University in 1992. From July 1992 to November 1999, he was a lecturer or associate professor at the Department of Mathematics, Shantou University. From December 1999, he became a full professor at the Institute of Mathematics, Shantou University. From September 2001, he has joined the Department of Information Science at the School of Mathematical Sciences, Peking University, where he is currently a full professor and a Ph.D. student tutor. During 1995 and 2003, he also visited several times at the Department of Computer Science and Engineering, the Chinese University of Hong Kong as a Research Associate or Fellow. He worked as Research Scientist at Amari Research Unit, RIKEN Brain Science Institute, Japan from September 2005 to August 2006. From September 2011 to February 2012, he further visited the Department of System Medicine and Biological Engineering, Research Center of Methodist Hospital System, Houston, USA, as a Scientist. He has published over 200 academic papers on neural networks, pattern recognition, computer vision, bioinformatics, and information theory.