Systematic Review
A Review of Object Detection Algorithms based on Deep Learning

ABSTRACT

With the continuous development of deep learning, object detection algorithms based on deep learning have made significant progress in the field of computer vision, widely applied in areas such as autonomous driving, industrial inspection, agriculture, transportation, and medicine. Traditional object detection algorithms face issues such as low detection efficiency and poor robustness. However, deep learning-based object detection algorithms significantly enhance detection accuracy and generalization by learning low-level and high-level image features. This article first introduces traditional object detection algorithms and their existing problems, then elaborates on the main processes, innovations, advantages, disadvantages, and experimental results on datasets of deep learning-based object detection algorithms. It focuses on the development of Two-Stage and One-Stage object detection algorithms, and provides an outlook on the future development of object detection algorithms, discussing challenges such as the coordination of detection speed and accuracy, difficulties in detecting small objects, real-time detection tasks, and multi-modal fusion applications, and proposes possible future directions.

Keywords: Object Detection; Computer Vision; Deep Learning;

1. INTRODUCTION

Object detection is one of the fundamental tasks in computer vision, aiming to identify objects in images or videos and determine their positions and categories. Tasks in computer vision such as object tracking, image segmentation, and face recognition are all built upon object detection[1]. Computer vision can reduce the consumption of human resources, making it of significant practical importance, hence object detection has become a research hotspot in recent years. With the rapid development of deep learning technologies like Convolutional Neural Networks (CNN), deep learning-based object detection algorithms have shown outstanding performance in various applications such as autonomous driving[2], industrial inspection[3], agriculture[4], transportation[5], and medicine[6]. This article extensively surveys domestic and international object detection methods, firstly introducing early object detection algorithms and pointing out their shortcomings, then detailing the Two-Stage object detection algorithm based on candidate windows and the One-Stage object detection algorithm based on regression. It analyzes the strengths and weaknesses of the relevant algorithms, and finally concludes and looks ahead to the future of object detection algorithms.

2. TRADITIONAL OBJECT DETECTION ALGORITHMS

Early object detection algorithms were mostly based on manually designed filter features. The basic approach[7], as shown in Figure 1, involves preprocessing the input image, constructing candidate box regions, extracting candidate boxes from the input image using a sliding window

approach, extracting features, and then using a classifier for classification. Non-maximum suppression (NMS) can be used to merge candidate boxes, eliminating overlapping or redundant candidate boxes, and outputting the final results.

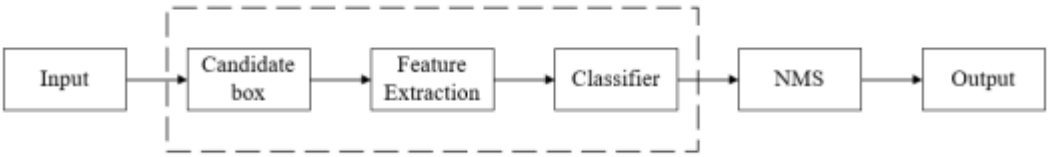

Figure 1: Basic process of traditional object detection methods.

In 2001, Viola at al.[8] proposed a real-time face detection algorithm based on Haar-like features and Adaboost classifier, introducing the Integral Image for rapid computation of Haar-like features, making it the first object detection algorithm suitable for real-time applications. In 2005, Dalal at al.[9] put forward the Histograms of Oriented Gradients (HOG) features for pedestrian detection, extracting features through HOG of image regions and combining them with SVM for detection. In 2008, Felzenszwalb at al.[10] introduced the Deformable Part Models (DPM), utilizing sliding windows for target localization, HOG components for feature extraction, and SVM for classification, exhibiting excellent detection performance. In 2013, Uijlings at al.[11] proposed the Selective Search algorithm, segmenting images into multiple regions using a hierarchical segmentation approach, then merging adjacent regions to generate candidate regions, laying the groundwork for later deep learning-based object detection algorithms like R-CNN.

Although traditional object detection algorithms achieved some success in the early stages, they have significant drawbacks. Firstly, traditional algorithms typically use a sliding window approach to generate candidate boxes, leading to exponentially increasing computational requirements with the growth of image pixels, thus demanding higher computational capabilities. Secondly, traditional object detection algorithms rely on manually designed features, which lack robustness against the diversity of targets and result in low detection efficiency and accuracy. The limitations of traditional object detection algorithms in terms of computational efficiency, robustness, and generalization capabilities have restricted their widespread application. With the development of deep learning technology, deep learning-based object detection algorithms have gradually overcome these shortcomings and become the mainstream direction of current research.

3. TWO-STAGE DETECTION ALGORITHM BASED ON CANDIDATE WINDOWS.

The two-stage algorithm processes input images by first determining possible target regions through the generation of candidate boxes (Region Proposals). The methods for generating candidate boxes mainly include Selective Search and Anchor-based methods. Selective Search segments the image initially, creating smaller regions that are then merged to form larger candidate boxes, filtering out the candidate boxes most similar to the target objects. The Anchor-based method generates fixed-sized anchor boxes with specific aspect ratios on the image, using convolutional neural networks to classify and regress each anchor box, obtaining confidence scores and positional offsets for each anchor box, and finally selecting candidate boxes with higher confidence based on the confidence scores. Although multi-step processing can improve the accuracy of target localization and detection precision, the complex detection process can also impact the detection speed of the algorithm. Figure 2 illustrates the development process of object detection algorithms.

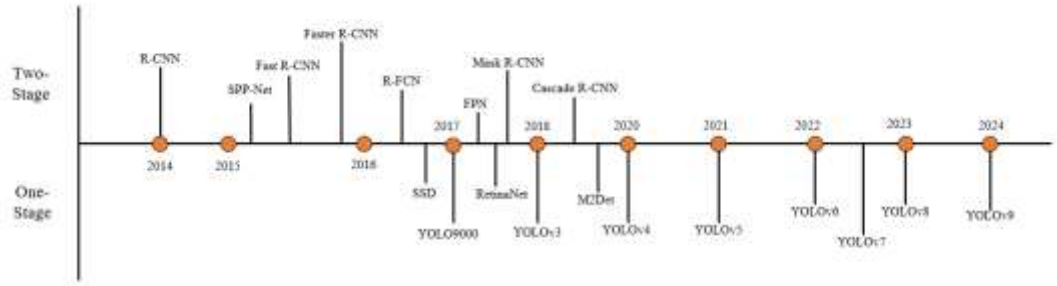

Figure 2: Development history of deep learning-based object detection algorithms.

3.1 R-CNN

In 2014, Girshick et al. [12] proposed the Region-based Convolutional Neural Network (R-CNN). R-CNN utilizes the structure of the AlexNet network and replaces the sliding window method with selective search to generate candidate regions. It introduces Convolutional Neural Networks (CNN) into the object detection task, uses Support Vector Machines (SVM) for classification and box regression, and employs a linear regression model to correct the positions of candidate boxes, significantly improving detection accuracy. R-CNN laid the foundation for subsequent two-stage object detection algorithms, as shown in the framework process in Figure 3. However, it also has shortcomings. Each candidate region needs to input the CNN independently for feature extraction, leading to large computational requirements and slow training speeds. Additionally, storing features for each candidate region individually consumes a significant amount of storage space. Furthermore, the cropping and scaling operations of candidate regions may alter the shape of the image, potentially disrupting the original information and affecting the accuracy of subsequent operations.

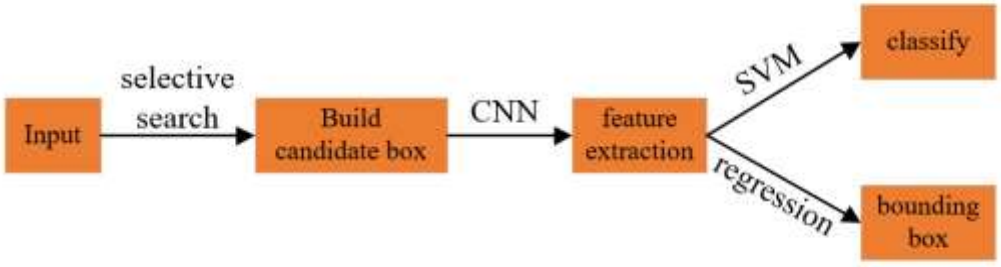

Figure 3: Architecture diagram of the R-CNN network.

3.2 SPP-NET

The Spatial Pyramid Pooling Network (SPP-Net) [13] was proposed to address the slow processing speed issue in feature extraction of R-CNN, which required convolution of all candidate regions individually. The workflow of SPP-Net is as follows: first, the input image undergoes convolution and pooling layers for feature extraction; second, the feature map generates a fixed-length feature vector through the spatial pyramid pooling layer; finally, the feature vector is input into subsequent classifiers for target classification. SPP-Net optimizes the convolution process by performing a single convolution operation on the entire image, speeding up the detection process. By introducing the spatial pyramid pooling layer, SPP-Net solves the issue of variable input image sizes, enabling the network to handle images of any size. The addition of the SPP structure between the convolutional layer and the fully connected layer ensures that the output feature map is of a fixed size, avoiding shape changes during

image normalization and improving detection accuracy. The comparison of the processes between R-CNN and SPP-Net is shown in Figure 4.

Figure 4: Comparison of the process between SPP-Net and R-CNN algorithms.

3.3 FAST R-CNN

Fast R-CNN[14], proposed by Girshick et al. in 2015, is an improved object detection algorithm based on R-CNN and SPP-Net. It aims to address issues such as redundant feature extraction and multi-stage pipeline training in the R-CNN algorithm, while enhancing detection speed and accuracy. Fast R-CNN simplifies the SPP structure into the ROI pooling layer, pooling candidate regions of different sizes into fixed-size feature vectors. At the end of the network, it includes two parallel branches: one for classification using the Softmax function and the other for bounding box regression. By simultaneously performing target classification and precise localization in a single network and introducing a multi-task loss function, Fast R-CNN reduces computational complexity, improving training and detection efficiency. On the VOC07 dataset, Fast R-CNN increased mAP from 58.5% (R-CNN) to 70.0% and achieved detection speeds over 200 times faster than R-CNN. By sharing convolutional calculations and utilizing the ROI pooling layer, Fast R-CNN significantly reduces redundant feature extraction computations. However, Fast R-CNN still uses selective search to generate candidate regions, which is a slow process and a bottleneck in the entire detection process. The framework process of Fast R-CNN is illustrated in Figure 5.

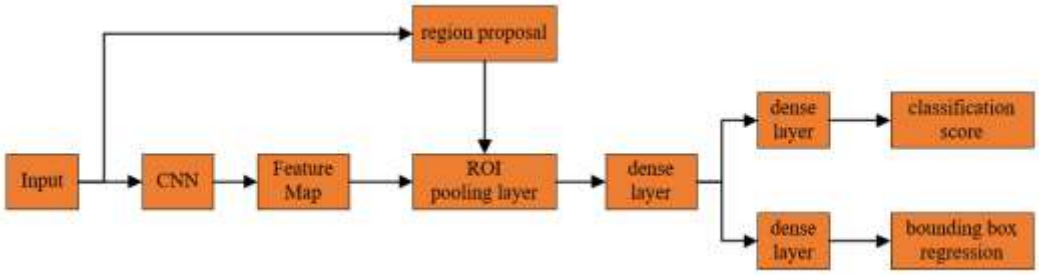

Figure 5: Network model of the Fast R-CNN algorithm.

3.4 FASTER R-CNN

Faster R-CNN[15], proposed by Ren et al. in 2015, is an advanced version of the object detection algorithm, building upon R-CNN and Fast R-CNN. This algorithm introduces the Region Proposal Network (RPN) to generate candidate regions, enabling true end-to-end training and significantly improving detection speed and accuracy. Faster R-CNN incorporates the RPN module to merge candidate region generation, feature extraction, classification, and regression, achieving end-to-end training. RPN utilizes a sliding window mechanism and anchor box generation to enhance detection speed and accuracy. By sharing convolutional layers with the detection network Fast R-CNN, RPN avoids redundant computations, enhancing computational efficiency. Faster R-CNN leverages GPU for computation, resulting

in a 10-fold increase in detection speed compared to Fast R-CNN. It integrates candidate region generation, feature extraction, classification, and regression into a single network structure, facilitating end-to-end training. However, the adverse effects of the ROI pooling layer on network translation invariance lead to decreased localization accuracy and poorer detection performance for small objects. The network structure of Faster R-CNN is depicted in the figure 6.

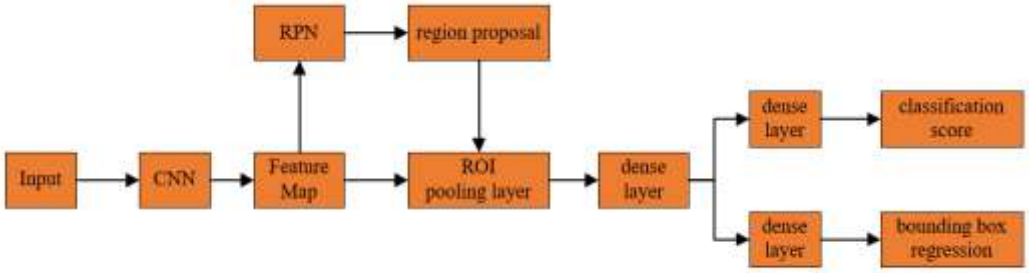

Figure 6: Network architecture diagram of Faster R-CNN.

3.5 OTHER TWO-STAGE ALGORITHMS

R-FCN (Region-based Fully Convolutional Networks) [16] is an improved algorithm proposed in 2016 to address the heavy computational burden in feature extraction of Faster R-CNN. By replacing the fully connected layers after the ROI pooling layer with convolutional layers, R-FCN significantly enhances detection speed. This algorithm utilizes ResNet as the backbone network, replacing the VGG network in Faster R-CNN, thereby improving the network's feature extraction capabilities and classification performance.

Mask R-CNN [17], an optimization algorithm proposed in 2017 for Faster R-CNN, enhances the backbone network by adding a Feature Pyramid Network (FPN) on top of ResNet. It consists of Faster R-CNN for object detection and FPN for semantic segmentation. By replacing the ROI pooling part with RoIAlign, it addresses the region mismatch issue and introduces an independent Mask branch for instance segmentation, improving segmentation accuracy.

Cascade R-CNN [18] is a cascade-structured object detector designed to address overfitting issues that occur during network training at low IOU thresholds. By incrementally increasing the IOU threshold across multiple stages, it aims to improve detection accuracy and resolve overfitting problems at low IOU thresholds. The proposal of Chained Cascade Network[19], an end-to-end learning of more than two cascaded classifiers and DCNNs for general-purpose object detection, extensions of Cascade R-CNN, and applications in synchronous object detection and instance segmentation[20] achieved success in the 2018 COCO detection challenge.

3.6 ALGORITHM COMPARISON

The two-stage object detection algorithm divides the detection process into two main steps: finding candidate regions containing the target objects in the image and performing classification and regression operations on these candidate regions to obtain the final detection results. In terms of candidate region extraction, early two-stage object detection algorithms such as R-CNN, SPP-Net, and Fast R-CNN primarily used selective search algorithms. Although these methods achieved good detection results, they were slow and less practical. Faster R-CNN significantly improved the speed and accuracy of candidate region extraction by introducing an RPN-based approach. In terms of feature utilization, new algorithms propose the use of the FPN structure, combining high-level semantic information with low-level spatial information to improve the detection of small objects.

Representative algorithms include Mask R-CNN, among others. While these improvements have significantly increased detection accuracy, the complexity of the network models has led to a decrease in the speed of training and detection, affecting real-time performance. The performance comparison of two-stage object detection algorithms is shown in Table 1, with "—" indicating no relevant data in the original literature.

Table.1 Performance comparison of Two-stage target detection algorithms

algorithms	backbone	FPS	mAP/%		
			VOC2007	VOC2012	COCO
R-CNN	AlexNet	0.03	58.5	—	—
	VGG16	0.50	66.0	53.3	—
SPPNet	ZF-5	2.00	59.2	—	—
Fast R-CNN	VGG16	7.00	70.0	68.4	19.7
Faster R-CNN	VGG16	7.00	73.2	70.4	21.9
	AlexNet101	5.00	76.4	73.8	34.9
R-FCN	AlexNet101	5.80	79.5	77.6	29.9
Mask R-CNN	AlexNet101+FPN	5.00	—	—	39.8
Libra R-CNN	AlexNet101+FPN	—	—	—	43.0
Grid R-CNN	AlexNet101+FPN	—	—	—	43.2

4. REGRESSION-BASED ONE-STAGE OBJECT DETECTION ALGORITHM

Single-stage detection algorithms can output detection results, including classification and bounding box prediction, in one network pass. These algorithms excel in detection speed and are well-suited for use on mobile devices. Additionally, this structure allows enough flexibility to add various algorithm modules to meet different detection requirements.

4.1 OVERFEAT

The OverFeat[21] algorithm was proposed in 2013 and is one of the pioneers of single-stage object detection. Its main principle is to perform image classification on multi-scale regions of the image using a sliding window approach and train a regressor on the same convolutional layer to predict the position of bounding boxes. The algorithm combines classification, localization, and detection by improving AlexNet, introducing a novel pooling method called offsetmax-pooling, predicting at different scales, and accumulating predictions to obtain bounding boxes. The algorithm provides both accurate and fast models. In the accurate model, the classification error rate is 14.18%, while in the fast model, it is 16.39%. When using a combination of 7 accurate models, the classification error rate decreases to 13.6%.

4.2 THE YOLO SERIES ALGORITHMS

The YOLO (You Only Look Once) [22] algorithm was first introduced in 2016. Inspired by the GoogLeNet [23] structure, it has 24 convolutional layers for extracting image features and 2 fully connected layers for predicting bounding boxes and class probabilities. Except for the last layer, which uses a linear activation function, the rest of the layers utilize Leaky ReLU activation function [24]. YOLO, as a single-stage object detection algorithm, can simultaneously output classification and bounding box predictions in one network pass, integrating classification, localization, and detection in a single network. This approach helps avoid

misclassifying the background as the target, achieving faster detection speed and lower detection error rates.

In order to improve the accuracy of YOLO detection, YOLOv2 [25] was proposed in 2017. Inspired by VGG (Visual Geometry Group) [26] and Network-in-Network [27], it utilizes the Darknet-19 backbone network to enhance feature extraction capabilities and detection speed. Batch normalization (BN) is introduced to improve convergence, resulting in a 2.4% increase in mAP. By fine-tuning the classification model with high-resolution images, mAP is improved by 3.7%. Removing fully connected layers, using anchor box convolution to predict bounding boxes, has increased recall rates. Employing the K-means clustering method to extract prior box scales enhances generalization. Additionally, training with multi-scale images significantly improves detection speed and accuracy, allowing real-time prediction of up to 9000 categories of objects.

YOLOv3 [28] was introduced in 2018, further enhancing the detection capability for small objects. YOLOv3 replaces the backbone network with Darknet-53, enhancing feature extraction capabilities. It achieves multi-scale prediction through feature fusion with residual networks, improving the performance of small object detection. The use of the logistic algorithm's binary cross-entropy loss function for classification improves the accuracy of multi-object classification. Object scores and class confidences are calculated using the Sigmoid function to achieve fine-grained object categorization.

YOLOv4 [29], proposed in 2020, further optimized YOLOv3. The Neck section utilizes SPP (Spatial Pyramid Pooling) [30] and PAN (Path Aggregation Network) [31] modules for feature fusion, enhancing detection accuracy. The use of CutMix and Mosaic data augmentation, along with DropBlock regularization, reduces overfitting and improves generalization. The introduction of CIoU (Complete Intersection over Union) [32] localization loss enhances the accuracy of bounding box localization. YOLOv4 shows significant improvements in both detection speed and accuracy, making it suitable for applications with high real-time requirements.

he YOLOv5 series includes S, M, L, and X versions, suitable for different real-time application scenarios. Compared to YOLOv4, YOLOv5 models are smaller, faster in computation speed, and have lower memory usage. YOLOv5 uses Mosaic data augmentation at the input end, reduces the number of GPUs used, calculates high-scoring anchors using adaptive anchor box computation, and uniformly resizes images to a suitable size using adaptive scaling. The Backbone section employs the NewCSP-Darknet53 model, performs slicing operations with the Focus module, utilizes CSP1 _ X for feature fusion, and obtains fixed-length outputs with the SPP module. The Neck section incorporates FPN (Feature Pyramid Network) and PAN (Path Aggregation Network) modules, enhancing multi-scale feature expression and strong localization information. CSP2 _ X further strengthens the feature fusion from the previous step.

YOLOX [33], proposed in 2021, draws inspiration from advanced anchor-free object detectors like CornerNet [34] and CenterNet [35] to further optimize YOLOv3 and YOLOv5. The addition of a decoupled head structure improves accuracy and convergence speed. By adopting anchor-free structures and sample matching strategies, parameter count is reduced, enhancing the detection performance for small objects. Utilizing data augmentation techniques such as Mosaic, MixUp[36], SimOTA (Optimal Transport Assignment)[37], and regularization techniques enhances the model's robustness. YOLOX balances accuracy and speed, making it suitable for various application scenarios.

YOLOv6[38], dedicated to industrial applications, optimizes the network structure to fully utilize hardware computing power. YOLOv6 improves the network structure by re-optimizing the backbone network based on hardware-aware neural network design principles, leveraging hardware computing power with strong representational capabilities and higher parallelism. On the Neck side, it designs Rep-PAN (Representation-PAN)[39] and introduces RepBlock to ensure efficient inference and better multi-scale feature fusion capabilities. On the Head side, it retains YOLOX's decoupled head design but improves the decoupled head structure with

Hybrid Channels strategy for streamlined design, maintaining detection accuracy while reducing latency. To further enhance detection performance, it introduces self-distillation strategies in regression and classification tasks, utilizes SimOTA label assignment strategy and SIoU (Scylla-IoU) [40] bounding box regression loss function to reduce regression freedom, accelerate network convergence speed, and improve regression accuracy.

YOLOv7[41], focusing on model architecture optimization and training process optimization, addresses the issues of model reparameterization and dynamic label assignment in object detection. It proposes planned reparameterization models and a guided label assignment method from coarse-grained to fine-grained for target detection. The YOLOv7 algorithm introduces a cascaded model scaling strategy to generate models of different sizes, reducing parameter count and computational load, enabling real-time object detection. When trained on large datasets, it achieves higher accuracy in detection with overall performance improvements. However, its network architecture is relatively complex, requiring significant computational resources for training and testing, and it shows poorer detection performance in small objects and dense scenes.

YOLOv8 supports multiple visual tasks, integrating algorithms for pose estimation, object detection, image classification, and instance segmentation. It combines numerous state-of-the-art (SOTA) technologies and is highly scalable, supporting other YOLO versions and algorithms beyond YOLO. YOLOv8's backbone network and Neck section use the C2f structure with a richer gradient flow, combining high-level features with contextual information. Different channel numbers are set for models of different scales, enhancing the overall model performance. The detection head uses a decoupled head structure to separate detection and classification, independently handling visual tasks. The loss function employs binary cross-entropy for classification loss, DFL (Distribution Focal Loss), and CIoU Loss for regression loss, improving object detection performance, especially for detecting smaller objects.

YOLOv9[42] is a new generation advanced object detection system introduced by research teams from institutions such as the Academia Sinica in Taipei and the Taipei Tech University. It is an improvement over its predecessor versions, aiming to address information loss in deep learning and enhance the model's performance across various tasks. YOLOv9 introduces PGI, a through-assisted reversible branch that generates reliable gradient information to update network parameters, resolving information loss in deep networks to improve training efficiency and model performance. YOLOv9 designs a new lightweight network architecture called GELAN, based on gradient path planning. By optimizing computational blocks and network depth, it enhances the model's parameter utilization and inference speed.

4.3 SSD SERIES ALGORITHM

Integrating the concept of regression into object detection has provided a new improvement approach for object detection algorithms. Liu et al.[43] proposed the SSD (Single Shot Multibox Detector) algorithm, combining the idea of extracting multiple candidate regions as regions of interest (ROI) from Faster R-CNN with the regression concept from YOLO. This integration partially addresses YOLO's shortcomings in recognizing small objects and insensitivity to scale.

The SSD model suffers from the issue of repeatedly detecting the same object, leading to increased computational complexity. To address this problem, Jeong et al. [44] proposed the RSSD algorithm, which replaces the VGGNet backbone network with ResNet to achieve weight sharing in the classifier network, thereby improving training speed. Fu et al.[45] introduced the DSSD algorithm, based on the ResNet101 backbone network, implementing upsampling through deconvolution to enhance the detection accuracy of small objects. However, deepening the backbone network can slow down the training detection speed. Li et al.[46] combined the idea of Feature Pyramid Network (FPN) to propose the FSSD algorithm, connecting features from different scales and layers in the feature fusion module to generate a new feature pyramid for predicting the final detection results. In 2019, Shen et al. [47]

integrated the concepts of SSD and DenseNet to introduce the DSOD algorithm, which reduces the parameter count without requiring additional data or pre-trained models. However, the hierarchical dense connections achieved through deep supervision may lead to feature redundancy and increased computational complexity. Also in the same year, Yi et al. [48] presented the ASSD algorithm, establishing feature relationships in the feature map space to learn useful regions based on global relationship information and suppress irrelevant information, providing a reliable basis for object detection.

4.4 ALGORITHM COMPARISON

The structure of the object detection network determines the initial advantages of the detection algorithm. For example, two-stage detection algorithms have the characteristics of accurate localization and high detection accuracy, while one-stage detection algorithms have faster detection speeds. However, according to a unified evaluation metric, both types of detection algorithms are addressing structural deficiencies and aiming to improve towards higher accuracy and faster detection speeds. Table 2 summarizes the performance of classic detection algorithms based on deep learning under a unified evaluation metric.

Table.2 Performance comparison of One-stage target detection algorithms

algorithm	backbone	FPS	mAP/%		
			VOC2007	VOC2012	COCO
OverFeat	AlexNet	—	24.3	—	—
YOLO	VGG16	45.0	63.4	57.9	—
YOLOv2	DarkNet19	40.0	—	—	21.6
YOLOv3	DarkNet53	78.0	—	—	33.0
YOL0v4	CSPDarkNet53	66.0	—	—	43.5
YOLOv5	Focus+CSP	140.0	—	—	—
YOLOx	ModifiedCSPv5	57.8	—	—	51.2
SSD300	VGG16	46.0	74.3	72.4	23.2
SSD512	VGG16	19.0	76.8	74.9	26.8
R-SSD300	VGG16	35.0	78.5	76.4	—
DSOD300	DS/64-192-48-1	17.4	77.7	76.3	29.3
F-SSD300	VGGNet	65.8	82.7	82.0	27.1

5. CONCLUSION

This article first introduces traditional object detection algorithms. Next, it introduces deep learning-based object detection algorithms based on regression and candidate box classifications, focusing on key representative algorithms, analyzing network structures, and comparing their strengths and weaknesses. Lastly, it discusses the directions for future improvements in practical applications of object detection algorithms, as follows:

a. Research on feature networks more suitable for object detection tasks. The design of feature networks should consider the differences between classification and detection modules. One of the future research trends is how to design feature networks that meet the actual task requirements.

b. Multi-modal object detection. Data fusion is an important trend in achieving object detection applications. The complementary nature of multimodal data can enhance the model's robustness and address issues like uneven lighting.

c. High-quality datasets. A good dataset can better train excellent models, so efforts should be made to acquire high-quality, diverse datasets for training.
d. Model lightweighting. With the development of computer hardware and software and the increasing demand for functionality, model lightweighting is one of the major trends in the future. Developing more efficient lightweight network structures, such as Google's MobileNet framework [49], followed by 2nd and 3rd generations [[49-51], ShuffleNet [52], ShuffleNet v2 [53], can achieve more efficient feature extraction with limited computing power.
e. Object detection based on GAN. Due to factors such as data acquisition costs, time constraints, and special scenarios, data scarcity may occur. Considering the use of GAN series networks, using a portion of real-world data to generate virtual data can expand the dataset, covering a wider range of scenarios to improve detection performance.

REFERENCES

1. Li, Meian, et al. "Research on object detection algorithm based on deep learning." Journal of Physics: Conference Series. Vol. 1995. No. 1. IOP Publishing, 2021.
2. Wang, Huijuan, et al. "A review of 3D object detection based on autonomous driving."The Visual Computer (2024): 1-19.
3. Zhang, Haigang, et al. "Adaptive visual detection of industrial product defects."PeerJ Computer Science 9 (2023): e1264.
4. Ariza-Sentís, Mar, et al. "Object detection and tracking in Precision Farming: a systematic review." Computers and Electronics in Agriculture 219 (2024): 108757.
5. He, Shouhui, et al. "Automatic recognition of traffic signs based on visual inspection." IEEE Access 9 (2021): 43253-43261.
6. Kaur, Amrita, et al. "A survey on deep learning approaches to medical images and a systematic look up into real-time object detection." Archives of Computational Methods in Engineering (2021): 1-41.
7. Lowe, David G. "Distinctive image features from scale-invariant keypoints." International journal of computer vision 60 (2004): 91-110.
8. Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001. Vol. 1. Ieee, 2001.
9. Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). Vol. 1. Ieee, 2005.
10. Felzenszwalb, Pedro F., et al. "Object detection with discriminatively trained part-based models." IEEE transactions on pattern analysis and machine intelligence 32.9 (2009): 1627-1645.
11. Uijlings, Jasper RR, et al. "Selective search for object recognition." International journal of computer vision 104 (2013): 154-171.
12. Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.
13. He, Kaiming, et al. "Spatial pyramid pooling in deep convolutional networks for visual recognition." IEEE transactions on pattern analysis and machine intelligence 37.9 (2015): 1904-1916.
14. Girshick, Ross. "Fast r-cnn." Proceedings of the IEEE international conference on computer vision. 2015.
15. Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems 28 (2015).
16. Dai, Jifeng, et al. "R-fcn: Object detection via region-based fully convolutional networks." Advances in neural information processing systems 29 (2016).

17. He, Kaiming, et al. "Mask r-cnn." Proceedings of the IEEE international conference on computer vision. 2017.
18. Cai, Zhaowei, and Nuno Vasconcelos. "Cascade r-cnn: Delving into high quality object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
19. Ouyang, Wanli, et al. "Chained cascade network for object detection." Proceedings of the IEEE International Conference on Computer Vision. 2017.
20. Chen, Kai, et al. "Hybrid task cascade for instance segmentation." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.
21. Sermanet, Pierre, et al. "Overfeat: Integrated recognition, localization and detection using convolutional networks." arxiv preprint arxiv:1312.6229 (2013).
22. Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
23. Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
24. Maas, Andrew L., Awni Y. Hannun, and Andrew Y. Ng. "Rectifier nonlinearities improve neural network acoustic models." Proc. icml. Vol. 30. No. 1. 2013.
25. Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
26. Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arxiv preprint arxiv:1409.1556 (2014).
27. Lin, Min, Qiang Chen, and Shuicheng Yan. "Network in network." arxiv preprint arxiv:1312.4400 (2013).
28. Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." arxiv preprint arxiv:1804.02767 (2018).
29. Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection." arxiv preprint arxiv:2004.10934 (2020).
30. He, Kaiming, et al. "Spatial pyramid pooling in deep convolutional networks for visual recognition." IEEE transactions on pattern analysis and machine intelligence 37.9 (2015): 1904-1916.
31. Liu, Shu, et al. "Path aggregation network for instance segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
32. Zheng, Zhaohui, et al. "Distance-IoU loss: Faster and better learning for bounding box regression." Proceedings of the AAAI conference on artificial intelligence. Vol. 34. No. 07. 2020.
33. Ge, Zheng, et al. "Yolox: Exceeding yolo series in 2021." arxiv preprint arxiv:2107.08430 (2021).
34. Law, Hei, and Jia Deng. "Cornernet: Detecting objects as paired keypoints." Proceedings of the European conference on computer vision (ECCV). 2018.
35. Duan, Kaiwen, et al. "Centernet: Keypoint triplets for object detection." Proceedings of the IEEE/CVF international conference on computer vision. 2019.
36. Zhang, Hongyi, et al. "mixup: Beyond empirical risk minimization." arxiv preprint arxiv:1710.09412 (2017).
37. Ge, Zheng, et al. "Ota: Optimal transport assignment for object detection." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.
38. Li, Chuyi, et al. "YOLOv6: A single-stage object detection framework for industrial applications." arxiv preprint arxiv:2209.02976 (2022).
39. Ding, **aohan, et al. "Repvgg: Making vgg-style convnets great again." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.
40. Gevorgyan, Zhora. "SIoU loss: More powerful learning for bounding box regression." arxiv preprint arxiv:2205.12740 (2022).
41. Wang, Chien-Yao, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors."

Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023.
42. Wang, Chien-Yao, I-Hau Yeh, and Hong-Yuan Mark Liao. "YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information." arxiv preprint arxiv:2402.13616 (2024).
43. Liu, Wei, et al. "Ssd: Single shot multibox detector." Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016.
44. Jeong, Jisoo, Hyo** Park, and Nojun Kwak. "Enhancement of SSD by concatenating feature maps for object detection." arxiv preprint arxiv:1705.09587 (2017).
45. Fu, Cheng-Yang, et al. "Dssd: Deconvolutional single shot detector." arxiv preprint arxiv:1701.06659 (2017).
46. Li, Zuoxin, Lu Yang, and Fuqiang Zhou. "FSSD: feature fusion single shot multibox detector." arxiv preprint arxiv:1712.00960 (2017).
47. Shen, Zhiqiang, et al. "Object detection from scratch with deep supervision." IEEE transactions on pattern analysis and machine intelligence 42.2 (2019): 398-412.
48. Yi, **gru, Pengxiang Wu, and Dimitris N. Metaxas. "ASSD: Attentive single shot multibox detector." Computer Vision and Image Understanding 189 (2019): 102827.
49. Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arxiv preprint arxiv:1704.04861 (2017).
50. Howard, Andrew, et al. "Searching for mobilenetv3." Proceedings of the IEEE/CVF international conference on computer vision. 2019.
51. Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
52. Zhang, **angyu, et al. "Shufflenet: An extremely efficient convolutional neural network for mobile devices." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
53. Ma, Ningning, et al. "Shufflenet v2: Practical guidelines for efficient cnn architecture design." Proceedings of the European conference on computer vision (ECCV). 2018.