Object Detection in Recent Years: An Overview

Jun Xiao
Guilin University of Electronic Technology School of
Computer Science and Information Security Guilin, China
22032303110@mails.guet.edu.cn

Jinlong Chen*
Guilin University of Electronic Technology School of
Computer Science and Information Security Guilin, China
Jinlong.chen@guet.edu.cn

Yi Ning
Guilin University of Electronic Technology School of
Continuing Education Guilin, China
296106092@qq.com

Yun Jiang
Guilin University of Electronic Technology School of
Computer Science and Information Security Guilin, China
jy973202065@gmail.com

ABSTRACT

Object detection is one of the important tasks in the field of computer vision, aiming at automatically identifying specific targets in images or videos and localizing their positions. This paper presents some milestone algorithms, performance metrics, and classical datasets for object detection in chronological order. First, the difference between one-stage and two-stage object detection algorithm frameworks is discussed, focusing on the strengths and weaknesses of each of the object detection algorithms, such as RCNN, YOLO, and others, as the pioneering work of the object detection niche. Then, precision, recall, and F1 value are introduced as commonly used performance metrics to measure the accuracy and completeness of the algorithms. In addition, IoU is introduced as a measure of the degree of overlap between predicted boxes and real labels, and mAP is explained as a metric for comprehensively evaluating the performance of algorithms. Finally, some publicly available datasets such as PASCAL VOC, ILSVRC, and MS-COCO, which play an important role in advancing the development of object detection algorithms, are discussed. By studying the experimental results of different algorithms and datasets, it can help to improve the performance and application of object detection algorithms.

CCS CONCEPTS

• **General and reference** → Document types; Surveys and overviews.

KEYWORDS

Object Detection, Deep Learning, Computer Vision, Technology Revolution

ACM Reference Format:
Jun Xiao, Jinlong Chen*, Yi Ning, and Yun Jiang. 2024. Object Detection in Recent Years: An Overview. In *The 3rd International Conference on Computer, Artificial Intelligence and Control Engineering (CAICE 2024), January 26–28, 2024, Xi' an, China.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3672758.3672862

1 INTRODUCTION

Object detection is a crucial task in the field of computer vision, aiming to accurately locate and identify objects of interest in images. It plays a vital role in various domains such as face recognition, video surveillance, autonomous driving, industrial inspection, and more.

With the rise of deep learning, particularly the extensive research and application of Convolutional Neural Networks (CNNs) [1], deep learning-based object detection has opened up a new research frontier. An increasing number of researchers have delved into this emerging field, continually proposing groundbreaking theories and algorithms. Simultaneously, object detection serves as the foundation for other computer vision tasks like instance segmentation and object tracking. In the realm of deep learning, object detection is a popular and rapidly evolving direction, evident from the sharp increase in publications related to object detection over the last decade, as illustrated in Figure 1.

Therefore, a comprehensive review and analysis of deep learning-based object detection are meaningful and necessary, providing readers with an in-depth understanding and awareness of the latest advancements in this field.

Compared to traditional object detection algorithms such as the Viola-Jones Detector [2], HOG Detector [3], and DPM [4], deep learning-based object detection algorithms have achieved significant breakthroughs in both accuracy and speed. This progress enables the widespread application of object detection in more complex real-world scenarios. This paper aims to comprehensively introduce the development of deep learning-based object detection, covering algorithmic advancements, datasets, and evaluation metrics. Following a chronological order, it will provide a thorough overview of the past, present, and future of deep learning-based object detection, offering a multidimensional comparison of the strengths and weaknesses of different algorithms at different stages.

The objective of this paper is to provide readers with a comprehensive understanding of the development and key algorithms in deep learning-based object detection, along with insights into the latest advancements in this field.

2 ALGORITHMS

The following passage provides a comprehensive review of milestone algorithms in object detection. This chapter will systematically introduce two-stage and one-stage detection algorithms over time, discussing the strengths and weaknesses of each algorithm

CAICE 2024, January 26–28, 2024, Xi' an, China

ACM ISBN 979-8-4007-1694-2/24/01
https://doi.org/10.1145/3672758.3672862

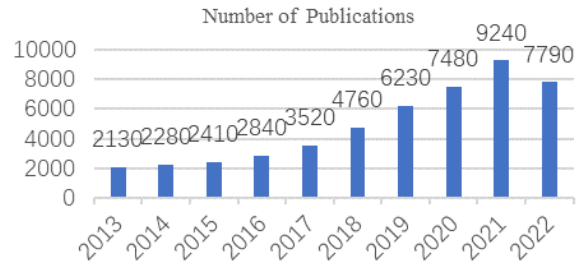


Figure 1: Number of published articles on object detection in the last 10 years. (Data from Google Scholar advanced search on the keyword "object detection")

and their impact on the development of their respective fields. Figure 2 shows the publication timeline of classic milestone algorithms in object detection over the past decade.

2.1 Two-Stage Detection Algorithms

Two-stage object detection algorithms are a commonly used framework that accomplishes object detection through two stages: candidate box generation and object classification. It is a coarse-to-fine process. In the candidate box generation stage, the algorithm generates a series of candidate boxes that may contain target objects. These boxes of different sizes and aspect ratios are often generated using specific algorithms such as sliding windows [5] and selective search [6]. In the object classification and localization stage, the classifier extracts features from candidate boxes and classifies them based on the extracted features and predefined object categories. Two-stage detection algorithms have advantages such as high accuracy, strong robustness, and interpretable detection accuracy. However, they also have drawbacks, including slow speed, the need for more computational resources, and challenges in detecting small objects [7]–[9]. The following sections will introduce typical two-stage deep learning object detection algorithms in chronological order.

2.1.1 RCNN.. RCNN, the pioneer of deep learning object detection, emerged when object detection faced a bottleneck [10]. In 2012, A. Krizhevsky et al. introduced the groundbreaking AlexNet [1] model based on convolutional neural networks (CNNs), significantly improving image recognition accuracy in the ImageNet challenge. Researchers recognized the robustness and ability of CNNs to learn high-level features from images. In 2014, R. Girshick introduced RCNN, bringing CNNs into the field of object detection.

RCNN is a simple and scalable object detection algorithm that combines two ideas: (1) applying high-capacity CNNs to generate bottom-up candidate boxes for object localization and segmentation, and (2) performing supervised pre-training in an auxiliary task when labeled training data is insufficient, followed by fine-tuning in the specific domain to significantly improve performance. RCNN consists of three modules. Firstly, the first module uses selective search to generate a series of candidate boxes of various sizes and aspect ratios. Secondly, in the second module, these differently sized and aspect ratio candidate boxes are transformed and scaled into fixed-size images as input to a pre-trained model on the ImageNet dataset to extract high-level features for each candidate box. Finally, in the third module, a linear SVM classifier predicts the classification based on the extracted high-level features for each candidate box. RCNN achieved significant performance improvement, with a 50% increase in mAP accuracy compared to the best previous result on the VOC2012 dataset, reaching 62.4%.

Despite RCNN's remarkable performance, it has clear drawbacks. The algorithm generates a large number of highly overlapping candidate boxes in the first module, and in the second stage, the CNN model needs to independently extract features for each candidate box and store them in memory. This leads to redundant computations and high memory consumption, resulting in slow speed and high memory usage. Additionally, the algorithm divides into three modules: generating candidate boxes, extracting features, and training the object classifier. The training process requires optimization over multiple stages, making it complex and time-consuming. To address these issues, in the same year, He et al. proposed the SPPNet algorithm, improving these problems.

2.1.2 SPPNet. Before this, object detection algorithms based on convolutional neural networks, such as RCNN, necessitated fixed-size input images (224 × 224) during the feature extraction stage. Furthermore, the independent feature extraction computations for 2000 candidate boxes resulted in numerous redundancies, leading to drawbacks such as slow algorithmic speed. Addressing these challenges, the SPPNet algorithm significantly mitigates these issues.

The SPPNet algorithm [11] is characterized by the following features: (1) it accommodates images of arbitrary sizes and aspect ratios as inputs for feature extraction, ultimately generating feature representation vectors of a consistent fixed length; (2) irrespective of the number of candidate boxes, only one convolutional operation for feature extraction on the entire image is required. SPPNet demonstrates flexibility in handling images of varying sizes, proportions, and rotation angles, exhibiting robustness and reduced overfitting; shared computations optimize the most time-consuming feature extraction process. Under conditions where accuracy is comparable to RCNN (VOC mAP=59.2%), SPPNet achieves a computation speed increase of 24-102 times. The fundamental idea of SPPNet involves introducing a spatial pyramid pooling layer before the fully connected layers of the CNN network, utilizing multi-level spatial modules to enhance algorithmic robustness [12], [13]. Specifically, assuming the size of the feature map outputted after the last convolutional layer in the CNN network is denoted as $a \times a$, and each spatial module has a size of $n \times n$, the pooling window size is $\lfloor a \ / \ n \rfloor$, the stride is $\lfloor a \ / \ n \rfloor$, resulting in a fixed-length feature vector after pooling. Subsequently, the feature vectors obtained after pooling in each spatial module are concatenated and fed into the subsequent fully connected layers.

Despite the significant progress achieved by the SPPNet algorithm in terms of speed, certain limitations persist. The introduced spatial pyramid pooling layer renders SPPNet insensitive to precise object location information, presenting challenges in achieving higher accuracy. Moreover, within the spatial pyramid pooling layer, pooling operations across multiple levels of spatial modules

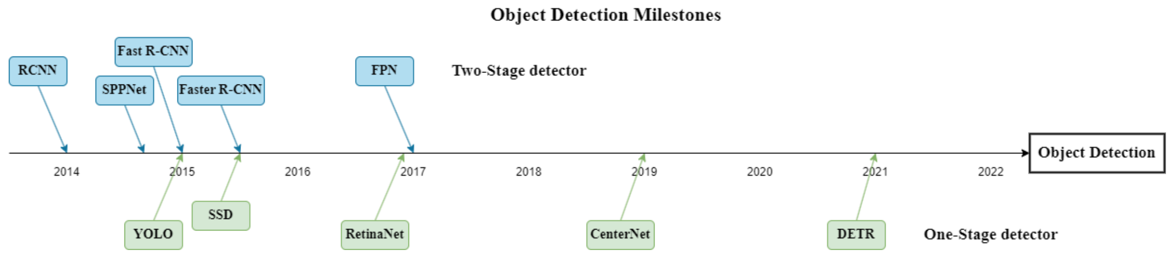


Figure 2: One-phase and two-phase detection algorithms for object detection classics published in chronological order in the last 10 years. Including one-stage detector algorithms YOLO, SSD, RetinaNet, CenterNet, and DETR; two-stage detector algorithms RCNN, SPPNet, Fast R-CNN, Faster R-CNN, and FPN

and their concatenation lead to heightened computational complexity and memory consumption when processing large-sized images. Additionally, fine-tuning algorithms only impact the fully connected layers post the spatial pyramid pooling layer, affecting accuracy when deploying models with deeper convolutional networks. Addressing these challenges, the Fast R-CNN proposed by Ross Girshick [7] has made noteworthy improvements.

2.1.3 *Fast R-CNN.*

Fast R-CNN [7] builds upon previous object detection algorithms based on convolutional neural networks (such as RCNN and SPPNet). It overcomes notable drawbacks in SPPNet training, such as issues in feature extraction and relying solely on the logarithmic loss function for model fine-tuning. The algorithm exhibits significant improvements in both speed and accuracy.

In 2015, R. Girshick proposed Fast R-CNN, introducing a one-stage training algorithm capable of simultaneously learning class detectors and bounding box regressors. During training, hierarchical sampling is employed, and a Region of Interest (RoI) pooling layer is introduced. This layer utilizes max pooling to transform features from any valid region of interest into a small feature map with a fixed spatial size of $H \times W$. RoIs from the same image share computations and memory during both forward and backward propagation. Fast R-CNN includes two sibling output layers: the first yields a discrete probability distribution for each RoI's class, computed using softmax; the second employs the $\mathrm{L}_{\mathrm{loc}}$ loss function to train the bounding box regressor. On the VOC 2012 dataset, Fast R-CNN is three times faster in training speed than SPPNet, achieving higher accuracy with a mAP of 68.4%.

Fast R-CNN represents a significant milestone in the field of object detection, demonstrating substantial improvements in both speed and accuracy compared to traditional R-CNN models. However, Fast R-CNN still has some limitations, particularly in the time-consuming and less accurate candidate box search algorithm. These limitations were further addressed by subsequent algorithms, notably Faster R-CNN [8].

2.1.4 *Faster R-CNN.*

The computational aspect of candidate box search in Fast R-CNN was considered a performance bottleneck. To address this issue, in 2015, S. Ren proposed Faster R-CNN [8]. This algorithm introduced a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, achieving almost zero-cost candidate box search. RPN is a fully convolutional network capable of simultaneously predicting object boundaries and object scores at each position.

Faster R-CNN employs RPN, allowing images of arbitrary sizes as input and producing a series of rectangular target boxes, each accompanied by a target score. The algorithm models this process using a fully convolutional network, and in comparison to Fast R-CNN, the Faster R-CNN network and the Fast R-CNN network share a common set of convolutional layers, enabling computational sharing. On the VGG-16 model [14], the detection system of Faster R-CNN achieves a frame rate of 5fps on GPU, with a detection accuracy of mAP=70.4% on the VOC 2012 dataset. Relative to Fast R-CNN, Faster R-CNN demonstrates significant improvements in both detection speed and accuracy.

Although Faster R-CNN successfully overcame the computational bottleneck of Fast R-CNN, there still exists some computational redundancy in subsequent detection stages. Subsequently, various improved algorithms have been proposed, such as RFCN [15] and Light Head RCNN [16]. These algorithms aim to further optimize the object detection system, enhancing both speed and accuracy.

2.1.5 *Feature Pyramid Networks.*

In practical scenarios, the same object may appear at different scales due to variations in shooting distance, posing a fundamental challenge for computer vision to recognize objects at different scales. To address this challenge and enhance detection performance, Tsung-Yi Lin proposed Feature Pyramid Networks (FPN) in 2017 [9].

FPN leverages the semantic pyramid feature hierarchy inherent in convolutional neural networks, ranging from low-level to high-level features, to construct a feature pyramid consistently possessing high-level semantics. This top-down architecture with lateral connections is effective in building high-level semantics for all scales. FPN takes single-scale images of arbitrary sizes as input and employs a fully convolutional approach to independently output feature maps at multiple levels, proportional in size. Importantly, this process is independent of the underlying convolutional architecture. When integrated into the basic Faster R-CNN system, FPN achieved state-of-the-art single-model results with mAP@.5=59.1% on the COCO detection benchmark. FPN has become a fundamental building block for many new object detection models.

2.2 One-Stage Detection Algorithms

The One-stage object detection algorithm represents a prevalent paradigm in the field of object detection. Diverging from the two-stage counterparts, which entail distinct processes of candidate region extraction and category determination, resulting in prolonged computation times, the One-stage algorithm accomplishes the prediction of all bounding boxes in a single pass through a neural network. Evidently, it yields the advantage of expedited processing, rendering it especially conducive to real-time detection systems and facile deployment on terminal devices like mobile platforms. However, it encounters performance bottlenecks in tasks involving the detection of diminutive and densely clustered objects.

2.2.1 YOLO.. "You Only Look Once" (YOLO) [17], proposed by J. Redmon in 2015, was the first one-stage object detection algorithm. YOLO treats the object detection task as a regression problem, directly predicting the bounding box positions and class probabilities from the image, simplifying model training and optimization.

The most distinctive feature of YOLO is its exceptional speed. In the initial version of YOLO, it achieved up to 45fps, meeting the standard for detecting real-time video streams. At this detection speed, its accuracy also surpassed that of contemporaneous state-of-the-art real-time detection systems by a factor of two. YOLO enables global inference on the entire image during prediction, reducing background detection errors by more than half compared to Fast R-CNN. Moreover, YOLO exhibits strong generalization, with models trained on specific datasets proving effective in new domains.

However, YOLO has its drawbacks. Compared to two-stage detection algorithms, its accuracy in precisely locating objects in images is lower, especially for small objects. The subsequent version, YOLO9000 [18], introduced in 2016, addressed the shortcomings of the first version by incorporating the Darknet-19 network architecture. It improved recall and localization accuracy and expanded its capabilities to detect images of various sizes and classify over 9000 different categories. YOLOv3, introduced in 2018, adopted the Darknet-53 network architecture, introduced multi-scale predictions, and utilized features from different levels for object detection. It employed more Anchor Boxes and introduced higher-resolution feature maps to enhance the detection performance of small objects. Subsequent versions, such as YOLOv4 [19], YOLOX [20], and YOLOv6 [21], although not published by the original authors, made significant contributions to the evolution of YOLO.

The latest publicly released paper for YOLO is YOLOv7 [22]. It introduces dynamic label assignment and reparameterizes the model structure to optimize its architecture. YOLOv7 outperforms most existing object detectors in terms of both speed and accuracy, ranging from 5fps to 160fps.

2.2.2 SSD.. The Single Shot MultiBox Detector (SSD) [23], introduced by W. Liu in 2015, marked a significant advancement in object detection. Prior to SSD, the computational complexity of multi-stage object detection architectures posed challenges, and even the fastest model, Faster R-CNN, achieved a detection speed of only 7fps. Many attempts to improve detection speed came at the cost of accuracy.

SSD presented the first deep network-based object detector. While it does not resample pixels or features for bounding box hypotheses, its accuracy matches that of resampling methods. The key contributions of SSD include being a single-shot detector for multiple classes, faster and more accurate than the leading single-shot detector (YOLO), and as accurate as slower techniques employing explicit region proposals (including Faster R-CNN). The core of SSD lies in using small convolutional filters applied to feature maps to predict class scores and box offsets for a set of fixed default bounding boxes. To achieve high detection accuracy, SSD generates predictions from feature maps of different scales, explicitly separating predictions by aspect ratio. These design features enable straightforward end-to-end training, achieving high accuracy even with low-resolution input images, further improving the trade-off between speed and accuracy.

On the VOC2007 dataset, SSD achieved a performance of 59fps with mAP=74.3%. This outperforms both the best one-stage object detection model, YOLO, which achieved 45fps with mAP=63.4%, and the best two-stage object detection model, Faster R-CNN, which achieved 7fps with mAP=73.2%. Despite its outstanding performance, SSD may face challenges in handling class imbalance. This challenge was addressed in the subsequent RetinaNet [24].

2.2.3 RetinaNet. RetinaNet, proposed by Tsung-Yi Lin in 2017, addressed the challenge of accuracy in one-stage object detectors compared to two-stage object detectors. The primary reason for this lag in accuracy was the extreme foreground-background class imbalance encountered during the training of dense detectors. RetinaNet aimed to tackle issues related to localization accuracy and class balance in object detection.

RetinaNet introduced a novel loss function called "focal loss," reshaping the standard cross-entropy loss. This adjustment focused the detector's attention more on challenging and misclassified examples during the training process. The focal loss enabled single-stage detectors to achieve comparable accuracy to two-stage detectors (COCO mAP@.5=59.1%), while maintaining a very high detection speed.

2.2.4 CenterNet. CenterNet, introduced by Xingyi Zhou in 2019, represents a departure from traditional object detection methods that recognize objects using axis-aligned bounding boxes and enumerate an exhaustive list of potential object positions, requiring additional post-processing. CenterNet simplifies this process.

In CenterNet, objects are represented by a single point or center, streamlining the detection task and reducing the complexity of object localization. To achieve precise object localization, CenterNet employs keypoint regression. Each object's center corresponds to a set of keypoints, describing information such as the object's size, orientation, position, and pose. By regressing the positions of these keypoints, the bounding box or other representations of the object can be reconstructed. CenterNet is designed to be simple and efficient, eliminating the need for complex network structures or multi-stage processes. It exhibits high localization accuracy and robustness, allowing the integration of tasks like 3D object detection and human pose estimation into a single framework. On the MSCOCO dataset, CenterNet achieves an optimal trade-off between speed and accuracy, with an AP of 28.1% at 142 fps, 37.4% at 52 fps, and 45.1% at 1.4 fps during multi-scale testing.

2.2.5 DETR.. In recent years, transformers [25] have had a significant impact on the field of computer vision. Transformers utilize attention mechanisms, overcoming the limited receptive field constraints of traditional CNN models by employing a global perspective. Inspired by this, in 2020, N. Carion introduced DETR [26].

The core idea of DETR is to transform the object detection task into a set prediction problem. It achieves this by taking the input image and a predefined set of object classes and directly outputs a set of predicted bounding boxes and class labels. DETR's architecture consists of two main components: an encoder and a decoder. Given a small set of fixed learned objects and a query set, DETR infers the relationships between objects and the global image context, parallelly producing the final prediction set. Compared to traditional object detection methods, DETR's design is simpler and easier to implement. DETR achieves accuracy and runtime performance comparable to the mature and highly optimized Faster RCNN baseline, with outstanding performance on the MSCOCO dataset, boasting a high mAP@.5 of 71.9%.

3 METRICS

In this chapter, the paper introduces performance metrics for object detection. These metrics provide an objective and quantifiable means to evaluate the performance of object detection algorithm models. By analyzing the algorithm's performance across different metrics, one can identify areas for improvement and optimize the algorithm's design and training processes accordingly. This contributes to the continuous development and enhancement of object detection algorithms. The following sections in this chapter will introduce various measurement metrics.

3.1 Precision VS Recall

Precision and Recall are commonly used evaluation metrics in object detection tasks to measure the accuracy and completeness of an algorithm. They are calculated based on the matching between detected results and ground truth labels.

Precision represents the proportion of true positive samples among the detected positive samples, i.e., the ratio of true positive samples in the detection results. Precision can be expressed using the following formula:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

Here, TP (True Positive) denotes the true positive instances, i.e., the correctly detected positive samples, and FP (False Positive) represents false positive instances, i.e., negative samples incorrectly detected as positive.

Recall indicates the proportion of true positive instances among the actual positive samples, i.e., the ratio of correctly detected positive samples to the total actual positive samples. Recall can be formulated using the following formula:

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

Here, FN (False Negative) represents false negative instances, i.e., positive samples incorrectly classified as negative. Precision and Recall are often trade-offs; improving precision may lead to a decrease in recall, and vice versa.

3.2 F1 Score

In object detection tasks, the aspiration is often to achieve high precision and recall concurrently. To comprehensively assess algorithmic performance, additional metrics such as the F1 score are employed. The F1 score, a harmonic mean of precision and recall, provides a balanced consideration of these two metrics. It is a commonly used comprehensive evaluation metric, particularly effective in tasks involving imbalanced datasets. The F1 score is computed using the following formula:

$$F1 = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \tag{3}$$

With a range between 0 and 1, higher F1 values indicate a better balance between accuracy and completeness. When both precision and recall are high, the F1 score will correspondingly be high. Compared to solely calculating precision or recall, the F1 score offers a more integrated assessment of the trade-off between the two.

The F1 score is particularly useful when dealing with imbalanced datasets, where there is a substantial difference in the quantities of positive and negative samples. In such cases, focusing solely on precision or recall may result in evaluation outcomes influenced by the distribution of sample quantities. The F1 score, by considering both precision and recall, provides a more comprehensive evaluation of the model's performance on positive and negative samples. It is worth noting that the F1 score assumes equal importance for precision and recall and may not be suitable for multi-class classification problems.

3.3 IoU

IoU (Intersection over Union) is a commonly used evaluation metric in object detection, measuring the overlap between the predicted bounding box and the ground truth label. IoU is widely employed in the assessment and optimization of object detection algorithms.

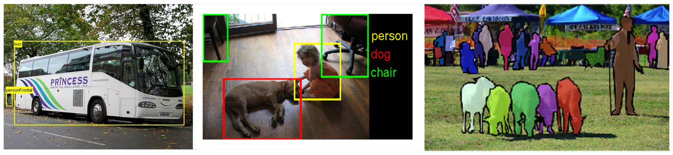
The IoU is calculated by determining the ratio of the intersection area of the predicted box and the ground truth box to their union area. Specifically, assuming the coordinates of the predicted box are $(x1, y1, x2, y2)$ representing the coordinates of the top-left and bottom-right corners, and the coordinates of the ground truth box are $(x1^{'}, y1^{'}, x2^{'}, y2^{'})$, the calculation of the intersection area (Intersection) and union area (Union) is as follows:

$$Intersection = \max\left(0, \min\left(x2, x2^{'}\right) - \max\left(x1, x1^{'}\right)\right)$$
$$\times \max\left(0, \min\left(y2, y2^{'}\right) - \max\left(y1, y1^{'}\right)\right)$$
$$Union = Area\ of\ Prediction\ box +$$
$$Area\ of\ ground\ truth\ box - Intersection$$
$$IoU = \frac{Intersection}{Union}$$

IoU ranges from 0 to 1, where a value closer to 1 indicates a higher overlap between the predicted box and the ground truth, signifying a better match between the detection result and the actual label. Typically, if IoU is greater than a predefined threshold (often set at 0.5 or 0.6), the predicted box is considered a match with the ground truth, indicating a correct detection result. If IoU is below the threshold, the predicted box is considered a mismatch with the ground truth, representing an incorrect detection result. In training and optimization processes, IoU is also commonly used in

Table 1: Selected classical object detection datasets and respective statistics.

Dataset	Train		Validation		Test	
	Images	Objects	Images	Objects	Images	Objects
VOC 2012	5,717	13,609	5,823	13,841	10,991	-
ILSVRC 2014	456,567	478,807	20,121	55,502	40,152	-
MS-COCO 2017	118,287	860,001	5,000	36,781	40,670	



(a) (b) (c)

Figure 3: (a), (b), (c) shows the image samples selected from the datasets PASCAL VOC2012, ILSVRC and MS-COCO respectively.

calculating loss functions. IoU is frequently employed in computing the mean Average Precision (mAP) for object detection algorithms.

3.4 mAP

mAP (mean Average Precision) is a commonly used evaluation metric in object detection tasks, providing a comprehensive assessment of algorithm accuracy across different categories. mAP measures algorithm performance by balancing precision and recall for each class.

The calculation process for mAP is as follows:1. For each class, calculate precision and recall. 2. Precision and recall are typically computed using various IoU thresholds. Common IoU threshold ranges from 0.5 to 0.95, with a set of thresholds covering different matching requirements. 3. For each class, calculate a set of precision and recall values based on different IoU thresholds, then obtain a smooth precision-recall curve through interpolation. 4. On the precision-recall curve, calculate the maximum precision at each recall level (referred to as interpolated precision). 5. Average the interpolated precision for all classes to derive the mAP value.

The mAP value ranges from 0 to 1, with a higher value indicating better algorithm performance in object detection tasks. Generally, a higher mAP value signifies higher precision and recall across different classes, enabling more accurate detection of target objects. mAP is frequently employed in competitions and benchmark datasets such as PASCAL VOC and COCO. It provides a comprehensive consideration of detection results for different classes and exhibits good robustness in cases of class imbalance.

4 DATASETS AND EXPERIMENTS

Data plays a crucial role in the realm of deep learning, and constructing large datasets with minimal bias and high quality is of paramount importance for the advancement of object detection algorithms. Over the past decade, numerous outstanding publicly available datasets have been released, propelling the development of object detection algorithms. These include datasets from the PASCAL VOC Challenges [27], ImageNet Large Scale Visual Recognition Challenge [28], and MS-COCO [29] Detection Challenge, among others. Figure 3 displays a selection of sample images from these datasets, Table 1 provides statistical information for each dataset, and Figure 4 illustrates experimental results on datasets such as VOC2012, ILSVRC2014, and MS-COCO 2017 from papers on various object detection algorithms, including RCNN, SPPNet, Fast RCNN, Faster RCNN, YOLO, SSD, RetinaNet, FPN, CenterNet, and DETR.

4.1 PASCAL VOC

The PASCAL VOC Challenge, which commenced in 2005 and concluded in 2012, provided a large publicly available annotated image dataset encompassing multiple object categories and pixel-level segmentation annotations. Two frequently utilized versions for object detection are VOC 2007 and VOC 2012, containing 20 categories such as people, animals, vehicles, and indoor scenes. The VOC 2007 object detection dataset comprises 9,963 images with a total of 24,640 annotated objects, distributed among training, validation, and test sets with 2,501, 2,510, and 4,952 images, respectively. The VOC 2012 object detection dataset includes approximately 23,080 images containing a total of 27,450 annotated objects, with 5,717, 5,823, and 11,540 images in the training, validation, and test sets, respectively.

4.2 ILSVRC

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) has significantly propelled the advancement of computer vision. Held annually from 2010 to 2017, ILSVRC is associated with the ImageNet dataset, characterized by high image resolution. The ILSVRC object detection subset comprises 200 categories, with a significantly larger number of images and annotated objects compared to PASCAL VOC.

4.3 MS-COCO

The MS-COCO dataset stands out as one of the most classic datasets in the field of object detection. It comprises 330,000 images with annotations for 1.5 million instances across 80 categories. While the number of categories in MS-COCO is fewer than ILSVRC, it surpasses ILSVRC in terms of annotated instances. Unlike ILSVRC, the images in the MS-COCO dataset are predominantly from real-life scenarios, featuring more complex backgrounds, a higher number of objects per image, and a greater abundance of small objects. Additionally, MS-COCO provides detailed segmentation annotations for each instance, enhancing localization precision.

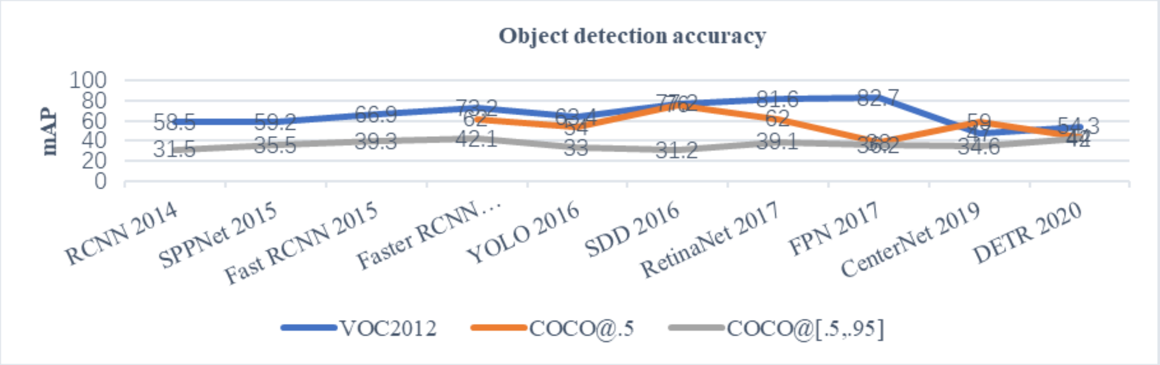


Figure 4: Experimental results of RCNN, SPPNet, Fast RCNN, Faster RCNN, YOLO, SDD, RetinaNet, FPN, CenterNet and DETR on datasets VOC2012, ILSVRC2014 and MS-COCO 2017.

5 CONCLUSIONS

Object detection plays a crucial role in the field of computer vision. This paper provides a comprehensive overview of the development of object detection algorithms and performance evaluation metrics. Through the introduction of one-stage and two-stage object detection algorithms, insights are gained into their differences in detection speed and accuracy, as well as their applicability in different scenarios. Additionally, detailed explanations are given for performance metrics such as precision, recall, F1 score, Intersection over Union (IoU), and mean Average Precision (mAP), serving as effective tools for assessing the accuracy, completeness, and robustness of algorithms.

In terms of datasets, the paper emphasizes the significance of publicly available datasets like PASCAL VOC, ILSVRC, and MS-COCO. These datasets provide rich image and annotation data, driving the research and evaluation of object detection algorithms. Researchers can conduct experiments and comparisons based on these datasets to enhance the performance and applications of algorithms.

The primary significance of this paper lies in systematically introducing the fundamental concepts, performance metrics, and datasets related to object detection algorithms, while discussing their importance. By understanding the characteristics and tradeoffs of different algorithms, researchers and practitioners can select algorithms suitable for their specific application scenarios and improve them based on performance metrics. Furthermore, the analysis of experiments and results on public datasets contributes to the advancement and application of object detection algorithms.

Future research could delve deeper into improving and optimizing object detection algorithms. This involves leveraging the strengths of different algorithms, designing novel models and architectures, and enhancing the accuracy and efficiency of object detection. Additionally, exploring more datasets and scenarios will challenge the robustness and generalization capabilities of object detection algorithms. The ultimate goal is to achieve a more accurate, efficient, and practical object detection system, thereby advancing the application and development of computer vision technology across various domains.

ACKNOWLEDGMENTS

This research work was supported by the Guangxi Key R&D Plan Project (AB21220011; AB21075004: AB21220038) and the Guilin Science and Technology Plan Project (20210220).

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Advances in Neural Information Processing Systems, Curran Associates, Inc., 2012. Accessed: Mar. 26, 2023. [Online]. Available: https://papers.nips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

[2] "Robust Real-Time Face Detection | SpringerLink." https://link.springer.com/article/10.1023/B:VISI.0000013087.49260.fb, accessed Jun. 05, 2023.

[3] "Histograms of oriented gradients for human detection | IEEE Conference Publication | IEEE Xplore." https://ieeexplore.ieee.org/document/1467360, accessed Jun. 05, 2023.

[4] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in 2008 IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2008, pp. 1–8. doi: 10.1109/CVPR.2008.4587597.

[5] "Rapid object detection using a boosted cascade of simple features | IEEE Conference Publication | IEEE Xplore." https://ieeexplore.ieee.org/abstract/document/990517, accessed Jun. 05, 2023.

[6] "Selective Search for Object Recognition | SpringerLink." https://link.springer.com/article/10.1007/s11263-013-0620-5, accessed Jun. 05, 2023.

[7] R. Girshick, "Fast R-CNN," presented at the Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448. Accessed: Jun. 05, 2023. [Online]. Available: https://openaccess.thecvf.com/content_iccv_2015/html/Girshick_Fast_R-CNN_ICCV_2015_paper.html

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in Advances in Neural Information Processing Systems, Curran Associates, Inc., 2015. Accessed: Jun. 05, 2023. [Online]. Available: https://proceedings.neurips.cc/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html

[9] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125. Accessed: Jun. 04, 2023. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017/html/Lin_Feature_Pyramid_Networks_CVPR_2017_paper.html

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-Based Convolutional Networks for Accurate Object Detection and Segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 1, pp. 142–158, Jan. 2016, doi: 10.1109/TPAMI.2015.2437384.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 9, pp. 1904–1916, Sep. 2015, doi: 10.1109/TPAMI.2015.2389824.

[12] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," in 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), Jun. 2006, pp. 2169–2178. doi: 10.1109/CVPR.2006.68.

[13] K. Grauman and T. Darrell, "The pyramid match kernel: discriminative classification with sets of image features," in Tenth IEEE International Conference

on Computer Vision (ICCV'05) Volume 1, Oct. 2005, pp. 1458-1465 Vol. 2. doi: 10.1109/ICCV.2005.239.

[14] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition." arXiv, Apr. 10, 2015. doi: 10.48550/arXiv.1409.1556.

[15] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks." arXiv, Jun. 21, 2016. doi: 10.48550/arXiv.1605.06409.

[16] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "Light-Head R-CNN: In Defense of Two-Stage Object Detector." arXiv, Nov. 22, 2017. doi: 10.48550/arXiv.1711.07264.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.

[18] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger." arXiv, Dec. 25, 2016. doi: 10.48550/arXiv.1612.08242.

[19] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection." arXiv, Apr. 22, 2020. doi: 10.48550/arXiv.2004.10934.

[20] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO Series in 2021." arXiv, Aug. 05, 2021. doi: 10.48550/arXiv.2107.08430.

[21] C. Li *et al.*, "YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications." arXiv, Sep. 07, 2022. doi: 10.48550/arXiv.2209.02976.

[22] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors." arXiv, Jul. 06, 2022. doi: 10.48550/arXiv.2207.02696.

[23] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," 2016, pp. 21–37. doi: 10.1007/978-3-319-46448-0_2.

[24] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," presented at the Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2980–2988. Accessed: Jun. 02, 2023. [Online]. Available: https://openaccess.thecvf.com/content_iccv_2017/html/Lin_Focal_Loss_for_ICCV_2017_paper.html

[25] A. Vaswani *et al.*, "Attention is All you Need," in Advances in Neural Information Processing Systems, Curran Associates, Inc., 2017. Accessed: Apr. 01, 2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[26] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers." arXiv, May 28, 2020. doi: 10.48550/arXiv.2005.12872.

[27] "The Pascal Visual Object Classes (VOC) Challenge | SpringerLink." https://link.springer.com/article/10.1007/s11263-009-0275-4 (accessed Jul. 09, 2023).

[28] "ImageNet: A large-scale hierarchical image database | IEEE Conference Publication | IEEE Xplore." https://ieeexplore.ieee.org/document/5206848 (accessed Jul. 09, 2023).

[29] T.-Y. Lin *et al.*, "Microsoft COCO: Common Objects in Context." arXiv, Feb. 20, 2015. doi: 10.48550/arXiv.1405.0312.