DOI: http://dx.doi.org/10.26483/ijarcs.v13i2.6804

ISSN No. 0976-5697

Volume 13, No. 2, March-April 2022

International Journal of Advanced Research in Computer Science

REVIEWARTICLE

Available Online at www.ijarcs.info

A REVIEW ON OBJECT DETECTION SINCE 2005

Nazeer Haider
Department of Computer Science and Engineering
National Institute of Technology
Jamshedpur, India

Abstract: Recent years have seen a significant increase in interest in object detection, which is considered to be one of the most fundamental and demanding computer vision tasks. It might be called the pinnacle of computer vision history, because of its rapid progress since 2005. If we regard today's object identification to be a kind of deep learning-powered technical aesthetics, then rewinding the clock since 2005 would allow us to observe the wisdom of the cold war period. This study examines studies on object detection since 2005 in the context of technological advancements that have occurred throughout a quarter-century. This article covers a wide range of subjects, including historical milestone detectors, metrics, datasets, speed-up strategies, etc. This article also reviews important traditional object detections (DPM and HOG), single-stage object detection methods (YOLOR, RetinaNet, SSD, YOLO), and two-stage object detection methods (Mask-RCNN, Faster-RCNN, Fast-RCNN, SPPNet, R-CNN).

Keywords: Object detection, Localization, Classification, Technical evolution, Deep learning, Computer vision, Convolutional neural networks (CNNs).

I. INTRODUCTION

The computer vision technique of object detection is an approach that is used to locate and classify occurrences of things in images or videos using computer vision algorithms. When it comes to delivering relevant results, object detection algorithms frequently rely on machine learning or deep learning approaches to do this. When gazing at or viewing images or videos, the human eye can recognize and pinpoint things of interest in a matter of seconds. One of the goals of object detection is to utilize a computer system to mimic human intelligence.

The detection of objects is a basic concept in computer vision, and it is also one of the most difficult. It serves as the foundation for a huge range of downstream computer vision operations, including instance segmentation, object tracking, image captioning, and other applications. The detection of pedestrians, counting of the number of people in a given area, face recognition, text detection, pose detection, and number-plate recognition all are examples of computer vision applications,specifically, object detection applications.

Fig.1 Classification

Fig.2 Localization

Object Detection = Classification + Localization

Fig. 3 Object Detection

Differences between Image Classification, Object Localization, and Object Detection

1) Classification: We can tell what kind of thing is in an image with only one object. This is a classification case as shown in Fig. 1.

2) *Localization:* The next step is localization, in which we not only determine what type of object it is but also where it is in the image. Programmatically, we must draw the correct bounding box around it, as shown in Fig. 2.

3) *Object Detection:* If the image contains one or multiple objects, we must locate and identify each one. This is known as Object Detection as shown in Fig. 3.

II. OBJECT DETECTION: A ROAD MAP

Generally, object detection has been divided into two historical periods over the past two decades, as illustrated in Fig. 4. The first is known as the *Before Deep Learning (Traditional way of Object Detection)* and the second is known as the *After Deep Learning (Modern way of Object Detection)* respectively.

A. Before 2014

Before Deep Learning (Classical or Traditional way of Object Detection)

1. HOG Detector (2005), is a feature descriptor for object detection that is largely used in image processing and computer vision.
2. DPM (2008) with Bounding box regression was introduced in this paper.

B. After 2014

After Deep Learning (CNN based or Modern way of Object Detection)

Two-stage most important object detection algorithms

1. SPPNet and RCNN (2014)

2. Faster RCNN and Fast RCNN (2015)
3. RFCN (2016)
4. Mask R-CNN (2017)

Single-stage that is the most significant method for object detection

1. YOLO (2016)
2. SSD (2016)
3. RetinaNet (2017)
4. YOLO Version3 (2018)
5. YOLO Version4 (2020)
6. YOLOR (2021)

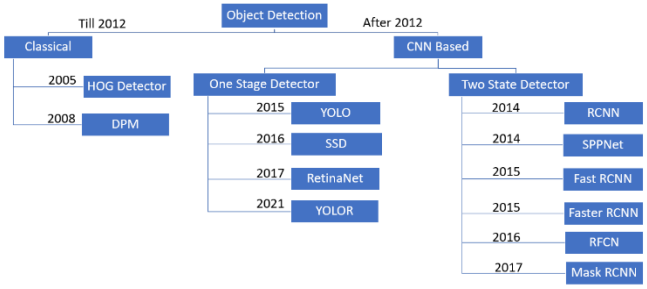


Fig. 4 A Roadmap of object detection since 2005

C. Object Detection in a Traditional Way

If we regard today's object identification to be a kind of deep learning-powered technical aesthetics, then rewinding the clock since 2005 would allow us to observe the wisdom of the cold war period. The majority of the first object identification systems were built with handcrafted properties. Because there was no good picture representation available at the time, people were forced to build complicated feature representations and employ a wide range of accelerations techniques to maximize the utilization with limited availability of computing power.

1) HOG (Histogram of Oriented Gradients)Detector: B. Triggs and N. Dalal [1] introduced the HOG detector in 2005, which is used to find features for object identification and classification. It was a significant advance over previous detectors [2] [3][4] [5]. Using HOG, you may generate a feature table by extracting the gradient and orientation of the edges. Each cell in the grid is represented by a histogram, which is created using the feature table created from the grid divisions and the feature table. HOG features are developed for the Region of Interest(ROI)and sent into a linear SVM (Support Vector Machine) classifier to detect their existence. The detector was originally developed for pedestrian detection; however, it may be taught to recognize a variety of other types of objects as needed.

2) *DPM (Deformable Part-based Model):* Felzenszwalb et al. developed the DPM [6], which won the Pascal VOC competition in 2009. It was able to recognize discrete sections of the item with more precision than HOG because it did not use the whole object. It is built on the divide and rule principle, in which individual elements of an object are identified independently at the period of inference, and a reasonable arrangement of these parts is signaled as detection. As an example, a human body's parts are trained individually such as the torso, legs, arms, head, and other things. One model is trained to capture one whole body, and the procedure will be repeated for all of the components that were captured by that model. After that model eliminates unlikely combinations of pieces to generate detection. Before the introduction of deep learning algorithms, DPM-based models [7] [8]were among the most successful.

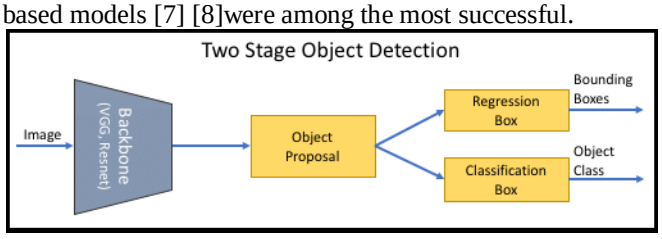


Fig. 4 Two-Stage Object Detection

D. Modern Way of Object Detection

1) Two-Stage Detectors: RCNN and its variants are examples of early-stage networks, as well as other two-stage networks, which are used to identification of potential regions of interest, or subsets of an image that may have an object. The second stage is concerned with categorizing the objects contained inside the region suggestions. Two-stage networks are capable of delivering exceptionally exact object detection results, although they are frequently slower in terms of processing speed than single-stage networks.

a) RCNN (Region-based Convolutional Neural Network): RCNN[9] was the first article in a series of selective search families, and it showed how CNNs may be used to significantly enhance the identification performance of images. CNNs are used in conjunction with a class independent region proposal module to transform an identification issue into a localization and classification problem. The region proposal module, which generates 2000 item candidates from a mean-subtracted input image, is the initial step in the processing. With the help of Selective Search [10], this module identifies areas of a picture where there is a greater likelihood of detecting an item. A CNN is used to extract a 2^{12}-dimension feature vector for every proposal after the candidates are distorted and propagated through the network. As the detector's backbone architecture, Girshick et al. used AlexNet [11]. The feature vectors are then fed into trained, class-specific SVMs, which compute confidence ratings for each class. Afterward, NMS (non-maximum suppression)is used to score the areas following their IoU and class. A trained bounding box regressor is used to forecast the bounding box of the class after it has been recognized. This predicts four parameters, namely the length, breadth, and center coordinate (x,y) value of the object.

R-CNN features a sophisticated multistage training method that takes a long time. A big classification dataset is used to pre-train the CNN in the initial step of the process. Using SGD (stochastic gradient descent), and then fine-tune are used for detecting the domain-specific images by substituting an (N+1) way classifier for the classification layer, where N is the no. of classes. For each class, a bounding box regressor and one linear SVM are trained separately.

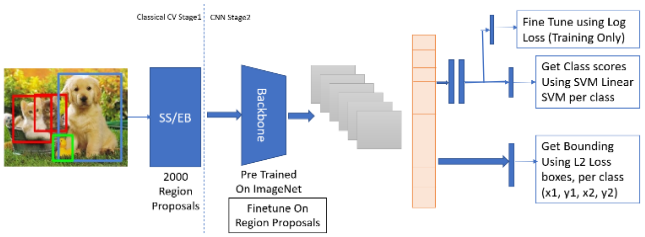


Fig. 5 Architecture of RCNN

Although R-CNN was fast (47 seconds per picture) and costly in terms of both time and space [12], it was a significant step forward in object identification. Although certain calculations were shared, the training method was difficult, and it takes a long time for training on small datasets.

b) *SPPNet:* To analyze images of any size or aspect ratio, He et al. [13]suggested the SPP(Spatial Pyramid Pooling) layer to pool images from several sources. They realized that only the completely linked portion of the CNN needed constant input. Putting the convolution layers first and then adding a pooling layer made the network able to work with any size or aspect ratio. This resulted in a smaller number of calculations and a more stable network. Candidate windows are generated by the application of the selective search [10] algorithm. To make feature maps, you feed an image via the convolution layers of a network called a ZF5 [14]. Feature maps are created from the candidate windows, which are then turned into fixed-length representations using SPP layer spatial bins. This vector is connected to a dense layer and then passed to SVM for classification. Same as RCNN, the SPPNet contains a post-processing layer that uses bounding box regression to increase localization accuracy. It likewise employs the same multistage training procedure, with the exception that fine-tuning is performed only on the layers that are completely connected.

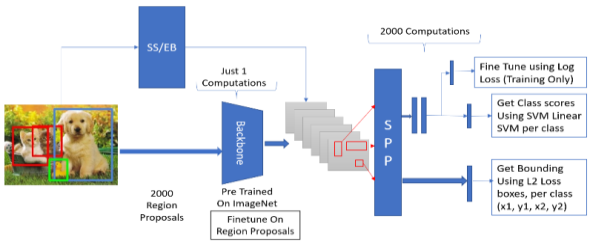


Fig.7 Architecture of SPPNet

The SPPNetmodel's accuracy is approximately the same as RCNN but training and testing are faster than RCNN. It can analyze photos with any shape or aspect ratio and, as a result, prevents object distortion caused by input warping from occurring. However, since its design is similar to RCNN, it has some disadvantages such as multistage training, computationally costly, and more training time.

c) Fast RCNN: One of the most significant problems with SPPNet and RCNN was that required individual training for multiple systems. Fast-RCNN [12]overcome this issue by constructing one end-to-end trainable system. This model takes input as an image and then applies the brute force method in the input image to generate ~2k region proposals. This image is passed through CNNsto generate the feature maps. Now, these feature maps are mapped to object proposals. Using the SPPNet [13], Girshick replaced the pyramidal structure of pooling layers with a single spatial bin, which he named the ROI pooling layer. Now, the output of this layer is also connected to 2dense layers individually for bounding box and classification as shown in Fig. 8. Now SoftMax layer gives N+1 (N classes and 1 for background) output for classification and each class, the smooth L1 loss function gives four outputs (two outputs for the center of an object, and two outputs for height and weight of an object) for the bounding box of an object. One more change here from SPPNet is that the loss function of the bounding box regressor was changed from L2 to smooth L1 to improve the performance of the model.

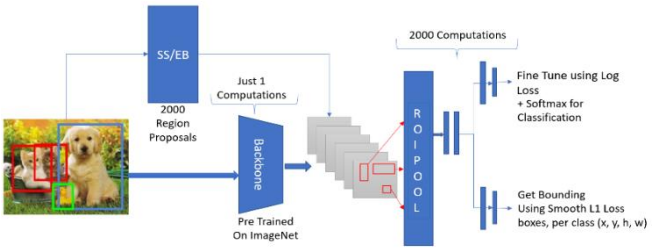


Fig. 8 Architecture of Fast RCNN

The introduction of fast R-CNN was motivated by the need to enhance speed (146x faster than R-CNN), with the increase in accuracy being a bonus. It has streamlined the training approach, abolished pyramidal pooling, and introduced a new loss function to compensate for these changes. Even without the use of a region proposal network, the object detector was able to report near the real-time speed and with high accuracy.

d) Faster RCNN: An object could be identified in real-time by a fast R-CNN, but its region proposal generation was still slower than that of an RCNN (0.2 sec per image). Using an FCA (fully convoluted network) [15], Ren et al. proposed an (RPN) region proposal network [16] that takes an image as an input and produces a collection of candidate regions. Each of these regions has an associated objectness score, which is used to assess the possibility of an object appearing in the region. [17], [6], and [12] used image pyramids to overcome the problem of different-sized objects. RPN introduces anchor boxes, which do not use image pyramids. When it came to localizing the item, it regressed across a lot of bounding boxes with different aspect ratios. The image is fed via the CNNs to get a collection of feature maps. These feature maps are transmitted to an RPN, which generates their classification and bounding boxes. In the ROI pooling layer, the feature map is mapped back to chosen proposals acquired from the preceding CNN layer and finally supplied to the dense layers, which are provided to the bounding box regressor and classifier for localization and classification respectively. Faster RCNN is faster than Fast RCNN because of RPN as a region proposal module.

Because of the existence of common layers between two models that perform quite distinct tasks, the training of Faster R-CNN is more complicated. Firstly, fine-tuned is pre-trained on the PASCAL VOC dataset [18] and RPN on the ImageNet dataset [19]. In the first phase, Fast RCNN is trained using RPNs. Until this stage, the networks don't share a convolution layer. Now we'll fine-tune the unique

layers in RPN and adjust the detector's convolution layers. Finally, the Fast RCNN is fine-tuned based on the newly updated RPN.

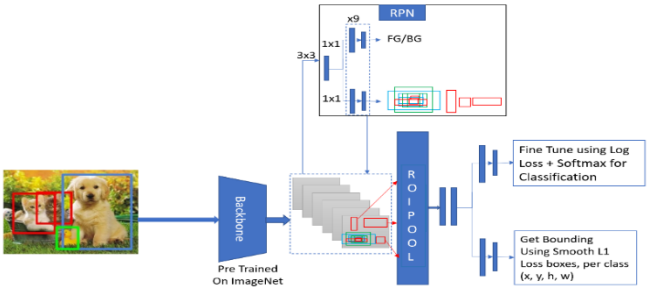


Fig.6 Architecture of Faster RCNN

In Faster RCNN, detection accuracy went up more than 3%, and inference time went down by a factor of magnitude. It eliminated the bottleneck caused by the sluggish region proposal and operated in near real-time at 5FPS. It was also advantageous to include a CNN in the region proposal since it allowed the system to learn to make better recommendations over time, increasing accuracy.

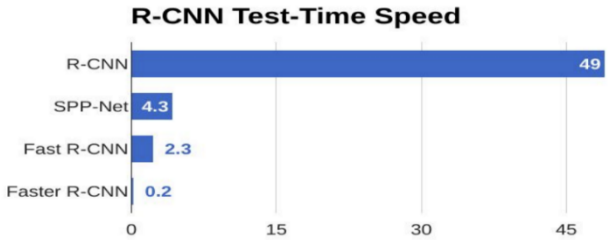


Fig. 7 Time comparison graph of RCNN, SPPNet, Fast RCNN & Faster RCNN

More rapid RCNN is noticeably faster than its predecessors, as illustrated in the graph to the top. As a result, it can even be used to identify objects in real-time if the conditions are right.

e) RFCN (Region-based Fully Convolutional Network): To overcome the limitations of previous two-stage detectors, Dai et al. came up with the Region-based Fully Convolutional Network (RFCN) [20], which did most of the work in one place. This was different from previous two-stage detectors, which used a lot of resources for each proposal. Rather than fully linked layers, they claimed that convolutional layers should be employed in their designs. Deeper layers of the convolutional network, on the other hand, are translation-invariant, rendering them unsuitable for jobs requiring localization. To address this issue, the authors advocated the usage of position-sensitive score maps. Now, these maps capture relative spatial information about the subject, which is then pooled to determine the subject's actual location in the environment. RFCN does this by splitting the ROI into a k × k grid and rating each cell based on its likelihood of being associated with the detected class feature map. The average of these scores is then used to forecast the object class. RFCN detector is made up of 4 CNNs. To get feature maps, the input picture is first passed through the ResNet-101 [17]. The output of the Conv4 layer is sent to an RPN, then the final output passes via a convolutional layer and is used by a regressor and classifier. When combined with the ROI ideas, the position-sensitive map is used to produce predictions, which are then passed on to the classification layer, which generates the bounding box information. When training R-FCN, we use a similar four-step procedure to that of Faster-RCNN [16], but we use a box regression loss and combined cross-entropy.

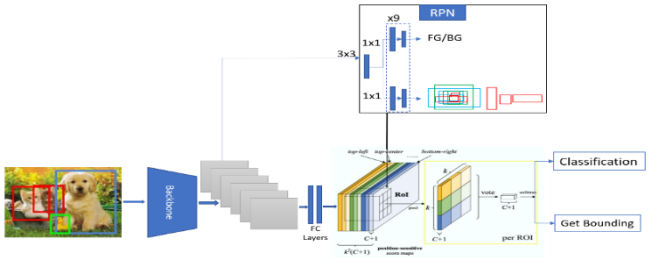


Fig. 8 Architecture of RFCN

To solve the issue of translation invariance in CNNs, Dai et al. proposed a unique technique that was implemented in MATLAB. RFCN is a combination of FCN and Faster RCNN that produces a faster and more accurate detector. Although it did not significantly enhance accuracy, it was 2.5-20 times quicker than its predecessor.

f) Mask RCNN: Mask RCNN [21] is a variant of Faster RCNN that adds pixel-level object instance segmentation to a parallel branch. The branch is an FCA that is used by ROIs to divide every pixel into segments while incurring the least amount of total computation. It has the same basic structure as Faster RCNN, but in addition, is a mask head next to the bounding box regressor and classifier. There was one significant difference, the usage of the ROIAlign layer, rather than the ROIPool layer, was used to eliminate pixel-level misalignment as a result of spatial quantization. To achieve more speed and accuracy, the authors used the ResNeXt-101 [22] as the network's backbone, in conjunction with the FPN(feature Pyramid Network). Faster RCNN's function is modified in response to the mask loss, and it employs the same five anchor boxes with three aspect ratios as FPN. Mask RCNN training is faster than RCNN in terms of overall performance.

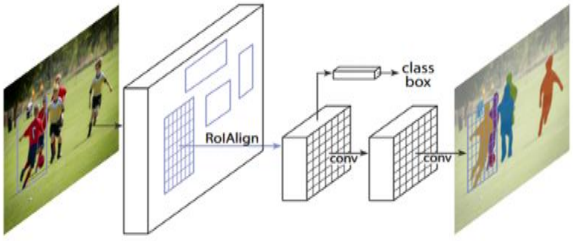


Fig. 9 Architecture of Mask RCNN

RCNN outperformed the state-of-the-art multimodal systems and introduced the feature of instance segmentation with little overhead calculations, outperforming the current state-of-the-art single-model systems as well. When used in applications such as key point recognition, human position prediction, and other similar tasks, it is straightforward to train and generalize. Despite this, the performance remained below the real-time benchmark (>30 frames per second).

2) *Single-Stage Detectors:* Single-stage networks produce final bounding boxes for objects by using anchor boxes to generate network predictions for regions throughout the whole image, for example, YOLO v2. The CNN generates final bounding boxes for objects utilizing anchor boxes in multistage networks such as YOLO v2 and other similar networks. Single-stage networks are faster than two-stage networks, but the accuracy of single-stage networks is less, particularly in cases involving small objects.

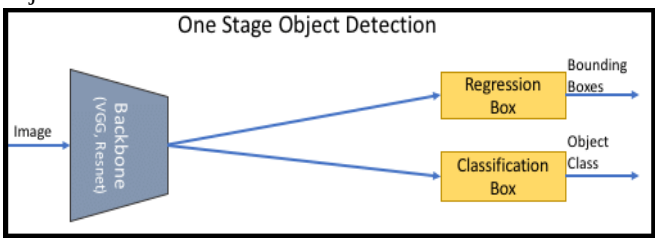


Fig. 10 One Stage Object Detection

a) YOLO (You Only Look Once): It is necessary to partition an image into an $S \times S$ grid, within which there are m bounding boxes, to show the resultant grid on a computer screen, to utilize the YOLO [23] algorithm. A class bounding box and probability offset values are calculated for each bounding box in the network, with the network serving as the basis for the calculations. To find the item in the image, bounding boxes with a class probability larger than a threshold value are selected and used, and the results are presented on the screen as they are calculated.

A typical object detection technique takes hundreds of milliseconds to detect a single item (45 frames per second). Although there are several advantages to using the YOLO approach, one downside is that it has difficulties differentiating between little items in photos; for example, it may have problems distinguishing between a flock of birds and other objects. This is specifically owing to the geographical limits imposed by the method.

The accuracy and speed of YOLO's single-stage real-time models outperformed their contemporaneous counterparts by a wide margin. It did, however, have some important flaws as well as strengths. The precision with which it could locate tiny or grouped items, as well as the restriction on the number of objects that could be placed in a cell, was its significant limitations. These flaws were addressed in subsequent versions of YOLO [24] [25] [26], which were released after this.

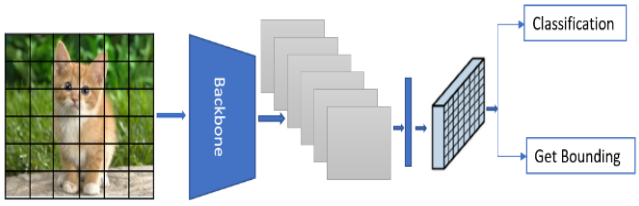


Fig. 11 Architecture of YOLO

Its one-stage detector design allows it to complete bounding box regression and classification in a single step, which makes it substantially faster than most convolutional neural networks. To give an example, when it comes to object identification and tracking, RCNN is 1000 times quicker than YOLO, while Fast RCNN is 100 times faster.

b) SSD (Single-shot detector): SSD [27] is the first one-stage detector that has the same accuracy as a two-stage detector with real-time speed. It was based on the VGG16 [28]architecture, with extra CNN layers added to boost overall performance. The size of these supplementary convolution layers, which are inserted towards the end of the model, decreases gradually over time. During the early stages of the network when the picture characteristics are not too crude, SSD recognizes smaller objects, while the deeper layers were in charge of adjusting the aspect ratios and offset of default boxes[29].

SSD selects the default boxes with the best Jaccard overlap for each ground truth box during training, like multi box [29], and then trains the network appropriately. They also used hard negative mining techniques as well as extensive data augmentation. In a similar vein to DPM [33], it trained the model using a weighted sum of the confidence loss and localization datasets. It is possible to achieve the final output by doing non-maximum suppression.

Although SSD was substantially quicker and more accurate than both the YOLO and Faster R-CNN networks, it had difficulties distinguishing between tiny items in the data set. This problem was subsequently resolved by the use of improved backbone topologies such as ResNet, as well as other tiny modifications.

This object detection component is simple to train, and it may be easy to integrate into software systems that require object identification functionality. SSD offers significantly higher accuracy when compared to other single-stage approaches, even when working with smaller input image sizes.

c) RetinaNet: Even though one-stage detectors are quicker and easier to use than two-stage detectors, they have consistently performed worse than two-stage detectors in terms of accuracy. T.Y. Lin et al. [30] investigated the reasons behind this and offered RetinaNet as a potential remedy in 2017. During the training of dense detectors, they observed a significant increase in the foreground-background class imbalance, which they concluded was the fundamental cause of the issue. RetinaNet has devised a novel loss function known as "focused loss" to do this by changing the traditional cross-entropy loss [31] in such a way that the detector places a greater emphasis on problematic, misclassified samples during training. Because of focal loss, one-stage detectors can achieve accuracy that is comparable to that of two-stage detectors while maintaining extremely fast detection speeds. (91.1 percent for COCO mAP@0.5, and 39.1 percent for mAP@[0.5,.95]).

d) YOLOR (You Only Learn One Representation): YOLOR [32] is a revolutionary object detector that was first introduced in the year 2021. Model training is accomplished using both implicit and explicit knowledge, which is applied simultaneously by the algorithm. Accordingly, YOLOR is capable of learning a general representation and performing a variety of tasks using this general representation.

Integrating implicit knowledge into explicit knowledge is accomplished by multi-task learning, kernel space alignment, prediction refinement, and other techniques.

Using this strategy, YOLOR can significantly increase the performance of its object detecting algorithms.
According to the COCO dataset benchmark, the mean absolute performance (MAP) of YOLOR is 3.8 percent greater than the mean absolute performance (MAP) of PP-YOLOv2, although both methods run at the same inference speed. According to the researchers, as compared to the Scaled-YOLOv4, the inference speed has been increased by 88 percent, making it the fastest real-time object detector currently available on the market. For further details, please see our dedicated topic on YOLOR - You Only Learn One Representation.

III. Object Detection Datasets and Metrics

It is essential for the development of improved computer vision algorithms that larger datasets with less bias are gathered and analyzed. On the subject of object detection, many well-known benchmarks and datasets have been published in the previous ten years, including datasets from the PASCAL VOC Challenges [18] [33] (for example, VOC2007, VOC2012), ILSVRC2014 [34], MS-COCO Detection Challenge, and others. These datasets and benchmarks include the PASCAL VOC Challenges Table 2 of this publication contains the statistics for the datasets discussed in this section. The detection accuracy of the VOC07, VOC12, and MS-COCO datasets has improved significantly, as shown in Fig. 3.

A. Pascal VOC

One of the most well-known competitions in the early computer vision industry was the PASCAL VOC (Visual Object Classes) Challenges, which were held every year from 2005 to 2012 [18] [33]. PASCAL VOC is capable of completing a wide range of tasks with semantic segmentation, object identification, image classification, action detection, etc. The most commonly used versions of Pascal-VOC in object identification are VOC12 and VOC07, with the former containing 5000training images plus 12000 annotated objects and the latter including 11000training images plus 27000 annotated objects. These two datasets contain annotations for 20 different classes of objects that are commonly encountered in everyday life (person; animal: sheep, horse, dog, cow, cat, bird; vehicle: train, motorbike, car, bus, boat, bicycle, airplane; indoor: monitor/tv, sofa, potted plant, dining table, chair, bottle). As more huge datasets become available, such as the ILSVRC and the upcoming MS-COCO (which will be launched), the VOC has rapidly lost favor, and it is now being used to evaluate the bulk of new detection technologies.

B. ILSVRC

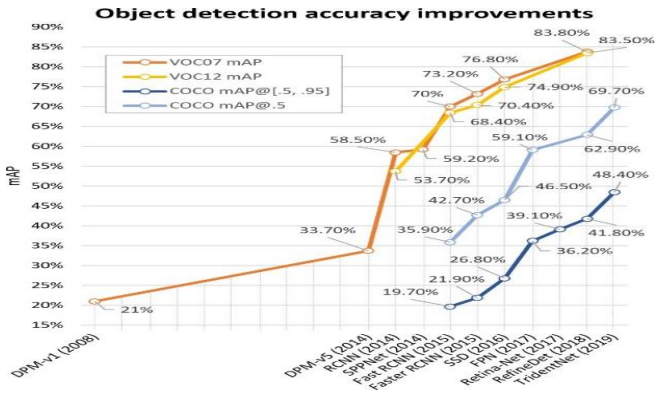
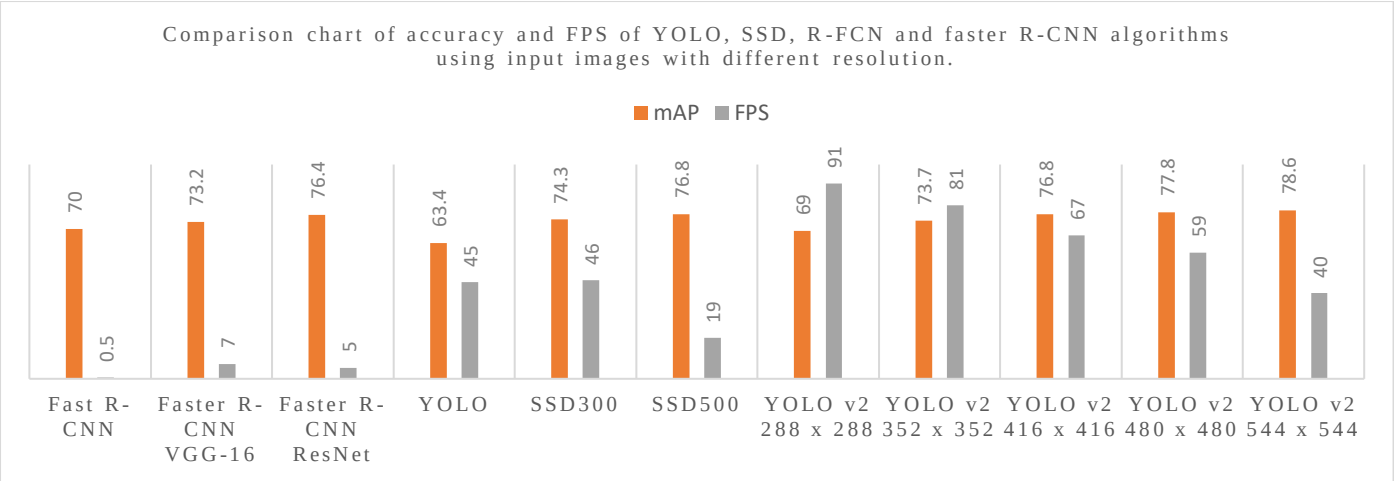


Fig.12: Object detection accuracy improvement graph on MS-COCO, VOC12, and VOC07, datasets. Detectors in this figure: DPM-v5, DPM-v1, Fast RCNN, Faster RCNN, SPPNet, RCNN

To push the state-of-the-art in generic object identification further, the ILSVRC (ImageNet Large Scale Visual Recognition Challenge) [34] has pushed the frontiers of what is achievable in the field. Each year from 2010 to 2017, the ILSVRC is held in a different location. It includes a detection task that makes use of ImageNet images [19]. In total, there are 200 classes of visual objects in the ILSVRC detection dataset. When compared to the amount of VOC images/object instances, it has two orders of magnitude more of the latter type of image/object instance. In the case of the ILSVRC-14, there are 517k photos and 534k annotated items in the collection.

C. MS-COCO

MS-COCO [35] is the most demanding dataset for object identification available at the time of writing. Since 2015, there has been a yearly competition based on the MS-COCO dataset. The number of object categories is less than that of the ILSVRC, However, there are a greater number of object instances. MS-COCO-17, for example, has 164,400 images and 897,400 tagged objects from 80 different categories. While MS-COCO has made significant strides in comparison to VOC and ILSVRC, the most significant difference is that the bounding box annotations, to facilitate exact localization, each item is additionally tagged using per-instance segmentation. Moreover, unlike MS-COCO and ILSVRC, VOC comprises a greater number of small objects (those with a surface area of less than one percent of the image) and objects that are more densely clustered. As a result of all of these qualities, the object distribution in MS-COCO is more similar to that of the real world. MS-COCO has risen to the top of the object detection community's priority list, just like ImageNet did when it first appeared. These characteristics bring the distribution of things in MS-COCO more like the actual world than it was previously. MS-COCO has established itself as the industry standard for object detection, much like ImageNet did when it was first introduced.



D. Open Images

In the year 2018, the Open Images Detection (OID) challenge, which is modeled after the MS-COCO challenge but on a much bigger scale, is introduced. It is possible to complete two tasks in Open Images, one of which is the regular object detection task, and the other of which is the visual connection identification test, which finds paired items that have certain relationships between them. The dataset used for object detection contains 1910k images including 15440k bounding boxes annotated on 600 different item categories.

Table 1 The test set PASCAL VOC 2007 was used to determine the performance of the SSD, Faster RCNN, and YOLO techniques in terms of speed and accuracy [24].

Detection Frameworks	***FPS***	***mAP***	***Train***
YOLO v2 544 x 544	40	78.6	2007+2012
YOLO v2 480 x 480	59	77.8	2007+2012
YOLO v2 416 x 416	67	76.8	2007+2012
YOLO v2 352 x 352	81	73.7	2007+2012
YOLO v2 288 x 288	91	69.0	2007+2012
SSD500	19	76.8	2007+2012
SSD300	46	74.3	2007+2012
YOLO	45	63.4	2007+2012
Faster RCNN ResNet	5	76.4	2007+2012
Faster RCNN VGG16	7	73.2	2007+2012
Fast RCNN	0.5	70.0	2007+2012

Table 2 Several well-known object identification datasets, as well as their associated statistics

Dataset	***Train***		***Validation***		***Train***		***Test***	
	Objects	***Images***	***Objects***	***Images***	***Objects***	***Images***	***Objects***	***Images***
VOC-2007	6,301	2,501	6,307	2,510	12,608	5,011	14,976	4,952
VOC-2012	13,609	5,717	13,841	5,823	27,450	11,540	-	10,991
ILSVRC-2014	478,807	456,567	55,502	20,121	534,309	476,688	-	40,152
ILSVRC-2017	478,807	456,567	55,502	20,121	534,309	476,688	-	65,500
MS-COCO-2015	604,907	82,783	291,875	40,504	896,782	123,287	-	81,434
MS-COCO-2018	860,001	118,287	36,781	5,000	896,782	123,287	-	40,670
OID-2018	14,610,229	1,743,042	204,621	41,620	14,814,850	1,784,662	625,282	125,436

Table 3 Strengths and weaknesses of object detection methods.

Methods	***Authors***	***Key Contributions***	***Strengths***	***Weaknesses***
HOG	Dalal and Triggs, 2005	Gradient orientation	• Shows invariance to geometric and photometric changes	• Very sensitive to image rotation, • Unable to detect overlapped objects
DPM	Pedro F.	Part templates	• Detect rotated and overlapped objects	• Complex computation and

	Felzenszwalb et al., 2008			time have taken a process
RCNN	Girshick et al., 2014	Selective Search	• The first to combine CNN and RP techniques. • Significantly better performance than prior object detectors.	• Sequentially-trained multistage pipeline (BBR training, SVM every warped RP feeding to CNN, CNN finetuning, and External RP computation). • Training is costly in terms of both time and space. • Testing takes a long time.
SPPNet	He et al., 2014	Spatial pyramid pooling layer	• Enable the exchange of convolutional features. • Increase the speed of RCNN assessment by an order of magnitude while maintaining performance. • Faster than Over-Feat. • 24 times faster than the RCNN	• Inherit the RCNN's drawbacks. • There isn't much of a speedup in training. • The CONV layers cannot be updated before the SPP layer in Finetuning. • For ~2k regions SPP has 3 layers
Fast RCNN	Ross Girshick, 2015	ROI Pooling	• For the first time, without RP generation, end-to-end detector training is achievable. • Create a layer for ROI pooling. • SPPNet is much slower and less precise. • For feature caching, no disc storage is needed. • 146 times faster than the RCNN	• For real-time applications, this is slow. • The new bottleneck has been identified as external RP computation. • Using an external candidate region generator slows down the detecting process.
Faster RCNN	Ren et al., 2015	Region Proposal Network	• Instead of selective search, propose RPN for high-quality and cost-free RPs. • 10 times faster than the Fast RCNN. • Introduce multiscale anchor boxes and translation invariant anchor boxes as RPN references. • With VGG16, testing can be done at 5 FPS. • By sharing CONV layers, combine Fast RCNN and RPN into a single network.	• Training is a complicated process, not a simple one. • Still lags in real-time. • Not for real-time application.
RFCN	Dai et al., 2016	Position-sensitive score maps	• Detection network that is fully convolutional. • Using a bank of customized CONV layers, create a collection of location-sensitive score maps. • Quicker than Faster RCNN with a high level of accuracy.	• Training is a time-consuming procedure that is not simplified still, it falls short of real-time. • RFCN is less accurate than Faster RCNN
Mask RCNN	He et al., 2017	ROI Align	• Adds additional branch to Faster RCNN for predicting an object mask in addition to the current branch for BB prediction. • Good performance. • The location of an object is more accurate due to segmentation. • FPN is utilized. • An effective, flexible, and simple framework for object detection segmentation.	• Real-time applications are not supported. • Its execution process is greater than Faster RCNN.
YOLO	Redmon et al., 2015	Grid-based Proposal	• The first and most effective unified detector. • Remove the RP process entirely. • Framework for detection that is both elegant and efficient. • Detectors that are much quicker than	• The detector's accuracy lags much behind that of modern detectors. • Small items are difficult to locate. • Struggles to detect both close

			earlier models. • YOLO runs at 45 FPS whereas Fast YOLO runs at 155 FPS. • Real-time applications. • Efficient for locating objects.	and small objects.
SSD	Liu et al., 2016	Featured Pyramid	• The first unified detector is both accurate and efficient. • To accomplish detection at multi-scale CONV layers, effectively incorporate principles from RPN and YOLO. • SSD is substantially quicker and more precise than YOLO. • At 59 FPS, this can run. • Easily detects a small object. • Faster than Faster RCNN.	• Small-item detection is poor. • Less accurate than Faster RCNN.
RetinaNet	Lin et al., 2017	Feature Pyramid Network	• Suggest a new Focal Loss that concentrates training on difficult cases. • This is the first one-stage detector whose accuracy is better than two-stage detectors.	• When training on a single-stage detector, it handles the issue of an imbalance of negative and positive samples well.
YOLOR	Wang et al., 2021	Analyzer Network	• It is the fastest and more accurate among all the above methods.	—

IV. Future Directions

The following factors may be the subject of future object detection research, although they are not limited to:

A. AutoML

The use of autonomous neural architecture search (NAS) to identify the properties of an object detector is already a rapidly expanding field. However, it is still in its infancy, as seen by the 16 detectors created by the National Aerospace and Space Administration (NAS). The search for an algorithm is a time-consuming and resource-intensive endeavor.

B. Lightweight detectors

For the detection method to work well on mobile devices, it has to be made faster. Mobile augmented reality, smart cameras, face identification, and other similar applications are examples of key applications. Despite significant progress in the last few years, the speed difference between a computer and human vision continues to be significant, particularly when it comes to recognizing certain microscopic things.

C. 3D object detection

When it comes to autonomous driving, 3D object identification is a particularly difficult challenge to solve. Although models have attained excellent accuracy, the deployment of anything that falls below human-level performance would raise safety concerns.

D. Object identification in the video

When used on individual images that do not have any association between them, object detectors execute their functions. The use of spatial and temporal relationships between frames to aid in object detection is still a work in progress.

V. Conclusion

Since 2005, significant advances have been achieved in the field of object detection. Throughout its decade-long history, this paper not only provides an in-depth review of some landmark detectors (e.g., YOLOR, RetinaNet, SSD, YOLO, Mask RCNN, RFCN, Faster RCNN, Fast RCNN, SPPNet, RCNN, DPM, HOG detector), metrics, datasets, speed-up methods, and key technologies but also discusses the challenges that the community is currently facing, as well as how these detectors can be further improved and enhanced.

VI. References

[1] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005.

[2] D. Lowe, "Object recognition from local scale-invariant features," in Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 1999.

[3] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," International Journal of Computer Vision, pp. 91-110, 2004.

[4] S. Belongie, J. Malik and J. Puzicha, "Shape matching and object recognition using shape contexts," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, pp. 509-522, 2002.

[5] T. Malisiewicz, A. Gupta and A. A. Efros, "Ensemble of exemplar-SVMs for object detection and beyond," in International Conference on Computer Vision, Barcelona, Spain, 2011.

[6] P. Felzenszwalb, D. McAllester and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 2008.

[7] R. B. Girshick, P. F. Felzenszwalb and D. McAllester, "Object Detection with Grammar Models," Advances in Neural Information Processing Systems, vol. 24, pp. 442-450, 2011.

[8] R. B. Girshick, "From rigid templates to grammars: Object detection with structured models," 2012.

[9] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[10] J. Uijlings, K. v. d. Sande, T. Gevers and A. Smeulders, "Selective Search for Object Recognition," International Journal of Computer Vision, pp. 154-171, 2013.

[11] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional," Advances in neural information processing systems, vol. 25, pp. 1097-1105, 2012.

[12] R. Girshick, "Fast R-CNN," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015.

[13] 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, 2015.

[14] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in Computer Vision and Pattern Recognition, 2013.

[15] E. S. T. D. Jonathan Long, "Fully Convolutional Networks for Semantic Segmentation," in Computer Vision and Pattern Recognition, 2015.

[16] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[17] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in Computer Vision and Pattern Recognition, 2015.

[18] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn and A. Zisserman, "The PASCAL Visual Object Classes (VOC) Challenge," International Journal of Computer Vision, vol. 88, no. 2, pp. 303-338, 2009.

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," Miami, FL, USA, 2009.

[20] J. Dai, Y. Li, K. He and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," in CVPR, 2016.

[21] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," in IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017.

[22] S. Xie, R. Girshick, P. Dollár, Z. Tu and K. He, "Aggregated Residual Transformations for Deep Neural Networks," in Computer Vision and Pattern Recognition, 2017.

[23] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection".

[24] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017.

[25] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," in Computer Vision and Pattern Recognition, 2018.

[26] A. Bochkovskiy, C.-Y. Wang and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," in Computer Vision and Pattern Recognition, 2020.

[27] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg, "SSD: Single Shot MultiBox Detector," in Proceedings of the European Conference on Computer Vision (ECCV), 2016.

[28] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in Computer Vision and Pattern Recognition, 2015.

[29] D. Erhan, C. Szegedy, A. Toshev and D. Anguelov, "Scalable Object Detection using Deep Neural Networks," in Computer Vision and Pattern Recognition, 2013.

[30] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He and P. Doll, "Focal Loss for Dense Object Detection," in IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017.

[31] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan and S. Belongie, "Feature Pyramid Networks for Object Detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[32] C.-Y. Wang, I.-H. Yeh and H.-Y. M. Liao, "You Only Learn One Representation: Unified Network for Multiple Tasks," ArXiv, 2021.

[33] M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. Winn and A. Zisserman, "The PASCAL Visual Object Classes Challenge: A Retrospective," International Journal of Computer Vision, vol. 111, pp. 98-136, 2014.

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," International Journal of Computer Vision, vol. 115, pp. 211-252, 2015.

[35] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in European Conference on Computer Vision, 2014.

