Object Detection in 20 Years: A Survey

Zhengxia Zou*, Keyan Chen, Zhenwei Shi, *Member, IEEE*, Yuhong Guo, and Jieping Ye*, *Fellow, IEEE*

***Abstract*—Object detection, as of one the most fundamental and challenging problems in computer vision, has received great attention in recent years. Over the past two decades, we have seen a rapid technological evolution of object detection and its profound impact on the entire computer vision field. If we consider today's object detection technique as a revolution driven by deep learning, then back in the 1990s, we would see the ingenious thinking and long-term perspective design of early computer vision. This paper extensively reviews this fast-moving research field in the light of technical evolution, spanning over a quarter-century's time (from the 1990s to 2022). A number of topics have been covered in this paper, including the milestone detectors in history, detection datasets, metrics, fundamental building blocks of the detection system, speed-up techniques, and the recent state-of-the-art detection methods.**

***Index Terms*—Object detection, Computer vision, Deep learning, Convolutional neural networks, Technical evolution.**

I. Introduction

OBJECT detection is an important computer vision task that deals with detecting instances of visual objects of a certain class (such as humans, animals, or cars) in digital images. The goal of object detection is to develop computational models and techniques that provide one of the most basic pieces of knowledge needed by computer vision applications: *What objects are where?* The two most significant metrics for object detection are accuracy (including classification accuracy and localization accuracy) and speed.

Object detection serves as a basis for many other computer vision tasks, such as instance segmentation [1–4], image captioning [5–7], object tracking [8], etc. In recent years, the rapid development of deep learning techniques [9] has greatly promoted the progress of object detection, leading to remarkable breakthroughs and propelling it to a research hotspot with unprecedented attention. Object detection has now been widely used in many real-world applications, such as autonomous driving, robot vision, video surveillance, etc. Fig. 1 shows the growing number of publications that are associated with "object detection" over the past two decades.

The work was supported by the National Natural Science Foundation of China under Grant 62125102, the National Key Research and Development Program of China (Titled "Brain-inspired General Vision Models and Applications"), and the Fundamental Research Funds for the Central Universities. *(Corresponding Author: Zhengxia Zou (zhengxiazou@buaa.edu.cn) and Jieping Ye (jpye@umich.edu)).*

Zhengxia Zou is with the Department of Guidance, Navigation and Control, School of Astronautics, Beihang University, Beijing 100191, China, and also with Shanghai Artificial Intelligence Laboratory, Shanghai 200232, China.

Keyan Chen and Zhenwei Shi are with the Image Processing Center, School of Astronautics, and with the Beijing Key Laboratory of Digital Media, and with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China, and also with the Shanghai Artificial Intelligence Laboratory, Shanghai 200232, China.

Yuhong Guo is with the School of Computer Science, Carleton University, Ottawa, Ontario, K1S 5B6, Canada.

Jieping Ye is with the Alibaba Group, Hangzhou 310030, China.

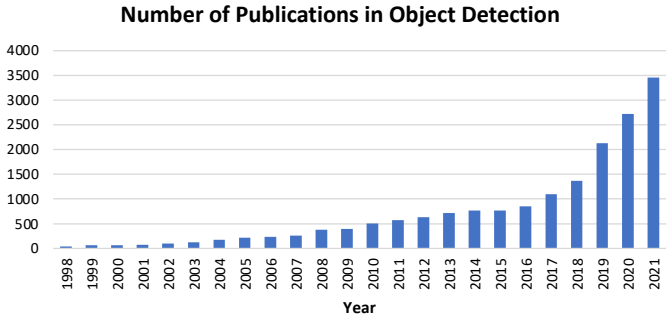

Fig. 1: The increasing number of publications in object detection from 1998 to 2021. (Data from Google scholar advanced search: *allintitle: "object detection" OR "detecting objects"*.)

As different detection tasks have totally different objectives and constraints, their difficulties may vary from each other. In addition to some common challenges in other computer vision tasks such as objects under different viewpoints, illuminations, and intraclass variations, the challenges in object detection include but are not limited to the following aspects: object rotation and scale changes (e.g., small objects), accurate object localization, dense and occluded object detection, speed up of detection, etc. In Sec. IV, we will give a more detailed analysis of these topics.

This survey seeks to provide novices with a complete grasp of object detection technology from many viewpoints, with an emphasis on its evolution. The key features are three-folds: A comprehensive review in the light of technical evolutions, an in-depth exploration of the key technologies and the recent state of the arts, and a comprehensive analysis of detection speed-up techniques. The main clue focuses on the past, present, and future, complemented with some other necessary components in object detection, like datasets, metrics, and acceleration techniques. Standing on the technical highway, this survey aims to present the evolution of related technologies, allowing readers to grasp the essential concepts and find potential future directions, while neglecting their technical specifics.

The rest of this paper is organized as follows. In Section II, we review the 20 years' evolution of object detection. In Section III, we review the speed-up techniques in object detection. The state-of-the-art detection methods of the recent three years are reviewed in Section IV. In Section V, we conclude this paper and make a deep analysis of the further research directions.

II. Object Detection in 20 Years

In this section, we will review the history of object detection from multiple views, including milestone detectors, datasets, metrics and the evolution of key techniques.

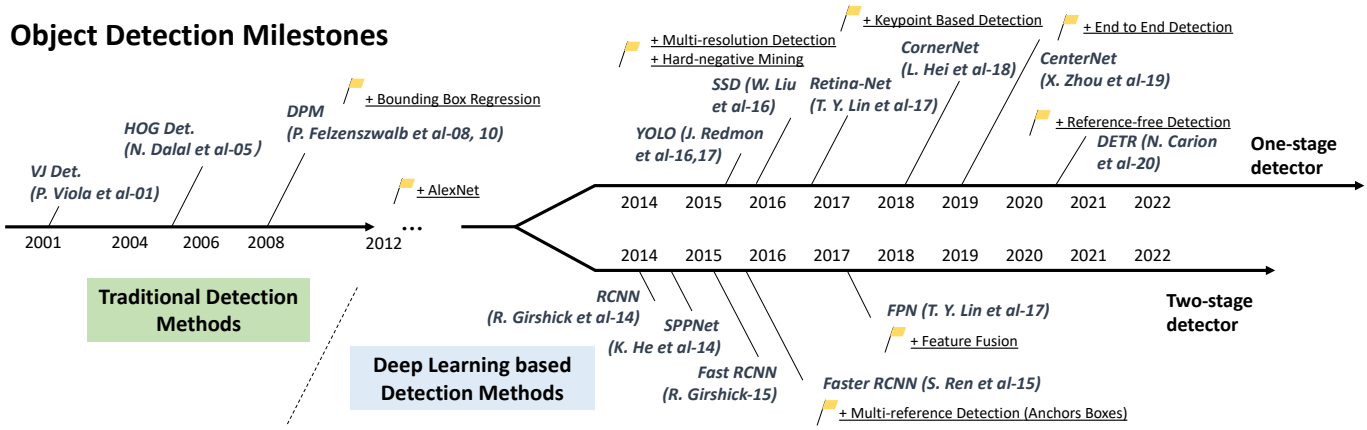

Fig. 2: A road map of object detection. Milestone detectors in this figure: VJ Det. [10, 11], HOG Det. [12], DPM [13–15], RCNN [16], SPPNet [17], Fast RCNN [18], Faster RCNN [19], YOLO [20–22], SSD [23], FPN [24], Retina-Net [25], CornerNet [26], CenterNet [27], DETR [28].

A. A Road Map of Object Detection

In the past two decades, it is widely accepted that the progress of object detection has generally gone through two historical periods: "traditional object detection period (before 2014)" and "deep learning based detection period (after 2014)", as shown in Fig. 2. In the following, we will summarize the milestone detectors of this period, with the emergence time and performance serving as the main clue to highlight the behind driving technology, seeing Fig. 3.

1) Milestones: Traditional Detectors: If we consider today's object detection technique as a revolution driven by deep learning, then back in the 1990s, we would see the ingenious design and long-term perspective of early computer vision. Most of the early object detection algorithms were built based on handcrafted features. Due to the lack of effective image representation at that time, people have to design sophisticated feature representations and a variety of speed-up skills.

Viola Jones Detectors: In 2001, P. Viola and M. Jones achieved real-time detection of human faces for the first time without any constraints (e.g., skin color segmentation) [10, 11]. Running on a 700MHz Pentium III CPU, the detector was tens or even hundreds of times faster than other algorithms in its time under comparable detection accuracy. The VJ detector follows a most straightforward way of detection, i.e., sliding windows: to go through all possible locations and scales in an image to see if any window contains a human face. Although it seems to be a very simple process, the calculation behind it was far beyond the computer's power of its time. The VJ detector has dramatically improved its detection speed by incorporating three important techniques: "integral image", "feature selection", and "detection cascades" (to be introduced in section III).

HOG Detector: In 2005, N. Dalal and B. Triggs proposed Histogram of Oriented Gradients (HOG) feature descriptor [12]. HOG can be considered as an important improvement of the scale-invariant feature transform [29, 30] and shape contexts [31] of its time. To balance the feature invariance (including translation, scale, illumination, etc) and the nonlinearity, the HOG descriptor is designed to be computed on a dense grid of uniformly spaced cells and use overlapping local contrast normalization (on "blocks"). Although HOG can be used to detect a variety of object classes, it was motivated primarily by the problem of pedestrian detection. To detect objects of different sizes, the HOG detector rescales the input image for multiple times while keeping the size of a detection window unchanged. The HOG detector has been an important foundation of many object detectors [13, 14, 32] and a large variety of computer vision applications for many years.

Deformable Part-based Model (DPM): DPM, as the winners of VOC-07, -08, and -09 detection challenges, was the epitome of the traditional object detection methods. DPM was originally proposed by P. Felzenszwalb [13] in 2008 as an extension of the HOG detector. It follows the detection philosophy of "divide and conquer", where the training can be simply considered as the learning of a proper way of decomposing an object, and the inference can be considered as an ensemble of detections on different object parts. For example, the problem of detecting a "car" can be decomposed to the detection of its window, body, and wheels. This part of the work, a.k.a. "star-model", was introduced by P. Felzenszwalb *et al.* [13]. Later on, R. Girshick has further extended the star model to the "mixture models" to deal with the objects in the real world under more significant variations and has made a series of other improvements [14, 15, 33, 34].

Although today's object detectors have far surpassed DPM in detection accuracy, many of them are still deeply influenced by its valuable insights, e.g., mixture models, hard negative mining, bounding box regression, context priming, etc. In 2010, P. Felzenszwalb and R. Girshick were awarded the "lifetime achievement" by PASCAL VOC.

2) Milestones: CNN based Two-stage Detectors: As the performance of hand-crafted features became saturated, the research of object detection reached a plateau after 2010. In 2012, the world saw the rebirth of convolutional neural networks [35]. As a deep convolutional network is able to learn robust and high-level feature representations of an image, a natural question arises: can we introduce it to object detection? R. Girshick *et al.* took the lead to break the deadlocks in

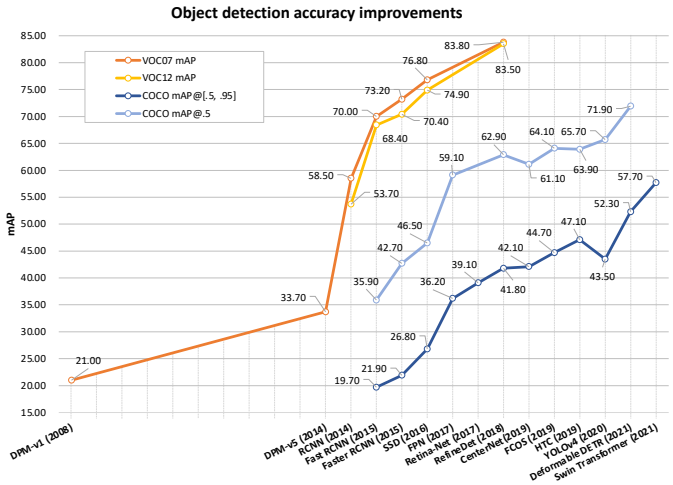

Fig. 3: Accuracy improvement of object detection on VOC07, VOC12 and MS-COCO datasets. Detectors in this figure: DPM-v1 [13], DPM-v5 [37], RCNN [16], SPPNet [17], Fast RCNN [18], Faster RCNN [19], SSD [23], FPN [24], RetinaNet [25], RefineDet [38], TridentNet [39] CenterNet [40], FCOS [41], HTC [42], YOLOv4 [22], Deformable DETR [43], Swin Transformer [44].

2014 by proposing the Regions with CNN features (RCNN) [16, 36]. Since then, object detection started to evolve at an unprecedented speed. There are two groups of detectors in the deep learning era: "two-stage detectors" and "one-stage detectors", where the former frames the detection as a "coarse-to-fine" process while the latter frames it as to "complete in one step".

RCNN: The idea behind RCNN is simple: It starts with the extraction of a set of object proposals (object candidate boxes) by selective search [45]. Then each proposal is rescaled to a fixed size image and fed into a CNN model pretrained on ImageNet (say, AlexNet [35]) to extract features. Finally, linear SVM classifiers are used to predict the presence of an object within each region and to recognize object categories. RCNN yields a significant performance boost on VOC07, with a large improvement of mean Average Precision (mAP) from 33.7% (DPM-v5 [46]) to 58.5%. Although RCNN has made great progress, its drawbacks are obvious: the redundant feature computations on a large number of overlapped proposals (over 2000 boxes from one image) lead to an extremely slow detection speed (14s per image with GPU). Later in the same year, SPPNet [17] was proposed and has solved this problem.

SPPNet: In 2014, K. He *et al.* proposed Spatial Pyramid Pooling Networks (SPPNet) [17]. Previous CNN models require a fixed-size input, e.g., a 224x224 image for AlexNet [35]. The main contribution of SPPNet is the introduction of a Spatial Pyramid Pooling (SPP) layer, which enables a CNN to generate a fixed-length representation regardless of the size of the image/region of interest without rescaling it. When using SPPNet for object detection, the feature maps can be computed from the entire image only once, and then fixed-length representations of arbitrary regions can be generated for training the detectors, which avoids repeatedly computing the convolutional features. SPPNet is more than 20 times faster than R-CNN without sacrificing any detection accuracy (VOC07 mAP=59.2%). Although SPPNet has effectively improved the detection speed, it still has some drawbacks: first, the training is still multi-stage, second, SPPNet only fine-tunes its fully connected layers while simply ignoring all previous layers. Later in the next year, Fast RCNN [18] was proposed and solved these problems.

Fast RCNN: In 2015, R. Girshick proposed Fast RCNN detector [18], which is a further improvement of R-CNN and SPPNet [16, 17]. Fast RCNN enables us to simultaneously train a detector and a bounding box regressor under the same network configurations. On VOC07 dataset, Fast RCNN increased the mAP from 58.5% (RCNN) to 70.0% while with a detection speed over 200 times faster than R-CNN. Although Fast-RCNN successfully integrates the advantages of R-CNN and SPPNet, its detection speed is still limited by the proposal detection (see Section II-C1 for more details). Then, a question naturally arises: "can we generate object proposals with a CNN model?" Later, Faster R-CNN [19] answered this question.

Faster RCNN: In 2015, S. Ren *et al.* proposed Faster RCNN detector [19, 47] shortly after the Fast RCNN. Faster RCNN is the first near-realtime deep learning detector (COCO mAP@.5=42.7%, VOC07 mAP=73.2%, 17fps with ZF-Net [48]). The main contribution of Faster-RCNN is the introduction of Region Proposal Network (RPN) that enables nearly cost-free region proposals. From R-CNN to Faster RCNN, most individual blocks of an object detection system, e.g., proposal detection, feature extraction, bounding box regression, etc, have been gradually integrated into a unified, end-to-end learning framework. Although Faster RCNN breaks through the speed bottleneck of Fast RCNN, there is still computation redundancy at the subsequent detection stage. Later on, a variety of improvements have been proposed, including RFCN [49] and Light head RCNN [50]. (See more details in Section III.)

Feature Pyramid Networks (FPN): In 2017, T.-Y. Lin *et al.* proposed FPN [24]. Before FPN, most of the deep learning based detectors run detection only on the feature maps of the networks' top layer. Although the features in deeper layers of a CNN are beneficial for category recognition, it is not conducive to localizing objects. To this end, a top-down architecture with lateral connections is developed in FPN for building high-level semantics at all scales. Since a CNN naturally forms a feature pyramid through its forward propagation, the FPN shows great advances for detecting objects with a wide variety of scales. Using FPN in a basic Faster R-CNN system, it achieves state-of-the-art single model detection results on the COCO dataset without bells and whistles (COCO mAP@.5=59.1%). FPN has now become a basic building block of many latest detectors.

3) Milestones: CNN based One-stage Detectors: Most of the two-stage detectors follow a coarse-to-fine processing paradigm. The coarse strives to improve recall ability, while the fine refines the localization on the basis of the coarse detection, and places more emphasis on the discriminate ability. They can easily attain a high precision without any bells and whistles, but rarely employed in engineering due to

the poor speed and enormous complexity. On the contrary, one-stage detectors can retrieve all objects in one-step inference. They are well-liked by mobile devices with real-time and easy-deployed features, but their performance suffers noticeably when detecting dense and small objects.

You Only Look Once (YOLO): YOLO was proposed by R. Joseph *et al.* in 2015. It was the first one-stage detector in the deep learning era [20]. YOLO is extremely fast: a fast version of YOLO runs at 155fps with VOC07 mAP=52.7%, while its enhanced version runs at 45fps with VOC07 mAP=63.4%. YOLO follows a totally different paradigm from two-stage detectors: to apply a single neural network to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region simultaneously. In spite of its great improvement of detection speed, YOLO suffers from a drop of localization accuracy compared with two-stage detectors, especially for some small objects. YOLO's subsequent versions [21, 22, 51] and the latter proposed SSD [23] has paid more attention to this problem. Recently, YOLOv7 [52], a follow-up work from YOLOv4 team, has been proposed. It outperforms most existing object detectors in terms of speed and accuracy (range from 5 FPS to 160 FPS) by introducing optimized structures like dynamic label assignment and model structure reparameterization.

Single Shot MultiBox Detector (SSD): SSD [23] was proposed by W. Liu *et al.* in 2015. The main contribution of SSD is the introduction of the multi-reference and multi-resolution detection techniques (to be introduced in Section II-C1), which significantly improves the detection accuracy of a one-stage detector, especially for some small objects. SSD has advantages in terms of both detection speed and accuracy (COCO mAP@.5=46.5%, a fast version runs at 59fps). The main difference between SSD and previous detectors is that SSD detects objects of different scales on different layers of the network, while the previous ones only run detection on their top layers.

RetinaNet: Despite its high speed and simplicity, the one-stage detectors have trailed the accuracy of two-stage detectors for years. T.-Y. Lin *et al.* have explored the reasons behind and proposed RetinaNet in 2017 [25]. They found that the extreme foreground-background class imbalance encountered during the training of dense detectors is the central cause. To this end, a new loss function named "focal loss" has been introduced in RetinaNet by reshaping the standard cross entropy loss so that detector will put more focus on hard, misclassified examples during training. Focal Loss enables the one-stage detectors to achieve comparable accuracy of two-stage detectors while maintaining a very high detection speed (COCO mAP@.5=59.1%).

CornerNet: Previous methods primarily used anchor boxes to provide classification and regression references. Objects frequently exhibit variation in terms of number, location, scale, ratio, etc. They have to follow the path of setting up a large number of reference boxes to better match ground truths in order to achieve high performance. However, the network would suffer from further category imbalance, lots of hand-designed hyper-parameters, and a long convergence time. To address these problems, H. Law *et al* [26] discard the previous detection paradigm, and view the task as a keypoint (corners of a box) prediction problem. After obtaining the key points, it will decouple and re-group the corner points using extra embedding information to form the bounding boxes. CornerNet outperforms most one-stage detectors at that time (COCO mAP@.5=57.8%).

CenterNet: X. Zhou *et al* proposed CenterNet [40] in 2019. It also follows a keypoint-based detection paradigm, but eliminates costly post-processes such as group-based keypoint assignment (in CornerNet [26], ExtremeNet [53], etc) and NMS, resulting in a fully end-to-end detection network. CenterNet considers an object to be a single point (the object's center) and regresses all of its attributes (such as size, orientation, location, pose, etc) based on the reference center point. The model is simple and elegant, and it can integrate 3-D object detection, human pose estimation, optical flow learning, depth estimation, and other tasks into a single framework. Despite using such a concise detection concept, CenterNet can also achieve comparative detection results (COCO mAP@.5=61.1%).

DETR: In recent years, Transformers have deeply affected the entire field of deep learning, particularly the field of computer vision. Transformers discard the traditional convolution operator in favor of attention-alone calculation in order to overcome the limitations of CNNs and obtain a global-scale receptive field. In 2020, N. Carion *et al* proposed DETR [28], where they viewed object detection as a set prediction problem and proposed an end-to-end detection network with Transformers. So far, object detection has entered a new era in which objects can be detected without the use of anchor boxes or anchor points. Later, X. Zhu *et al* proposed Deformable DETR [43] to address the DETR's long convergence time and limited performance on detecting small objects. It achieves state-of-the-art performance on MSCOCO dataset (COCO mAP@.5=71.9%).

B. Object Detection Datasets and Metrics

1) Datasets: Building larger datasets with less bias is essential for developing advanced detection algorithms. A number of well-known detection datasets have been released in the past 10 years, including the datasets of PASCAL VOC Challenges [54, 55] (e.g., VOC2007, VOC2012), ImageNet Large Scale Visual Recognition Challenge (e.g., ILSVRC2014) [56], MS-COCO Detection Challenge [57], Open Images Dataset [58, 59], Objects365 [60], etc. The statistics of these datasets are given in Table I. Fig. 4 shows some image examples of these datasets. Fig. 3 shows the improvements of detection accuracy on VOC07, VOC12 and MS-COCO datasets from 2008 to 2021.

Pascal VOC: The PASCAL Visual Object Classes (VOC) Challenges[1] (from 2005 to 2012) [54, 55] was one of the most important competitions in the early computer vision community. Two versions of Pascal-VOC are mostly used in object detection: VOC07 and VOC12, where the former consists of 5k tr. images + 12k annotated objects, and the latter consists of 11k tr. images + 27k annotated objects. 20 classes

[1]http://host.robots.ox.ac.uk/pascal/VOC/

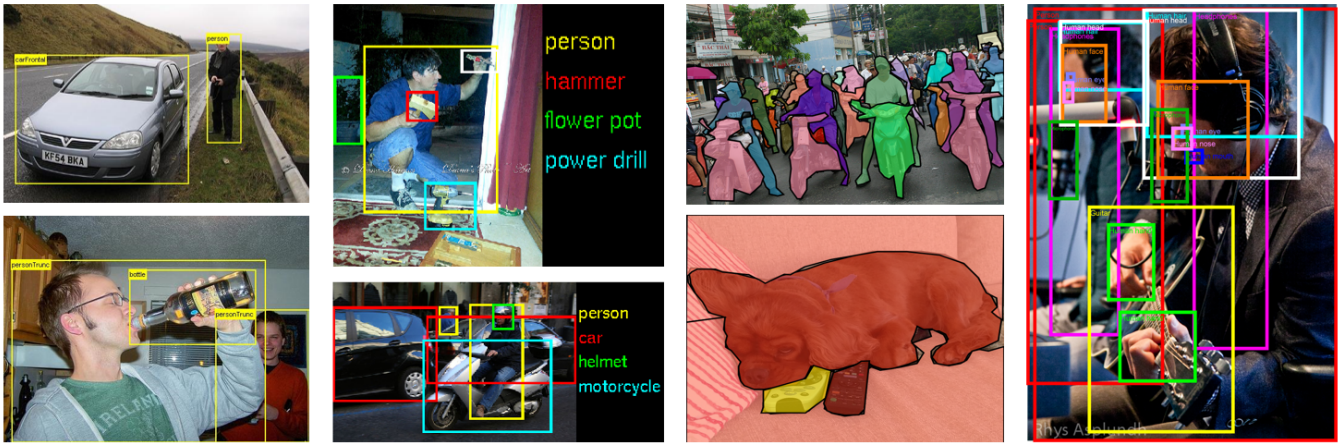

Fig. 4: Some example images and annotations in (a) PASCAL-VOC07, (b) ILSVRC, (c) MS-COCO, and (d) Open Images.

Dataset	train		validation		trainval		test	
	images	objects	images	objects	images	objects	images	objects
VOC-2007	2,501	6,301	2,510	6,307	5,011	12,608	4,952	14,976
VOC-2012	5,717	13,609	5,823	13,841	11,540	27,450	10,991	-
ILSVRC-2014	456,567	478,807	20,121	55,502	476,688	534,309	40,152	-
ILSVRC-2017	456,567	478,807	20,121	55,502	476,688	534,309	65,500	-
MS-COCO-2015	82,783	604,907	40,504	291,875	123,287	896,782	81,434	-
MS-COCO-2017	118,287	860,001	5,000	36,781	123,287	896,782	40,670	-
Objects365-2019	600,000	9,623,000	38,000	479,000	638,000	10,102,000	100,000	1,700,00
OID-2020	1,743,042	14,610,229	41,620	303,980	1,784,662	14,914,209	125,436	937,327

TABLE I: Some well-known object detection datasets and their statistics.

of objects that are common in life are annotated in these two datasets, e.g., "person", "cat", "bicycle", "sofa", etc.

ILSVRC: The ImageNet Large Scale Visual Recognition Challenge (ILSVRC)[2] [56] has pushed forward the state of the art in generic object detection. ILSVRC is organized each year from 2010 to 2017. It contains a detection challenge using ImageNet images [61]. The ILSVRC detection dataset contains 200 classes of visual objects. The number of its images/object instances is two orders of magnitude larger than VOC.

MS-COCO: MS-COCO[3] [57] is one of the most challenging object detection dataset available today. The annual competition based on MS-COCO dataset has been held since 2015. It has less number of object categories than ILSVRC, but more object instances. For example, MS-COCO-17 contains 164k images and 897k annotated objects from 80 categories. Compared with VOC and ILSVRC, the biggest progress of MS-COCO is that apart from the bounding box annotations, each object is further labeled using per-instance segmentation to aid in precise localization. In addition, MS-COCO contains more small objects (whose area is smaller than 1% of the image) and more densely located objects. Just like ImageNet in its time, MS-COCO has become the de facto standard for the object detection community.

Open Images: The year of 2018 sees the introduction of the Open Images Detection (OID) challenge[4] [62], following MS-COCO but at an unprecedented scale. There are two tasks in Open Images: 1) the standard object detection, and 2) the visual relationship detection which detects paired objects in particular relations. For the standard detection task, the dataset consists of 1,910k images with 15,440k annotated bounding boxes on 600 object categories.

2) Metrics: How can we evaluate the accuracy of a detector? This question may have different answers at different times. In the early time's detection research, there are no widely accepted evaluation metrics on detection accuracy. For example, in the early research of pedestrian detection [12], the "miss rate vs. false positives per window (FPPW)" was commonly used as the metric. However, the per-window measurement can be flawed and fails to predict full image performance [63]. In 2009, the Caltech pedestrian detection benchmark was introduced [63, 64] and since then, the evaluation metric has changed from FPPW to false positives per-image (FPPI).

In recent years, the most frequently used evaluation for detection is "Average Precision (AP)", which was originally introduced in VOC2007. AP is defined as the average detection precision under different recalls, and is usually evaluated in

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

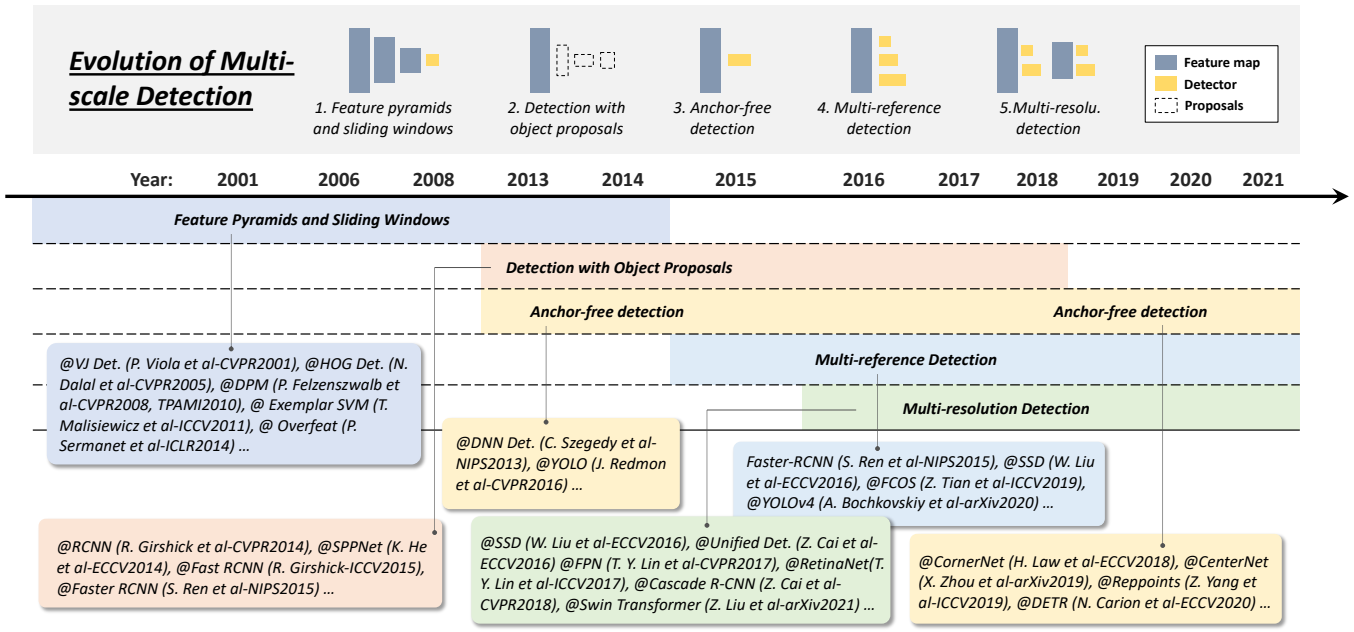

Fig. 5: Evolution of multi-scale detection techniques in object detection. Detectors in this figure: VJ Det. [10], HOG Det. [12], DPM [13], Exemplar SVM [32], Overfeat [65], RCNN [16], SPPNet [17], Fast RCNN [18], Faster RCNN [19], DNN Det. [66], YOLO [20], SSD [23], Unified Det. [67], FPN [24], RetinaNet [25], RefineDet [38], Cascade R-CNN [68], Swin Transformer [44], FCOS [41], YOLOv4 [22], CornerNet [26], CenterNet [40], Reppoints [69], DETR [28].

a category-specific manner. The mean AP (mAP) averaged over all categories is usually used as the final metric of performance. To measure the object localization accuracy, the IoU between the predicted box and the ground truth is used to verify whether it is greater than a predefined threshold, say, 0.5. If yes, the object will be identified as "detected", otherwise, "missed". The 0.5-IoU mAP has then become the de facto metric for object detection.

After 2014, due to the introduction of MS-COCO datasets, researchers started to pay more attention to the accuracy of object localization. Instead of using a fixed IoU threshold, MS-COCO AP is averaged over multiple IoU thresholds between 0.5 and 0.95, which encourages more accurate object localization and may be of great importance for some real-world applications (e.g., imagine there is a robot trying to grasp a spanner).

C. Technical Evolution in Object Detection

In this section, we will introduce some important building blocks of a detection system and their technical evolutions. First, we describe the multi-scale and context priming on model designing, followed by the sample selection strategy and the design of the loss function in the training process, and lastly, the Non-Maximum Suppression in the inference. The time-stamp in the chart and text is supplied by the publication time of papers. The evolution order shown in the figures is primarily to assist readers in understanding and there may be temporal overlap.

1) Technical Evolution of Multi-Scale Detection: Multi-scale detection of objects with "different sizes" and "different aspect ratios" is one of the main technical challenges in object detection. In the past 20 years, multi-scale detection has gone through multiple historical periods, as shown in Fig. 5.

Feature pyramids + sliding windows: After the VJ detector, researchers started to pay more attention to a more intuitive way of detection, i.e. by building "feature pyramid + sliding windows". From 2004, a number of milestone detectors were built based on this paradigm, including the HOG detector, DPM, etc. They frequently glide a fixed size detection window over the image, paying little attention to "different aspect ratios". To detect objects with a more complex appearance, R. Girshick *et al.* began to seek better solutions outside the feature pyramid. The "mixture model" [15] was a solution at that time, i.e. to train multiple detectors for objects of different aspect ratios. Apart from this, exemplar-based detection [32, 70] provided another solution by training individual models for every object instance (exemplar).

Detection with object proposals: Object proposals refer to a group of class-agnostic reference boxes that are likely to contain any objects. Detection with object proposals helps to avoid the exhaustive sliding window search across an image. We refer readers to the following papers for a comprehensive review on this topic [71, 72]. Early time's proposal detection methods followed a bottom-up detection philosophy [73, 74]. After 2014, with the popularity of deep CNN in visual recognition, the top-down, learning-based approaches began to show more advantages in this problem [19, 75, 76]. Now, the proposal detection gradually slipped out of sight after the rise of one-stage detectors.

Deep regression and anchor-free detection: In recent years, with the increase of GPU's computing power, multi-scale detection has become more and more straightforward

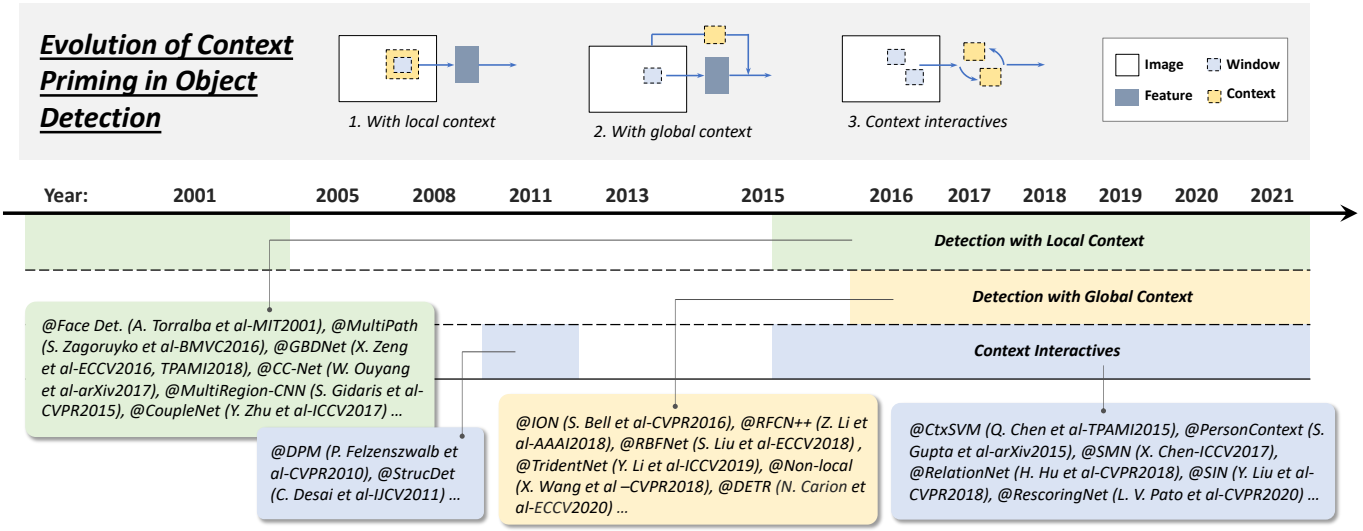

Fig. 6: Evolution of context priming in object detection. Detectors in this figure: Face Det. [78], MultiPath [79], GBDNet [80, 81], CC-Net [82], MultiRegion-CNN [83], CoupleNet [84], DPM [14, 15], StructDet [85], ION [86], RFCN++ [87], RBFNet [88], TridentNet [39], Non-local [89], DETR [28], CtxSVM [90], PersonContext [91], SMN [92], RelationNet [93], SIN [94], RescoringNet [95].

and brute-force. The idea of using the deep regression to solve multi-scale problems becomes simple, i.e., to directly predict the coordinates of a bounding box based on the deep learning features [20, 66]. After 2018, researchers began to think about the object detection problem from the perspective of keypoint detection. These methods often follow two ideas: One is the group-based method which detects keypoints (corners, centers, or representative points) and then conducts object-wise grouping [26, 53, 69, 77]; the other is the group-free method which regards an object as one/many points and then regresses the object attributes (size, ratio, etc.) under the reference of the points [40, 41].

Multi-reference/-resolution detection: Multi-reference detection is now the most used method for multi-scale detection [19, 22, 23, 41, 47, 51]. The main idea of multi-reference detection [19, 22, 23, 41, 47, 51] is to first define a set of references (a.k.a. anchors, including boxes and points) at every location of an image, and then predict the detection box based on these references. Another popular technique is multi-resolution detection [23, 24, 44, 67, 68], i.e. by detecting objects of different scales at different layers of the network. Multi-reference and multi-resolution detection have now become two basic building blocks in the state-of-the-art object detection systems.

2) Technical Evolution of Context Priming: Visual objects are usually embedded in a typical context with the surrounding environments. Our brain takes advantage of the associations among objects and environments to facilitate visual perception and cognition [96]. Context priming has long been used to improve detection. Fig. 6 shows the evolution of context priming in object detection.

Detection with local context: Local context refers to the visual information in the area that surrounds the object to detect. It has long been acknowledged that local context helps improve object detection. In the early 2000s, Sinha and Torralba [78] found that the inclusion of local contextual regions such as the facial bounding contour substantially improves face detection performance. Dalal and Triggs also found that incorporating a small amount of background information improves the accuracy of pedestrian detection [12]. Recent deep learning based detectors can also be improved with local context by simply enlarging the networks' receptive field or the size of object proposals [79–84, 97].

Detection with global context: Global context exploits scene configuration as an additional source of information for object detection. For early time detectors, a common way of integrating global context is to integrate a statistical summary of the elements that comprise the scene, like Gist [96]. For recent detectors, there are two methods to integrate the global context. The first method is to take advantage of deep convolution, dilated convolution, deformable convolution, pooling operation [39, 87, 88] to receive a large receptive field (even larger than the input image). But now, researchers have explored the potential to apply attention based mechanisms (non-local, transformers, etc.) to achieve a full-image receptive field and have obtained great success [28, 89]. The second method is to think of the global context as a kind of sequential information and to learn it with the recurrent neural networks [86, 98].

Context interactive: Context interactive refers to the constraints and dependencies that conveys between visual elements. Some recent researches suggested that modern detectors can be improved by considering context interactives. Some recent improvements can be grouped into two categories, where the first one is to explore the relationship between individual objects [15, 85, 90, 92, 93, 95], and the second one is to explore the dependencies between objects and scenes [91, 94].

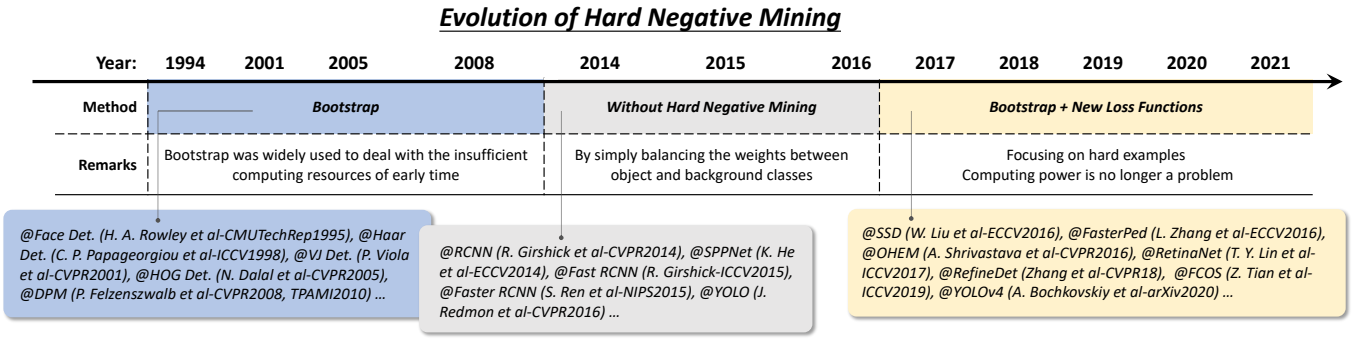

Fig. 7: Evolution of hard negative mining techniques in object detection. Detectors in this figure: Face Det. [99], Haar Det. [100], VJ Det. [10], HOG Det. [12], DPM [13, 15], RCNN [16], SPPNet [17], Fast RCNN [18], Faster RCNN [19], YOLO [20], SSD [23], FasterPed [101], OHEM [102], RetinaNet [25], RefineDet [38], FCOS [41], YOLOv4 [22].

3) Technical Evolution of Hard Negative Mining: The training of a detector is essentially an imbalanced learning problem. In the case of sliding window based detectors, the imbalance between backgrounds and objects could be as extreme as 10^7:1 [71]. In this case, using all backgrounds will be harmful to training as the vast number of easy negatives will overwhelm the learning process. Hard negative mining (HNM) aims to overcome this problem. The technical evolution of HNM is shown in Fig. 7.

Bootstrap: Bootstrap in object detection refers to a group of training techniques in which the training starts with a small part of background samples and then iteratively adds new miss-classified samples. In early times detectors, bootstrap was commonly used with the purpose of reducing the training computations over millions of backgrounds [10, 99, 100]. Later it became a standard technique in DPM and HOG detectors [12, 13] for solving the data imbalance problem.

HNM in deep learning based detectors: In the deep learning era, due to the increase of computing power, bootstrap was shortly discarded in object detection during 2014-2016 [16–20]. To ease the data-imbalance problem during training, detectors like Faster RCNN and YOLO simply balance the weights between the positive and negative windows. However, researchers later noticed this cannot completely solve the imbalanced problem [25]. To this end, the bootstrap was re-introduced to object detection after 2016 [23, 38, 101, 102]. An alternative improvement is to design new loss functions [25] by reshaping the standard cross entropy loss so that it will put more focus on hard, misclassified examples [25].

4) Technical Evolution of Loss Function: The loss function measures how well the model matches the data (i.e., the deviation of the predictions from the true labels). Calculating the loss yields the gradients of the model weights, which can subsequently be updated by backpropagation to better suit the data. Classification loss and localization loss make up the supervision of the object detection problem, seeing Eq. 1. A general form of the loss function can be written as follows:

$$L(p,p^*,t,t^*) = L_{cls.}(p,p^*) + \beta I(t) L_{loc.}(t,t^*)$$
$$I(t) = \begin{cases} 1 & \text{IoU}\{a,a^*\} > \eta \\ 0 & \text{else} \end{cases} \quad (1)$$

where t and t^* are the locations of predicted and ground-truth bounding boxes, p and p^* are their category probabilities. IoU$\{a,a^*\}$ is the IoU between the reference box/point a and its ground-truth a^*. η is an IoU threshold, say, 0.5. If an anchor box/point does not match any objects, its localization loss does not count in the final loss.

Classification loss: Classification loss is used to evaluate the divergence of the predicted category from the actual category, which was not thoroughly investigated in prevIoUs work such as YOLOv1 [20] and YOLOv2 [51] employing MSE/L2 loss (Mean Squared Error). Later, CE loss (Cross-Entropy) is typically used [21, 23, 47]. L2 loss is a measure in Euclidean space, whereas CE loss can measure distribution differences (termed as a form of likelihood). The prediction of classification is a probability, so CE loss is preferable to L2 loss with greater misclassification cost and lower gradient vanishing effect. For improving categorization efficiency, Label Smooth has been proposed to enhance the model generalization ability and solve the overconfidence problem on noise labels [103, 104], and Focal loss is designed to solve the problem of category imbalance and differences in classification difficulty [25].

Localization loss: Localization loss is used to optimize position and size deviation. L2 loss is prevalent in early research [16, 20, 51], but it is highly affected by outliers and prone to gradient explosion. Combining the benefits of L1 loss and L2 loss, the researchers propose Smooth L1 loss [18], as illustrated in the following formula,

$$\text{Smooth}_{L1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{else} \end{cases} \quad (2)$$

where x denotes the difference between the target and predicted values. When calculating the error, the above losses treat four numbers (x, y, w, h) representing a bounding box as independent variables, however, a correlation exists between them. Moreover, IoU is utilized to determine if the prediction box corresponds to the actual ground truth box in evaluation. Equal Smooth L1 values will have totally different IoU values, hence IoU loss [105] is introduced as follows:

$$\text{IoU loss} = -\log(\text{IoU}) \quad (3)$$

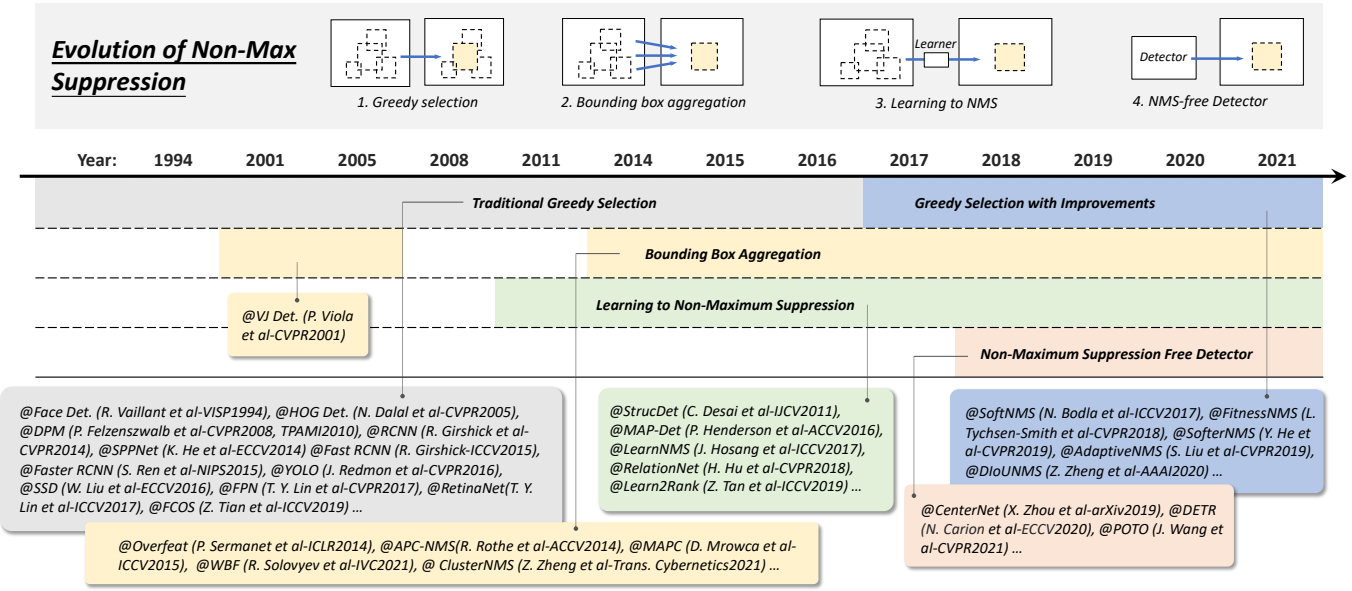

Fig. 8: Evolution of non-max suppression (NMS) techniques in object detection from 1994 to 2021: 1) Greedy selection, 2) Bounding box aggregation, 3) Learning to NMS, and 4) NMS-free detection. Detectors in this figure: Face Det. [108], HOG Det. [12], DPM [13, 15], RCNN [16], SPPNet [17], Fast RCNN [18], Faster RCNN [19], YOLO [20], SSD [23], FPN [24], RetinaNet [25], FCOS [41], StrucDet [85], MAP-Det [109], LearnNMS [110], RelationNet [93], Learn2Rank [111], SoftNMS [112], FitnessNMS [113], SofterNMS [114], AdaptiveNMS [115], DIoUNMS [107], Overfeat [65], APC-NMS [116], MAPC [117], WBF [118], ClusterNMS [119], CenterNet [40], DETR [28], POTO [120].

Following that, several algorithms improved IoU loss. G-IoU (Generalized IoU) [106] improved the case when IoU loss could not optimize the non-overlapping bounding boxes, i.e., IoU $= 0$. According to Distance-IoU [107], a successful detection regression loss should meet three geometric metrics: overlap area, center point distance, and aspect ratio. So, based on IoU loss and G-IoU loss, DIoU (Distance IoU) is defined as the distance between the center point of the prediction and the ground truth, and CIoU (Complete IoU) [107] considered the aspect ratio difference on the basis of DIoU.

5) Technical Evolution of Non-Maximum Suppression: As the neighboring windows usually have similar detection scores, the non-maximum suppression is used as a post-processing step to remove the replicated bounding boxes and obtain the final detection result. At early times of object detection, NMS was not always integrated [121]. This is because the desired output of an object detection system was not entirely clear at that time. Fig. 8 shows the evolution of NMS in the past 20 years.

Greedy selection: Greedy selection is an old-fashioned but the most popular way to perform NMS. The idea behind it is simple and intuitive: for a set of overlapped detections, the bounding box with the maximum detection score is selected while its neighboring boxes are removed according to a predefined overlap threshold. Although greedy selection has now become the de facto method for NMS, it still has some space for improvement. First, the top-scoring box may not be the best fit. Second, it may suppress nearby objects. Finally, it does not suppress false positives [116]. Many works have been proposed to solve the problems mentioned above [107, 112, 114, 115].

Bounding Box aggregation: BB aggregation is another group of techniques for NMS [10, 65, 116, 117] with the idea of combining or clustering multiple overlapped bounding boxes into one final detection. The advantage of this type of method is that it takes full consideration of object relationships and their spatial layout [118, 119]. Some well-known detectors use this method, such as the VJ detector [10] and the Overfeat (winner of ILSVRC-13 localization task) [65].

Learning based NMS: A recent group of NMS improvements that have recently received much attention is learning based NMS [85, 93, 109–111, 122]. The main idea is to think of NMS as a filter to re-score all raw detections and to train the NMS as part of a network in an end-to-end fashion or train a net to imitate NMS's behavior. These methods have shown promising results in improving occlusion and dense object detection over traditional hand-crafted NMS methods.

NMS-free detector: To release from NMS and achieve a fully end-to-end object detection training network, researchers developed a series of methods to complete one-to-one label assignment (a.k.a. one object with just one prediction box) [28, 40, 120]. These methods frequently adhere to a rule that calls for the use of the highest-quality box for training in order to achieve free NMS. NMS-free detectors are more similar to the human visual perception system and are also a possible way to the future of object detection.

III. Speed-Up of Detection

The acceleration of a detector has long been a challenging problem. The speed-up techniques in object detection can be divided into three levels of groups: speed up of "detection

pipeline", "detector backbone", and "numerical computation". , as shown in Fig. 9. Refer to [123] for a more detailed version.

A. Feature Map Shared Computation

Among the different computational stages of a detector, feature extraction usually dominates the amount of computation. The most commonly used idea to reduce the feature computational redundancy is to compute the feature map of the whole image only once [18, 19, 124], which have achieved tens or even hundreds of times of acceleration.

B. Cascaded Detection

Cascaded detection is a commonly used technique [10, 125]. It takes a coarse to fine detection philosophy: to filter out most of the simple background windows using simple calculations, then to process those more difficult windows with complex ones. In recent years, cascaded detection has been especially applied to those detection tasks of "small objects in large scenes", e.g., face detection [126, 127], pedestrian detection [101, 124, 128], etc.

C. Network Pruning and Quantification

"Network pruning" and "network quantification" are two commonly used methods to speed up a CNN model. The former refers to pruning the network structure or weights and the latter refers to reducing their code length. The research of "network pruning" can be traced back to as early as the 1980s [129]. The recent network pruning methods usually take an iterative training and pruning process, i.e., to remove only a small group of unimportant weights after each stage of training, and to repeat those operations [130]. The recent works on network quantification mainly focus on network binarization, which aims to compress a network by quantifying its activations or weights to binary variables (say, 0/1) so that the floating-point operation is converted to logical operations.

D. Lightweight Network Design

The last group of methods to speed up a CNN based detector is to directly design lightweight networks. In addition to some general designing principles like "fewer channels and more layers" [131], some other methods have been proposed in recent years [132–136].

1) Factorizing Convolutions: Factorizing convolutions is the most straightforward way to build a lightweight CNN model. There are two groups of factorizing methods. The first group is to factorize a large convolution filter into a set of small ones [50, 87, 137], as shown in Fig. 10 (b). For example, one can factorize a 7x7 filter into three 3x3 filters, where they share the same receptive field but the latter one is more efficient. The second group is to factorize convolutions in their channel dimension [138, 139], as shown in Fig. 10 (c).

2) Group Convolution: Group convolution aims to reduce the number of parameters in a convolution layer by dividing the feature channels into different groups, and then convolve on each group independently [140, 141], as shown in Fig. 10 (d). If we evenly divide the features into m groups, without changing other configurations, the computation will be theoretically reduced to $1/m$ of that before.

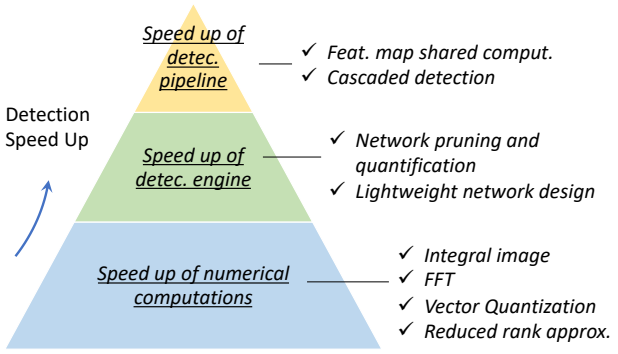

Fig. 9: An overview of the speed-up techniques in object detection.

3) Depth-wise Separable Convolution: Depth-wise separable convolution [142], as shown in Fig. 10 (e) can be viewed as a special case of the group convolution when the number of groups is set equal to the number of channels. Usually, a number of 1x1 filters are used to make a dimension transform so that the final output will have the desired number of channels. By using depth-wise separable convolution, the computation can be reduced from $\mathcal{O}(dk^2c)$ to $\mathcal{O}(ck^2)+\mathcal{O}(dc)$. This idea has been recently applied to object detection and fine-grain classification [143–145].

4) Bottle-neck Design: A bottleneck layer in a neural network contains few nodes compared to the previous layers. In recent years, the bottle-neck design has been widely used for designing lightweight networks [50, 133, 146–148]. Among these methods, the input layer of a detector can be compressed to reduce the amount of computation from the very beginning of the detection [133, 146, 147]. One can also compress the feature map to make it thinner, so that to speed up subsequent detection [50, 148].

5) Detection with NAS: Deep learning-based detectors are becoming increasingly sophisticated, relying heavily on handcrafted network architecture and training parameters. Neural architecture search (NAS) is primarily concerned with defining the proper space of candidate networks, improving strategies for searching quickly and accurately, and validating the searching results at a low cost. When designing a detection model, NAS can reduce the need for human intervention on the design of the network backbone and anchor boxes [149–155].

E. Numerical Acceleration

Numerical Acceleration aims to accelerate object detectors from the bottom of their implementations.

1) Speed Up with Integral Image: The integral image is an important method in image processing. It helps to rapidly calculate summations over image sub-regions. The essence of integral image is the integral-differential separability of convolution in signal processing:

$$f(x) * g(x) = (\int f(x)dx) * (\frac{dg(x)}{dx}), \tag{4}$$

where if $dg(x)/dx$ is a sparse signal, then the convolution can be accelerated by the right part of this equation [10, 156].

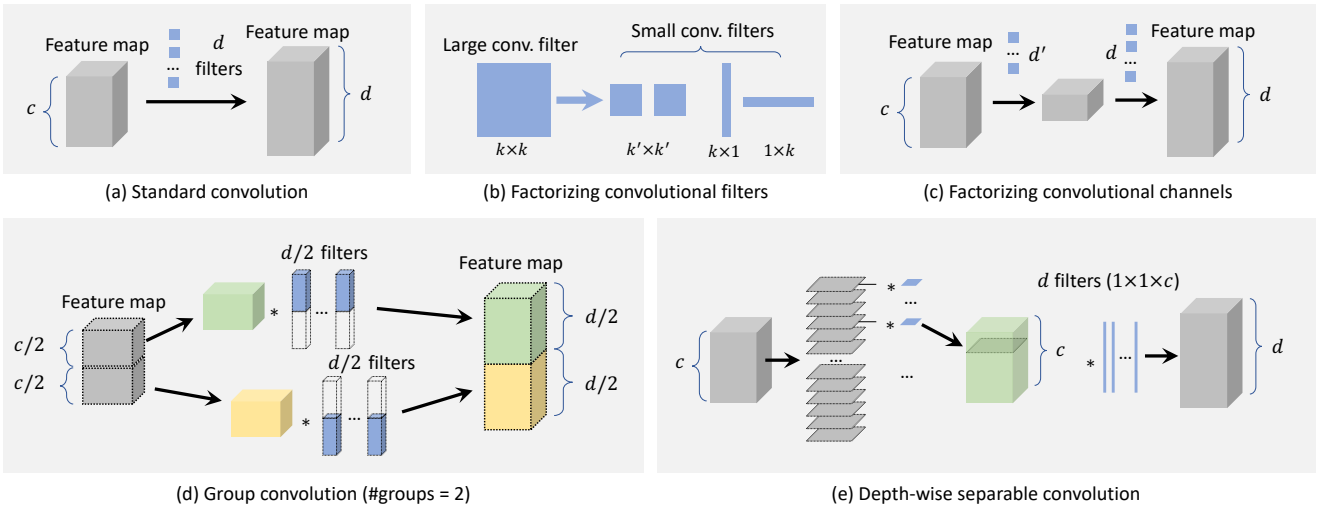

Fig. 10: An overview of speed up methods of a CNN's convolutional layer and the comparison of their computational complexity: (a) Standard convolution: $\mathcal{O}(dk^2c)$. (b) Factoring convolutional filters ($k\times k \rightarrow (k^{'}\times k^{'})^2$ or $1\times k, k\times 1$): $\mathcal{O}(dk^{'2}c)$ or $\mathcal{O}(dkc)$. (c) Factoring convolutional channels: $\mathcal{O}(d'k^2c)+\mathcal{O}(dk^2d')$. (d) Group convolution (#groups=m): $\mathcal{O}(dk^2c/m)$. (e) Depth-wise separable convolution: $\mathcal{O}(ck^2)+\mathcal{O}(dc)$.

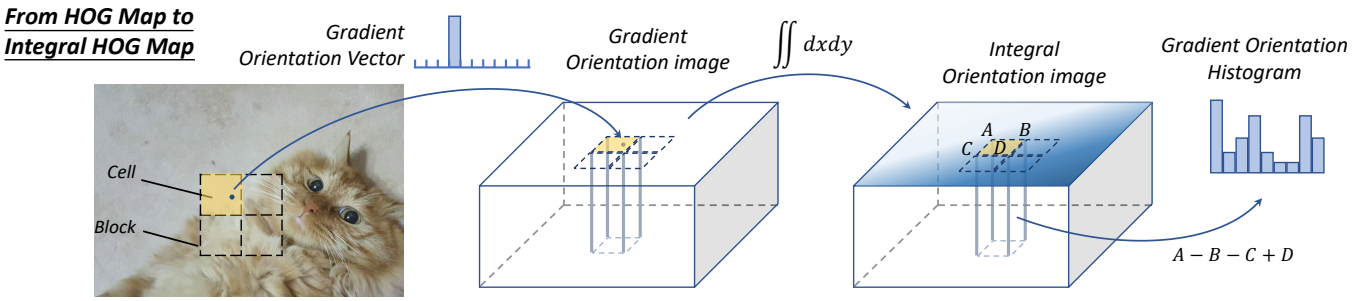

Fig. 11: An illustration of how to compute the "Integral HOG Map" [124]. With integral image techniques, we can efficiently compute the histogram feature of any location and any size with constant computational complexity.

The integral image can also be used to speed up more general features in object detection, e.g., color histogram, gradient histogram [124, 157–159], etc. A typical example is to speed up HOG by computing integral HOG maps [124, 157], as shown in Fig. 11. Integral HOG map has been used in pedestrian detection and has achieved dozens of times' acceleration without losing any accuracy [124].

2) Speed Up in Frequency Domain: Convolution is an important type of numerical operation in object detection. As the detection of a linear detector can be viewed as the window-wise inner product between the feature map and detector's weights, which can be implemented by convolutions. The Fourier transform is a very practical way to speed up convolutions, where the theoretical basis is the convolution theorem in signal processing, i.e. under suitable conditions, the Fourier transform F of a convolution of two signals $I*W$ is the point-wise product in their Fourier space:

$$I * W = F^{-1}(F(I) \odot F(W)) \tag{5}$$

where F is Fourier transform, F^{-1} is Inverse Fourier transform, and $\odot$ is the point-wise product. The above calculation can be accelerated by using the Fast Fourier Transform (FFT) and the Inverse FFT (IFFT) [160–163].

3) Vector Quantization: The Vector Quantization (VQ) is a classical quantization method in signal processing that aims to approximate the distribution of a large group of data by a small set of prototype vectors. It can be used for data compression and accelerating the inner product operation in object detection [164, 165].

IV. Recent Advances in Object Detection

The continual appearance of new technologies over the past two decades has a considerable influence on object detection, while its fundamental principles and underlying logic have remained unchanged. In the above sections, we introduced the evolution of technology over the past two decades in a large-scale time range to help readers comprehend object detection; in this section, we will focus more on state-of-the-art algorithms in recent years on a short time range to help readers understand object detection. Some are expansions of previously discussed techniques (e.g., Sec. IV-A – IV-E), while others are novel crossovers that mix concepts (e.g., Sec. IV-F – IV-H).

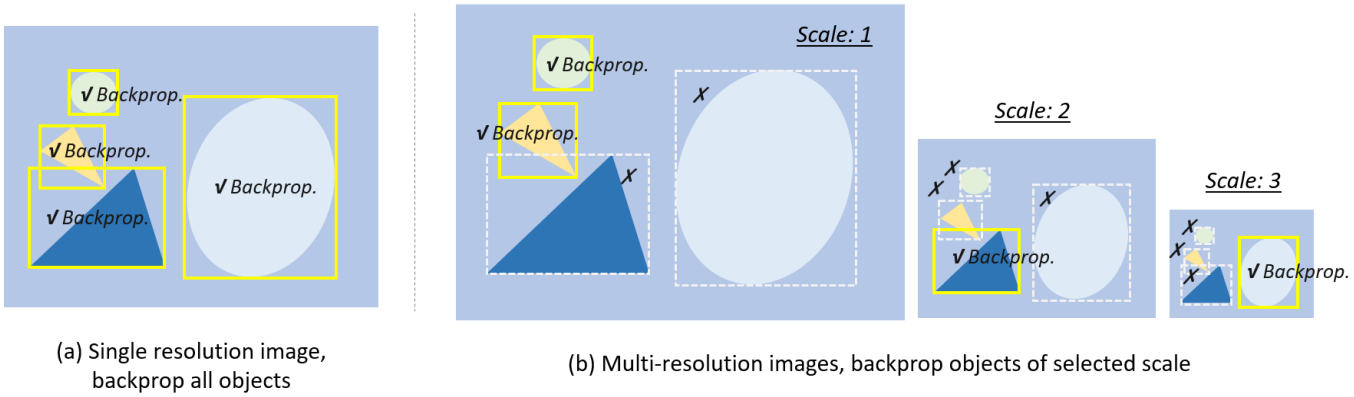

Fig. 12: Different training strategies for multi-scale object detection: (a): Training on a single resolution image, back propagate objects of all scales [17–19, 23]. (b) Training on multi-resolution images (image pyramid), back propagate objects of selected scale. If an object is too large or too small, its gradient will be discarded [39, 176, 177].

A. Beyond Sliding Window Detection

Since an object in an image can be uniquely determined by its upper left corner and lower right corner of the ground truth box, the detection task, therefore, can be equivalently framed as a pair-wise key points localization problem. One recent implementation of this idea is to predict a heat-map for the corners [26]. Some other methods follow the idea and utilize more key points (corner and center [77], extreme and center points [53], representative points [69]) to obtain better performance. Another paradigm views an object as a point/points and directly predicts the object's attributes (e.g. height and width) without grouping. The advantage of this approach is that it can be implemented under a semantic segmentation framework, and there is no need to design multi-scale anchor boxes. Furthermore, by viewing object detection as a set prediction, DETR [28, 43] completely liberates it in a reference-based framework.

B. Robust Detection of Rotation and Scale Changes

In recent years, efforts have been made on robust detection of rotation and scale changes.

1) Rotation Robust Detection: Object rotation is common to see in face detection, text detection, and remote sensing object detection. The most straightforward solution to this problem is to perform data augmentation so that an object in any orientation can be well covered by the augmented data distribution [166], or to train independent detectors separately for each orientation [167, 168]. Designing rotation invariant loss functions is a recent popular solution, where a constraint on the detection loss is added so that the feature of rotated objects keeps unchanged [169–171]. Another recent solution is to learn geometric transformations of the objects candidates [172–175]. In two-stage detectors, ROI pooling aims to extract a fixed-length feature representation for an object proposal with any location and size. Since the feature pooling usually is performed in Cartesian coordinates, it is not invariant to rotation transform. A recent improvement is to perform ROI pooling in polar coordinates so that the features can be robust to the rotation changes [167].

2) Scale Robust Detection: Recent studies have been made for scale robust detection at both training and detection stages.

Scale adaptive training: Modern detectors usually re-scale input images to a fixed size and back propagate the loss of the objects in all scales. A drawback of doing this is there will be a "scale imbalance" problem. Building an image pyramid during detection could alleviate this problem but not fundamentally [49, 178]. A recent improvement is Scale Normalization for Image Pyramids (SNIP) [176], which builds image pyramids at both training and detection stages and only backpropagates the loss of some selected scales, as shown in Fig. 12. Some researchers have further proposed a more efficient training strategy: SNIP with Efficient Resampling (SNIPER) [177], i.e. to crop and re-scale an image to a set of sub-regions so that to benefit from large batch training.

Scale adaptive detection: In CNN based detectors, the size of and aspect ratio of anchors are usually carefully designed. A drawback of doing this is the configurations cannot be adaptive to unexpected scale changes. To improve the detection of small objects, some "adaptive zoom-in" techniques are proposed in some recent detectors to adaptively enlarge the small objects into the "larger ones" [179, 180]. Another recent improvement is to predict the scale distribution of objects in an image, and then adaptively re-scaling the image according to it [181, 182].

C. Detection with Better Backbones

The accuracy/speed of a detector depends heavily on the feature extraction networks, a.k.a, backbones, e.g. the ResNet [178], CSPNet [183], Hourglass [184], and Swin Transformer [44]. For a detailed introduction of some important detection backbones in deep learning era, we refer readers to the following surveys [185]. Fig. 13 shows the detection accuracy of three well-known detection systems: Faster RCNN [19], R-FCN [49] and SSD [23] with different backbones [186]. Object detection has recently benefited from the powerful feature extraction capabilities of Transformers. On the COCO dataset, the top-10 detection methods are all transformer-based [5]. The

[5]https://paperswithcode.com/sota/object-detection-on-coco

performance gap between Transformers and CNNs have been gradually widened.

D. Improvements of Localization

To improve localization accuracy, there are two groups of methods in recent detectors: 1) bounding box refinement, and 2) new loss functions for accurate localization.

1) Bounding Box Refinement: The most intuitive way to improve localization accuracy is bounding box refinement, which can be considered as a post-processing of the detection results. One recent method is to iteratively feed the detection results into a BB regressor until the prediction converges to a correct location and size [187–189]. However, some researchers also claimed that this method does not guarantee the monotonicity of localization accuracy [187] and may degenerate the localization if the refinement is applied for multiple times.

2) New Loss Functions for Accurate Localization: In most modern detectors, object localization is considered as a coordinate regression problem. However, the drawbacks of this paradigm are obvious. First, the regression loss does not correspond to the final evaluation of localization, especially for some objects with very large aspect ratios. Second, the traditional BB regression method does not provide the confidence of localization. When there are multiple BB's overlapping with each other, this may lead to failure in non-maximum suppression. The above problems can be alleviated by designing new loss functions. The most intuitive improvement is to directly use IoU as the localization loss [105–107, 190]. Besides, some researchers also tried to improve localization under a probabilistic inference framework [191]. Different from the previous methods that directly predict the box coordinates, this method predicts the probability distribution of a bounding box location.

E. Learning with Segmentation Loss

Object detection and semantic segmentation are two fundamental tasks in computer vision. Recent researches suggest object detection can be improved by learning with semantic segmentation losses.

To improve detection with segmentation, the simplest way is to think of the segmentation network as a fixed feature extractor and to integrate it into a detector as auxiliary features [83, 192, 193]. The advantage of this approach is that it is easy to implement, while the disadvantage is that the segmentation network may bring additional computation.

Another way is to introduce an additional segmentation branch on top of the original detector and to train this model with multi-task loss functions (seg. + det.) [4, 42, 192]. The advantage is the seg. brunch will be removed at the inference stage and the detection speed will not be affected. However, the disadvantage is that the training requires pixel-level image annotations.

F. Adversarial Training

The Generative Adversarial Networks (GAN) [194], introduced by A. Goodfellow *et al.* in 2014, has received great attention in many tasks such as image generation[194, 195], image style transfer [196], and image super-resolution [197].

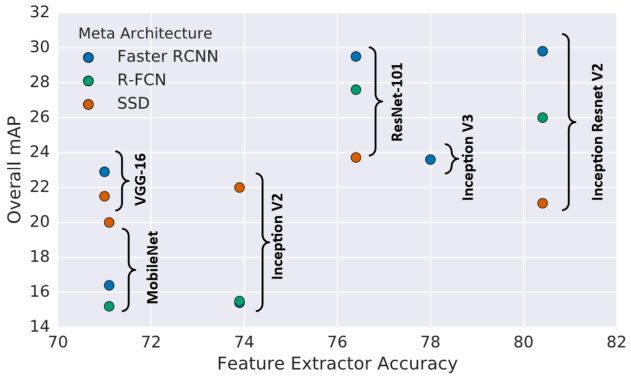

Fig. 13: A comparison of detection accuracy of three detectors: Faster RCNN [19], R-FCN [49] and SSD [23] on MS-COCO dataset with different detection backbones. Image from J. Huang *et al.* CVPR 2017 [186].

Recently, adversarial training has also been applied to object detection, especially for improving the detection of the small and occluded objects. For small object detection, GAN can be used to enhance the features of small objects by narrowing the representations between small and large ones [198, 199]. To improve the detection of occluded objects, one recent idea is to generate occlusion masks by using adversarial training [200]. Instead of generating examples in pixel space, the adversarial network directly modifies the features to mimic occlusion.

G. Weakly Supervised Object Detection

Training a deep learning based object detector usually requires a large amount of manually labeled data. Weakly Supervised Object Detection (WSOD) aims at easing the reliance on data annotation by training a detector with only image-level annotations instead of bounding boxes [201].

Multi-instance learning is a group of supervised learning algorithms that has seen widespread application in WSOD [202–209]. Instead of learning with a set of instances which are individually labeled, a multi-instance learning model receives a set of labeled bags, each containing many instances. If we consider object candidates in an image as a bag and image-level annotation as the label, then the WSOD can be formulated as a multi-instance learning process.

Class activation mapping is another recent group of methods for WSOD [210, 211]. The research on CNN visualization has shown that the convolutional layer of a CNN behaves as object detectors despite there is no supervision on the location of the object. Class activation mapping shed light on how to enable a CNN with localization capability despite being trained on image-level labels [212].

In addition to the above approaches, some other researchers considered the WSOD as a proposal ranking process by selecting the most informative regions and then training these regions with image-level annotation [213]. Some other researchers proposed to mask out different parts of the image. If the detection score drops sharply, then the masked region may

contain an object with high probability [214]. More recently, generative adversarial training has also been used for WSOD [215].

H. Detection with Domain Adaptation

The training process of most object detectors can be essentially viewed as a likelihood estimation process under the assumption of independent and identically distributed (i.i.d.) data. Object detection with non-i.i.d. data, especially for some real-world applications, still remains a challenge. Aside from collecting more data or applying proper data augmentation, domain adaptation offers the possibility of narrowing the gap between domains. To obtain domain-invariant feature representation, feature regularization and adversarial training based methods have been explored at the image, category, or object levels [216–221]. Cycle-consistent transformation [222] has also been applied to bridge the gap between source and target domain [223, 224]. Some other methods also incorporate both ideas [225] to acquire better performance.

V. Conclusion and Future Directions

Remarkable achievements have been made in object detection over the past 20 years. This paper extensively reviews some milestone detectors, key technologies, speed-up methods, datasets, and metrics in its 20 years of history. Some promising future directions may include but are not limited to the following aspects to help readers get more insights beyond the scheme mentioned above.

Lightweight object detection: Lightweight object detection aims to speed up the detection inference to run on low-power edge devices. Some important applications include mobile augmented reality, automatic driving, smart city, smart cameras, face verification, etc. Although a great effort has been made in recent years, the speed gap between a machine and human eyes still remains large, especially for detecting some small objects or detecting with multi-source information [226, 227].

End-to-End object detection: Although some methods have been developed to detect objects in a fully end-to-end manner (image to box in a network) using one-to-one label assignment training, the majority still use a one-to-many label assignment method where the non-maximum suppression operation is separately designed. Future research on this topic may focus on designing end-to-end pipelines that maintain both high detection accuracy and efficiency [228].

Small object detection: Detecting small objects in large scenes has long been a challenge. Some potential application of this research direction includes counting the population of people in crowd or animals in the open air and detecting military targets from satellite images. Some further directions may include the integration of the visual attention mechanisms and the design of high resolution lightweight networks [229, 230].

3D object detection: Despite recent advances in 2-D object detection, applications like autonomous driving rely on access to the objects' location and pose in a 3D world. The future of object detection will receive more attention in the 3D world and the utilization of multi-source and multi-view data (e.g., RGB images and 3D lidar points from multiple sensors) [231, 232].

Detection in videos: Real-time object detection/tracking in HD videos is of great importance for video surveillance and autonomous driving. Traditional object detectors are usually designed under for image-wise detection, while simply ignores the correlations between videos frames. Improving detection by exploring the spatial and temporal correlation under the calculation limitation is an important research direction [233, 234].

Cross-modality detection: Object detection with multiple sources/modalities of data, e.g., RGB-D image, lidar, flow, sound, text, video, etc, is of great importance for a more accurate detection system which performs like human-being's perception. Some open questions include: how to immigrate well-trained detectors to different modalities of data, how to make information fusion to improve detection, etc [235, 236].

Towards open-world detection: Out-of-domain generalization, zero-shot detection, and incremental detection are emerging topics in object detection. The majority of them devised ways to reduce catastrophic forgetting or utilized supplemental information. Humans have an instinct to discover objects of unknown categories in the environment. When the corresponding knowledge (label) is given, humans will learn new knowledge from it, and get to keep the patterns. However, it is difficult for current object detection algorithms to grasp the detection ability of unknown classes of objects. Object detection in the open world aims at discovering unknown categories of objects when supervision signals are not explicitly given or partially given, which holds great promise in applications such as robotics and autonomous driving [237, 238].

Standing on the highway of technical evolutions, we believe this paper will help readers to build a complete road map of object detection and to find future directions of this fast-moving research field.

References

[1] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *ECCV*. Springer, 2014, pp. 297–312.

[2] ——, "Hypercolumns for object segmentation and fine-grained localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 447–456.

[3] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *CVPR*, 2016, pp. 3150–3158.

[4] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*. IEEE, 2017, pp. 2980–2988.

[5] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *CVPR*, 2015, pp. 3128–3137.

[6] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *ICML*, 2015, pp. 2048–2057.

[7] Q. Wu, C. Shen, P. Wang, A. Dick, and A. van den Hengel, "Image captioning and visual question an-

swering based on attributes and external knowledge," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 6, pp. 1367–1381, 2018.

[8] K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang, Z. Wang, R. Wang, X. Wang *et al.*, "T-cnn: Tubelets with convolutional neural networks for object detection from videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 2896–2907, 2018.

[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[10] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *CVPR*, vol. 1. IEEE, 2001, pp. I–I.

[11] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, vol. 1. IEEE, 2005, pp. 886–893.

[13] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *CVPR*. IEEE, 2008, pp. 1–8.

[14] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, "Cascade object detection with deformable part models," in *CVPR*. IEEE, 2010, pp. 2241–2248.

[15] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014, pp. 580–587.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *ECCV*. Springer, 2014, pp. 346–361.

[18] R. Girshick, "Fast r-cnn," in *ICCV*, 2015, pp. 1440–1448.

[19] 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*, 2015, pp. 91–99.

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016, pp. 779–788.

[21] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

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

[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *ECCV*. Springer, 2016, pp. 21–37.

[24] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection." in *CVPR*, vol. 1, no. 2, 2017, p. 4.

[25] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[26] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 734–750.

[27] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[28] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision*. Springer, 2020, pp. 213–229.

[29] D. G. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, vol. 2. Ieee, 1999, pp. 1150–1157.

[30] ——, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[31] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," CALIFORNIA UNIV SAN DIEGO LA JOLLA DEPT OF COMPUTER SCIENCE AND ENGINEERING, Tech. Rep., 2002.

[32] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-svms for object detection and beyond," in *ICCV*. IEEE, 2011, pp. 89–96.

[33] R. B. Girshick, P. F. Felzenszwalb, and D. A. Mcallester, "Object detection with grammar models," in *Advances in Neural Information Processing Systems*, 2011, pp. 442–450.

[34] R. B. Girshick, *From rigid templates to grammars: Object detection with structured models*. Citeseer, 2012.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[36] 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, 2016.

[37] M. A. Sadeghi and D. Forsyth, "30hz object detection with dpm v5," in *ECCV*. Springer, 2014, pp. 65–79.

[38] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," in *CVPR*, 2018.

[39] Y. Li, Y. Chen, N. Wang, and Z. Zhang, "Scale-aware trident networks for object detection," *arXiv preprint arXiv:1901.01892*, 2019.

[40] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.

[41] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proceedings of the IEEE/CVF international conference on computer*

vision, 2019, pp. 9627–9636.

[42] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang *et al.*, "Hybrid task cascade for instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4974–4983.

[43] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[44] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *arXiv preprint arXiv:2103.14030*, 2021.

[45] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[46] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester, "Discriminatively trained deformable part models, release 5," http://people.cs.uchicago.edu/ rbg/latent-release5/.

[47] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 6, pp. 1137–1149, 2017.

[48] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *ECCV*. Springer, 2014, pp. 818–833.

[49] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379–387.

[50] 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 preprint arXiv:1711.07264*, 2017.

[51] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *arXiv preprint*, 2017.

[52] 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 preprint arXiv:2207.02696*, 2022.

[53] X. Zhou, J. Zhuo, and P. Krahenbuhl, "Bottom-up object detection by grouping extreme and center points," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 850–859.

[54] M. Everingham, L. Van Gool, C. K. 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, 2010.

[55] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International journal of computer vision*, vol. 111, no. 1, pp. 98–136, 2015.

[56] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[57] 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 *ECCV*. Springer, 2014, pp. 740–755.

[58] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *IJCV*, 2020.

[59] R. Benenson, S. Popov, and V. Ferrari, "Large-scale interactive object segmentation with human annotators," in *CVPR*, 2019.

[60] S. Shao, Z. Li, T. Zhang, C. Peng, G. Yu, X. Zhang, J. Li, and J. Sun, "Objects365: A large-scale, high-quality dataset for object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8430–8439.

[61] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*. Ieee, 2009, pp. 248–255.

[62] I. Krasin and T. e. a. Duerig, "Openimages: A public dataset for large-scale multi-label and multi-class image classification." *Dataset available from https://storage.googleapis.com/openimages/web/index.html*, 2017.

[63] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *CVPR*. IEEE, 2009, pp. 304–311.

[64] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[65] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[66] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Advances in neural information processing systems*, 2013, pp. 2553–2561.

[67] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *ECCV*. Springer, 2016, pp. 354–370.

[68] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6154–6162.

[69] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, "Reppoints: Point set representation for object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9657–9666.

[70] T. Malisiewicz, *Exemplar-based representations for object detection, association and beyond*. Carnegie Mellon University, 2011.

[71] J. Hosang, R. Benenson, P. Dollár, and B. Schiele,

"What makes for effective detection proposals?" *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 4, pp. 814–830, 2016.

[72] J. Hosang, R. Benenson, and B. Schiele, "How good are detection proposals, really?" *arXiv preprint arXiv:1406.6962*, 2014.

[73] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?" in *CVPR*. IEEE, 2010, pp. 73–80.

[74] ——, "Measuring the objectness of image windows," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2189–2202, 2012.

[75] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "Bing: Binarized normed gradients for objectness estimation at 300fps," in *CVPR*, 2014, pp. 3286–3293.

[76] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *CVPR*, 2014, pp. 2147–2154.

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

[78] A. Torralba and P. Sinha, "Detecting faces in impoverished images," MASSACHUSETTS INST OF TECH CAMBRIDGE ARTIFICIAL INTELLIGENCE LAB, Tech. Rep., 2001.

[79] S. Zagoruyko, A. Lerer, T.-Y. Lin, P. O. Pinheiro, S. Gross, S. Chintala, and P. Dollár, "A multipath network for object detection," *arXiv preprint arXiv:1604.02135*, 2016.

[80] X. Zeng, W. Ouyang, B. Yang, J. Yan, and X. Wang, "Gated bi-directional cnn for object detection," in *ECCV*. Springer, 2016, pp. 354–369.

[81] X. Zeng, W. Ouyang, J. Yan, H. Li, T. Xiao, K. Wang, Y. Liu, Y. Zhou, B. Yang, Z. Wang *et al.*, "Crafting gbd-net for object detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 9, pp. 2109–2123, 2018.

[82] W. Ouyang, K. Wang, X. Zhu, and X. Wang, "Learning chained deep features and classifiers for cascade in object detection," *arXiv preprint arXiv:1702.07054*, 2017.

[83] S. Gidaris and N. Komodakis, "Object detection via a multi-region and semantic segmentation-aware cnn model," in *ICCV*, 2015, pp. 1134–1142.

[84] Y. Zhu, C. Zhao, J. Wang, X. Zhao, Y. Wu, H. Lu *et al.*, "Couplenet: Coupling global structure with local parts for object detection," in *ICCV*, vol. 2, 2017.

[85] C. Desai, D. Ramanan, and C. C. Fowlkes, "Discriminative models for multi-class object layout," *International journal of computer vision*, vol. 95, no. 1, pp. 1–12, 2011.

[86] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *CVPR*, 2016, pp. 2874–2883.

[87] Z. Li, Y. Chen, G. Yu, and Y. Deng, "R-fcn++: Towards accurate region-based fully convolutional networks for object detection." in *AAAI*, 2018.

[88] S. Liu, D. Huang *et al.*, "Receptive field block net for accurate and fast object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 385–400.

[89] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[90] Q. Chen, Z. Song, J. Dong, Z. Huang, Y. Hua, and S. Yan, "Contextualizing object detection and classification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 1, pp. 13–27, 2015.

[91] S. Gupta, B. Hariharan, and J. Malik, "Exploring person context and local scene context for object detection," *arXiv preprint arXiv:1511.08177*, 2015.

[92] X. Chen and A. Gupta, "Spatial memory for context reasoning in object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4086–4096.

[93] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3588–3597.

[94] Y. Liu, R. Wang, S. Shan, and X. Chen, "Structure inference net: Object detection using scene-level context and instance-level relationships," in *CVPR*, 2018, pp. 6985–6994.

[95] L. V. Pato, R. Negrinho, and P. M. Q. Aguiar, "Seeing without looking: Contextual rescoring of object detections for ap maximization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[96] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert, "An empirical study of context in object detection," in *CVPR*. IEEE, 2009, pp. 1271–1278.

[97] C. Chen, M.-Y. Liu, O. Tuzel, and J. Xiao, "R-cnn for small object detection," in *Asian conference on computer vision*. Springer, 2016, pp. 214–230.

[98] J. Li, Y. Wei, X. Liang, J. Dong, T. Xu, J. Feng, and S. Yan, "Attentive contexts for object detection," *IEEE Transactions on Multimedia*, vol. 19, no. 5, pp. 944–954, 2017.

[99] H. A. Rowley, S. Baluja, and T. Kanade, "Human face detection in visual scenes," in *Advances in Neural Information Processing Systems*, 1996, pp. 875–881.

[100] C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *ICCV*. IEEE, 1998, pp. 555–562.

[101] L. Zhang, L. Lin, X. Liang, and K. He, "Is faster r-cnn doing well for pedestrian detection?" in *ECCV*. Springer, 2016, pp. 443–457.

[102] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *CVPR*, 2016, pp. 761–769.

[103] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[104] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?" *Advances in neural information processing systems*, vol. 32, 2019.

[105] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "Unitbox: An advanced object detection network," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 516–520.

[106] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 658–666.

[107] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-iou loss: Faster and better learning for bounding box regression," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12 993–13 000.

[108] R. Vaillant, C. Monrocq, and Y. Le Cun, "Original approach for the localisation of objects in images," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 141, no. 4, pp. 245–250, 1994.

[109] P. Henderson and V. Ferrari, "End-to-end training of object class detectors for mean average precision," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 198–213.

[110] J. H. Hosang, R. Benenson, and B. Schiele, "Learning non-maximum suppression." in *CVPR*, 2017, pp. 6469–6477.

[111] Z. Tan, X. Nie, Q. Qian, N. Li, and H. Li, "Learning to rank proposals for object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8273–8281.

[112] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-nms—improving object detection with one line of code," in *ICCV*. IEEE, 2017, pp. 5562–5570.

[113] L. Tychsen-Smith and L. Petersson, "Improving object localization with fitness nms and bounded iou loss," *arXiv preprint arXiv:1711.00164*, 2017.

[114] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang, "Bounding box regression with uncertainty for accurate object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2888–2897.

[115] S. Liu, D. Huang, and Y. Wang, "Adaptive nms: Refining pedestrian detection in a crowd," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6459–6468.

[116] R. Rothe, M. Guillaumin, and L. Van Gool, "Non-maximum suppression for object detection by passing messages between windows," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 290–306.

[117] D. Mrowca, M. Rohrbach, J. Hoffman, R. Hu, K. Saenko, and T. Darrell, "Spatial semantic regularisation for large scale object detection," in *ICCV*, 2015, pp. 2003–2011.

[118] R. Solovyev, W. Wang, and T. Gabruseva, "Weighted boxes fusion: Ensembling boxes from different object detection models," *Image and Vision Computing*, vol. 107, p. 104117, 2021.

[119] Z. Zheng, P. Wang, D. Ren, W. Liu, R. Ye, Q. Hu, and W. Zuo, "Enhancing geometric factors in model learning and inference for object detection and instance segmentation," *IEEE Transactions on Cybernetics*, 2021.

[120] J. Wang, L. Song, Z. Li, H. Sun, J. Sun, and N. Zheng, "End-to-end object detection with fully convolutional network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 849–15 858.

[121] C. Papageorgiou and T. Poggio, "A trainable system for object detection," *International journal of computer vision*, vol. 38, no. 1, pp. 15–33, 2000.

[122] L. Wan, D. Eigen, and R. Fergus, "End-to-end integration of a convolution network, deformable parts model and non-maximum suppression," in *CVPR*, 2015, pp. 851–859.

[123] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *arXiv preprint arXiv:1905.05055*, 2019.

[124] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan, "Fast human detection using a cascade of histograms of oriented gradients," in *CVPR*, vol. 2. IEEE, 2006, pp. 1491–1498.

[125] F. Fleuret and D. Geman, "Coarse-to-fine face detection," *International Journal of computer vision*, vol. 41, no. 1-2, pp. 85–107, 2001.

[126] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *CVPR*, 2015, pp. 5325–5334.

[127] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[128] Z. Cai, M. Saberian, and N. Vasconcelos, "Learning complexity-aware cascades for deep pedestrian detection," in *ICCV*, 2015, pp. 3361–3369.

[129] Y. LeCun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in neural information processing systems*, 1990, pp. 598–605.

[130] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[131] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *CVPR*, 2015, pp. 5353–5360.

[132] Z. Qin, Z. Li, Z. Zhang, Y. Bao, G. Yu, Y. Peng, and J. Sun, "Thundernet: Towards real-time generic object detection on mobile devices," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6718–6727.

[133] R. J. Wang, X. Li, and C. X. Ling, "Pelee: A real-time object detection system on mobile devices," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 1967–1976.

[134] R. Huang, J. Pedoeem, and C. Chen, "Yolo-lite: a real-

time object detection algorithm optimized for non-gpu computers," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 2503–2510.

[135] H. Law, Y. Teng, O. Russakovsky, and J. Deng, "Cornernet-lite: Efficient keypoint based object detection," *arXiv preprint arXiv:1904.08900*, 2019.

[136] G. Yu, Q. Chang, W. Lv, C. Xu, C. Cui, W. Ji, Q. Dang, K. Deng, G. Wang, Y. Du *et al.*, "Pp-picodet: A better real-time object detector on mobile devices," *arXiv preprint arXiv:2111.00902*, 2021.

[137] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *CVPR*, 2016, pp. 2818–2826.

[138] X. Zhang, J. Zou, X. Ming, K. He, and J. Sun, "Efficient and accurate approximations of nonlinear convolutional networks," in *CVPR*, 2015, pp. 1984–1992.

[139] X. Zhang, J. Zou, K. He, and J. Sun, "Accelerating very deep convolutional networks for classification and detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 1943–1955, 2016.

[140] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," 2017.

[141] G. Huang, S. Liu, L. van der Maaten, and K. Q. Weinberger, "Condensenet: An efficient densenet using learned group convolutions," *group*, vol. 3, no. 12, p. 11, 2017.

[142] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *arXiv preprint*, pp. 1610–02 357, 2017.

[143] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[144] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *CVPR*. IEEE, 2018, pp. 4510–4520.

[145] Y. Li, J. Li, W. Lin, and J. Li, "Tiny-dsod: Lightweight object detection for resource-restricted usages," *arXiv preprint arXiv:1807.11013*, 2018.

[146] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[147] B. Wu, F. N. Iandola, P. H. Jin, and K. Keutzer, "Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving." in *CVPR Workshops*, 2017, pp. 446–454.

[148] T. Kong, A. Yao, Y. Chen, and F. Sun, "Hypernet: Towards accurate region proposal generation and joint object detection," in *CVPR*, 2016, pp. 845–853.

[149] Y. Chen, T. Yang, X. Zhang, G. Meng, C. Pan, and J. Sun, "Detnas: Neural architecture search on object detection," *arXiv preprint arXiv:1903.10979*, 2019.

[150] H. Xu, L. Yao, W. Zhang, X. Liang, and Z. Li, "Auto-fpn: Automatic network architecture adaptation for object detection beyond classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6649–6658.

[151] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "Nas-fpn: Learning scalable feature pyramid architecture for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7036–7045.

[152] J. Guo, K. Han, Y. Wang, C. Zhang, Z. Yang, H. Wu, X. Chen, and C. Xu, "Hit-detector: Hierarchical trinity architecture search for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 405–11 414.

[153] N. Wang, Y. Gao, H. Chen, P. Wang, Z. Tian, C. Shen, and Y. Zhang, "Nas-fcos: Fast neural architecture search for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 943–11 951.

[154] L. Yao, H. Xu, W. Zhang, X. Liang, and Z. Li, "Sm-nas: structural-to-modular neural architecture search for object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12 661–12 668.

[155] C. Jiang, H. Xu, W. Zhang, X. Liang, and Z. Li, "Sp-nas: Serial-to-parallel backbone search for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 863–11 872.

[156] P. Simard, L. Bottou, P. Haffner, and Y. LeCun, "Boxlets: a fast convolution algorithm for signal processing and neural networks," in *Advances in Neural Information Processing Systems*, 1999, pp. 571–577.

[157] X. Wang, T. X. Han, and S. Yan, "An hog-lbp human detector with partial occlusion handling," in *ICCV*. IEEE, 2009, pp. 32–39.

[158] F. Porikli, "Integral histogram: A fast way to extract histograms in cartesian spaces," in *CVPR*, vol. 1. IEEE, 2005, pp. 829–836.

[159] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," 2009.

[160] M. Mathieu, M. Henaff, and Y. LeCun, "Fast training of convolutional networks through ffts," *arXiv preprint arXiv:1312.5851*, 2013.

[161] H. Pratt, B. Williams, F. Coenen, and Y. Zheng, "Fcnn: Fourier convolutional neural networks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2017, pp. 786–798.

[162] N. Vasilache, J. Johnson, M. Mathieu, S. Chintala, S. Piantino, and Y. LeCun, "Fast convolutional nets with fbfft: A gpu performance evaluation," *arXiv preprint arXiv:1412.7580*, 2014.

[163] O. Rippel, J. Snoek, and R. P. Adams, "Spectral representations for convolutional neural networks," in *Advances in neural information processing systems*, 2015, pp. 2449–2457.

[164] M. A. Sadeghi and D. Forsyth, "Fast template evaluation with vector quantization," in *Advances in neural*

information processing systems, 2013, pp. 2949–2957.

[165] I. Kokkinos, "Bounding part scores for rapid detection with deformable part models," in *ECCV*. Springer, 2012, pp. 41–50.

[166] H. Zhu, X. Chen, W. Dai, K. Fu, Q. Ye, and J. Jiao, "Orientation robust object detection in aerial images using deep convolutional neural network," in *ICIP*. IEEE, 2015, pp. 3735–3739.

[167] B. Cai, Z. Jiang, H. Zhang, Y. Yao, and S. Nie, "Online exemplar-based fully convolutional network for aircraft detection in remote sensing images," *IEEE Geoscience and Remote Sensing Letters*, no. 99, pp. 1–5, 2018.

[168] G. Cheng, J. Han, P. Zhou, and L. Guo, "Multi-class geospatial object detection and geographic image classification based on collection of part detectors," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 98, pp. 119–132, 2014.

[169] G. Cheng, P. Zhou, and J. Han, "Rifd-cnn: Rotation-invariant and fisher discriminative convolutional neural networks for object detection," in *CVPR*, 2016, pp. 2884–2893.

[170] ——, "Learning rotation-invariant convolutional neural networks for object detection in vhr optical remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 12, pp. 7405–7415, 2016.

[171] G. Cheng, J. Han, P. Zhou, and D. Xu, "Learning rotation-invariant and fisher discriminative convolutional neural networks for object detection," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 265–278, 2018.

[172] X. Shi, S. Shan, M. Kan, S. Wu, and X. Chen, "Real-time rotation-invariant face detection with progressive calibration networks," in *CVPR*, 2018, pp. 2295–2303.

[173] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *Advances in neural information processing systems*, 2015, pp. 2017–2025.

[174] D. Chen, G. Hua, F. Wen, and J. Sun, "Supervised transformer network for efficient face detection," in *ECCV*. Springer, 2016, pp. 122–138.

[175] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu, "Learning roi transformer for oriented object detection in aerial images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2849–2858.

[176] B. Singh and L. S. Davis, "An analysis of scale invariance in object detection–snip," in *CVPR*, 2018, pp. 3578–3587.

[177] B. Singh, M. Najibi, and L. S. Davis, "Sniper: Efficient multi-scale training," *arXiv preprint arXiv:1805.09300*, 2018.

[178] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[179] M. Gao, R. Yu, A. Li, V. I. Morariu, and L. S. Davis, "Dynamic zoom-in network for fast object detection in large images," in *CVPR*, 2018.

[180] Y. Lu, T. Javidi, and S. Lazebnik, "Adaptive object detection using adjacency and zoom prediction," in *CVPR*, 2016, pp. 2351–2359.

[181] S. Qiao, W. Shen, W. Qiu, C. Liu, and A. L. Yuille, "Scalenet: Guiding object proposal generation in supermarkets and beyond." in *ICCV*, 2017, pp. 1809–1818.

[182] Z. Hao, Y. Liu, H. Qin, J. Yan, X. Li, and X. Hu, "Scale-aware face detection," in *CVPR*, vol. 3, 2017.

[183] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "Cspnet: A new backbone that can enhance learning capability of cnn," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 390–391.

[184] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *European conference on computer vision*. Springer, 2016, pp. 483–499.

[185] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, L. Wang, G. Wang *et al.*, "Recent advances in convolutional neural networks," *arXiv preprint arXiv:1512.07108*, 2015.

[186] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *CVPR*, vol. 4, 2017.

[187] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *CVPR*, vol. 1, no. 2, 2018, p. 10.

[188] R. N. Rajaram, E. Ohn-Bar, and M. M. Trivedi, "Refinenet: Iterative refinement for accurate object localization," in *ITSC*. IEEE, 2016, pp. 1528–1533.

[189] M.-C. Roh and J.-y. Lee, "Refining faster-rcnn for accurate object detection," in *Machine Vision Applications (MVA), 2017 Fifteenth IAPR International Conference on*. IEEE, 2017, pp. 514–517.

[190] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Proceedings of the ECCV, Munich, Germany*, 2018, pp. 8–14.

[191] S. Gidaris and N. Komodakis, "Locnet: Improving localization accuracy for object detection," in *CVPR*, 2016, pp. 789–798.

[192] S. Brahmbhatt, H. I. Christensen, and J. Hays, "Stuffnet: Using 'stuff'to improve object detection," in *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*. IEEE, 2017, pp. 934–943.

[193] A. Shrivastava and A. Gupta, "Contextual priming and feedback for faster r-cnn," in *ECCV*. Springer, 2016, pp. 330–348.

[194] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[195] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[196] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *arXiv preprint*, 2017.

[197] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network." in *CVPR*, vol. 2, no. 3, 2017, p. 4.

[198] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual generative adversarial networks for small object detection," in *CVPR*, 2017.

[199] Y. Bai, Y. Zhang, M. Ding, and B. Ghanem, "Sodmtgan: Small object detection via multi-task generative adversarial network," *Computer Vision-ECCV*, pp. 8–14, 2018.

[200] X. Wang, A. Shrivastava, and A. Gupta, "A-fast-rcnn: Hard positive generation via adversary for object detection," in *CVPR*, 2017.

[201] D. Zhang, J. Han, G. Cheng, and M.-H. Yang, "Weakly supervised object localization and detection: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 9, pp. 5866–5885, 2021.

[202] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial intelligence*, vol. 89, no. 1-2, pp. 31–71, 1997.

[203] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Advances in neural information processing systems*, 2003, pp. 577–584.

[204] R. G. Cinbis, J. Verbeek, and C. Schmid, "Weakly supervised object localization with multi-fold multiple instance learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 1, pp. 189–203, 2017.

[205] D. P. Papadopoulos, J. R. Uijlings, F. Keller, and V. Ferrari, "We don't need no bounding-boxes: Training object class detectors using only human verification," in *CVPR*, 2016, pp. 854–863.

[206] D. Zhang, W. Zeng, J. Yao, and J. Han, "Weakly supervised object detection using proposal-and semantic-level relationships," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[207] P. Tang, X. Wang, S. Bai, W. Shen, X. Bai, W. Liu, and A. Yuille, "Pcl: Proposal cluster learning for weakly supervised object detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 1, pp. 176–191, 2018.

[208] E. Sangineto, M. Nabi, D. Culibrk, and N. Sebe, "Self paced deep learning for weakly supervised object detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 3, pp. 712–725, 2018.

[209] D. Zhang, J. Han, L. Zhao, and D. Meng, "Leveraging prior-knowledge for weakly supervised object detection under a collaborative self-paced curriculum learning framework," *International Journal of Computer Vision*, vol. 127, no. 4, pp. 363–380, 2019.

[210] Y. Zhu, Y. Zhou, Q. Ye, Q. Qiu, and J. Jiao, "Soft proposal networks for weakly supervised object localization," in *ICCV*, 2017, pp. 1841–1850.

[211] A. Diba, V. Sharma, A. M. Pazandeh, H. Pirsiavash, and L. Van Gool, "Weakly supervised cascaded convolutional networks." in *CVPR*, vol. 1, no. 2, 2017, p. 8.

[212] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *CVPR*, 2016, pp. 2921–2929.

[213] H. Bilen and A. Vedaldi, "Weakly supervised deep detection networks," in *CVPR*, 2016, pp. 2846–2854.

[214] L. Bazzani, A. Bergamo, D. Anguelov, and L. Torresani, "Self-taught object localization with deep networks," in *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*. IEEE, 2016, pp. 1–9.

[215] Y. Shen, R. Ji, S. Zhang, W. Zuo, and Y. Wang, "Generative adversarial learning towards fast weakly supervised detection," in *CVPR*, 2018, pp. 5764–5773.

[216] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain adaptive faster r-cnn for object detection in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3339–3348.

[217] Y. Wang, R. Zhang, S. Zhang, M. Li, Y. Xia, X. Zhang, and S. Liu, "Domain-specific suppression for adaptive object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9603–9612.

[218] L. Hou, Y. Zhang, K. Fu, and J. Li, "Informative and consistent correspondence mining for cross-domain weakly supervised object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9929–9938.

[219] X. Zhu, J. Pang, C. Yang, J. Shi, and D. Lin, "Adapting object detectors via selective cross-domain alignment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 687–696.

[220] K. Saito, Y. Ushiku, T. Harada, and K. Saenko, "Strong-weak distribution alignment for adaptive object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6956–6965.

[221] C.-D. Xu, X.-R. Zhao, X. Jin, and X.-S. Wei, "Exploring categorical regularization for domain adaptive object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 724–11 733.

[222] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[223] T. Kim, M. Jeong, S. Kim, S. Choi, and C. Kim, "Diversify and match: A domain adaptive representation learning paradigm for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 456–12 465.

[224] N. Inoue, R. Furuta, T. Yamasaki, and K. Aizawa, "Cross-domain weakly-supervised object detection through progressive domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern*

recognition, 2018, pp. 5001–5009.

[225] H.-K. Hsu, C.-H. Yao, Y.-H. Tsai, W.-C. Hung, H.-Y. Tseng, M. Singh, and M.-H. Yang, "Progressive domain adaptation for object detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 749–757.

[226] B. Bosquet, M. Mucientes, and V. M. Brea, "Stdnet-st: Spatio-temporal convnet for small object detection," *Pattern Recognition*, vol. 116, p. 107929, 2021.

[227] C. Yang, Z. Huang, and N. Wang, "Querydet: Cascaded sparse query for accelerating high-resolution small object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 668–13 677.

[228] P. Sun, Y. Jiang, E. Xie, W. Shao, Z. Yuan, C. Wang, and P. Luo, "What makes for end-to-end object detection?" in *International Conference on Machine Learning*. PMLR, 2021, pp. 9934–9944.

[229] X. Zhou, X. Xu, W. Liang, Z. Zeng, S. Shimizu, L. T. Yang, and Q. Jin, "Intelligent small object detection for digital twin in smart manufacturing with industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1377–1386, 2021.

[230] G. Cheng, X. Yuan, X. Yao, K. Yan, Q. Zeng, and J. Han, "Towards large-scale small object detection: Survey and benchmarks," *arXiv preprint arXiv:2207.14096*, 2022.

[231] Y. Wang, V. C. Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. Solomon, "Detr3d: 3d object detection from multi-view images via 3d-to-2d queries," in *Conference on Robot Learning*. PMLR, 2022, pp. 180–191.

[232] Y. Wang, T. Ye, L. Cao, W. Huang, F. Sun, F. He, and D. Tao, "Bridged transformer for vision and point cloud 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 114–12 123.

[233] X. Cheng, H. Xiong, D.-P. Fan, Y. Zhong, M. Harandi, T. Drummond, and Z. Ge, "Implicit motion handling for video camouflaged object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 864–13 873.

[234] Q. Zhou, X. Li, L. He, Y. Yang, G. Cheng, Y. Tong, L. Ma, and D. Tao, "Transvod: End-to-end video object detection with spatial-temporal transformers," *arXiv preprint arXiv:2201.05047*, 2022.

[235] R. Cong, Q. Lin, C. Zhang, C. Li, X. Cao, Q. Huang, and Y. Zhao, "Cir-net: Cross-modality interaction and refinement for rgb-d salient object detection," *IEEE Transactions on Image Processing*, 2022.

[236] Y. Wang, L. Zhu, S. Huang, T. Hui, X. Li, F. Wang, and S. Liu, "Cross-modality domain adaptation for freespace detection: A simple yet effective baseline," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 4031–4042.

[237] C. Feng, Y. Zhong, Z. Jie, X. Chu, H. Ren, X. Wei, W. Xie, and L. Ma, "Promptdet: Expand your detector vocabulary with uncurated images," *arXiv preprint arXiv:2203.16513*, 2022.

[238] Y. Zhong, J. Yang, P. Zhang, C. Li, N. Codella, L. H. Li, L. Zhou, X. Dai, L. Yuan, Y. Li *et al.*, "Regionclip: Region-based language-image pretraining," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 793–16 803.