Dear Author

Please use this PDF proof to check the layout of your article. If you would like any changes to be made to the layout, you can leave instructions in the online proofing interface. Making your changes directly in the online proofing interface is the quickest, easiest way to correct and submit your proof. Please note that changes made to the article in the online proofing interface will be added to the article before publication, but are not reflected in this PDF proof.

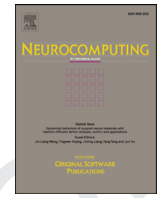
If you would prefer to submit your corrections by annotating the PDF proof, please download and submit an annotatable PDF proof by clicking here and you'll be redirected to our PDF Proofing system.

Neurocomputing xxx (xxxx) xxx

Contents lists available at ScienceDirect

Neurocomputing

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



Recent advances in deep learning for object detection

Xiongwei Wu[a,*], Doyen Sahoo[a,1], Steven C.H. Hoi[a,b]

[a] School of Information System, Singapore Management University, Singapore
[b] Salesforce Research Asia, USA

A R T I C L E I N F O

Article history:
Received 11 August 2019
Revised 9 January 2020
Accepted 21 January 2020
Available online xxx

Communicated by Dr Zenglin Xu

Keywords:
Object detection
Deep learning
Deep convolutional neural networks

A B S T R A C T

Object detection is a fundamental visual recognition problem in computer vision and has been widely studied in the past decades. Visual object detection aims to find objects of certain target classes with precise localization in a given image and assign each object instance a corresponding class label. Due to the tremendous successes of deep learning based image classification, object detection techniques using deep learning have been actively studied in recent years. In this paper, we give a comprehensive survey of recent advances in visual object detection with deep learning. By reviewing a large body of recent related work in literature, we systematically analyze the existing object detection frameworks and organize the survey into three major parts: (i) detection components, (ii) learning strategies, and (iii) applications & benchmarks. In the survey, we cover a variety of factors affecting the detection performance in detail, such as detector architectures, feature learning, proposal generation, sampling strategies, etc. Finally, we discuss several future directions to facilitate and spur future research for visual object detection with deep learning.

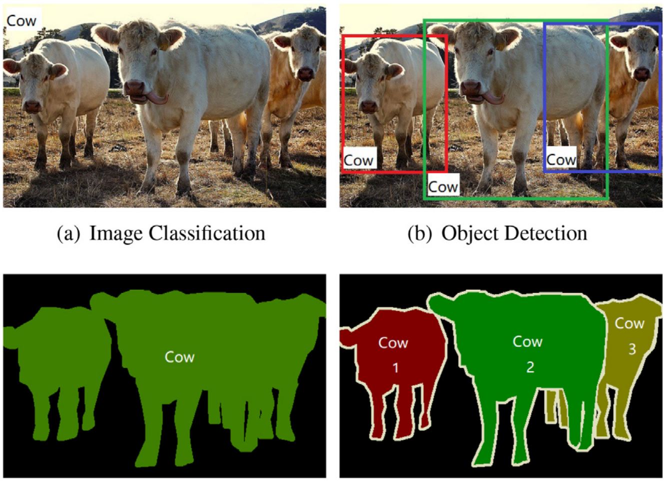
1. Introduction

In the field of computer vision, there are several fundamental visual recognition problems: image classification [1], object detection and instance segmentation [2,3], and semantic segmentation [4] (see Fig. 1). In particular, image classification (Fig 1.1(a)), aims to recognize semantic categories of objects in a given image. Object detection not only recognizes object categories, but also predicts the location of each object by a bounding box (Fig. 1(b)). Semantic segmentation (Fig. 1(c)) aims to predict pixel-wise classifiers to assign a specific category label to each pixel, thus providing an even richer understanding of an image. However, in contrast to object detection, semantic segmentation does not distinguish between multiple objects of the same category. A relatively new setting at the intersection of object detection and semantic segmentation, named "instance segmentation" (Fig. 1(d)), is proposed to identify different objects and assign each of them a separate categorical pixel-level mask. In fact, instance segmentation can be viewed as a special setting of object detection, where instead of localizing an object by a bounding box, pixel-level localization is desired. In this survey, we direct our attention to review the major efforts in deep learning based object detection. A good detection algorithm should have a strong understanding of semantic cues as well as the spatial information about the image. In fact, object detection is the basic step towards many computer vision applications, such as face recognition [5–7], pedestrian detection [8–10], video analysis [11,12], and logo detection [13–15].

In the early stages, before the deep learning era, the pipeline of object detection was divided into three steps: (i) proposal generation; (ii) feature vector extraction; and (iii) region classification. During proposal generation, the objective was to search locations in the image which may contain objects. These locations are also called regions of interest (roi). An intuitive idea is to scan the whole image with sliding windows [16–20]. In order to capture information about multi-scale and different aspect ratios of objects, input images were resized into different scales and multi-scale windows were used to slide through these images. During the second step, on each location of the image, a fixed-length feature vector was obtained from the sliding window, to capture discriminative semantic information of the region covered. This feature vector was commonly encoded by low-level visual descriptors such as SIFT (Scale Invariant Feature Transform) [21], Haar [22], HOG (Histogram of Gradients) [19] or SURF (Speeded Up Robust Features) [23], which showed a certain robustness to scale, illumination and rotation variance. Finally, in the third step, the region classifiers were learned to assign categorical labels to the covered regions. Commonly, support vector machines (SVM) [24] were used here due to their good performance on small scale training data. In addition, some classification techniques such as bagging [25],

* Corresponding author.
E-mail addresses: xwwu.2015@phdis.smu.edu.sg (X. Wu), chhoi@smu.edu.sg (S.C.H. Hoi).
[1] doyens.smu.edu.sg

https://doi.org/10.1016/j.neucom.2020.01.085
0925-2312/© 2020 Published by Elsevier B.V.



(a) Image Classification (b) Object Detection

(c) Semantic Segmentation (d) Instance Segmentation

Fig. 1. Comparison of different visual recognition tasks in computer vision. (a) "Image Classification" only needs to assign categorical class labels to the image; (b) "Object detection" not only predict categorical labels but also localize each object instance via bounding boxes; (c) "Semantic segmentation" aims to predict categorical labels for each pixel, without differentiating object instances; (d) "Instance segmentation", a special setting of object detection, differentiates different object instances by pixel-level segmentation masks.

cascade learning [20] and adaboost [26] were used in region classification step, leading to further improvements in detection accuracy.

Most of the successful traditional methods for object detection focused on carefully designing feature descriptors to obtain embedding for a region of interest. With the help of good feature representations as well as robust region classifiers, impressive results [27,28] were achieved on Pascal VOC dataset [29] (a publicly available dataset used for benchmarking object detection). Notably, deformable part based machines (DPMs) [30], a breakthrough detection algorithm, were 3-time winners on VOC challenges in 2007, 2008 and 2009. DPMs learn and integrate multiple part models with a deformable loss and mine hard negative examples with a latent SVM for discriminative training. However, during 2008 to 2012, the progress on Pascal VOC based on these traditional methods had become incremental, with minor gains from building complicated ensemble systems. This showed the limitations of these traditional detectors. Most prominently, these limitations included: (i) during proposal generation, a huge number of proposals were generated, and many of them were redundant; this resulted in a large number of false positives during classification. Moreover, window scales were designed manually and heuristically, and could not match the objects well; (ii) feature descriptors were hand-crafted based on low level visual cues [23,31,32], which made it difficult to capture representative semantic information in complex contexts. (iii) each step of the detection pipeline was designed and optimized separately, and thus could not obtain a global optimal solution for the whole system.

After the success of applying deep convolutional neural networks (DCNN) for image classification [1,33], object detection also achieved remarkable progress based on deep learning techniques [2,34]. The new deep learning based algorithms outperformed the traditional detection algorithms by huge margins. Deep convolutional neural network is a biologically-inspired structure for computing hierarchical features. An early attempt to build such a hierarchical and spatial-invariant model for image classification was "neocognitron" [35] proposed by Fukushima. However, this early attempt lacked effective optimization techniques for supervised learning. Based on this model, Lecun et al. [36] optimized a convolutional neural network by stochastic gradient descent (SGD) via back-propagation and showed competitive performance on digit recognition. After that, however, deep convolutional neural networks were not heavily explored, with support vector machines becoming more prominent. This was because deep learning had some limitations: (i) lack of large scale annotated training data, which caused overfitting; (ii) limited computation resources; and (iii) weak theoretical support compared to SVMs. In 2009, Jia et al. [37] collected a large scale annotated image dataset ImageNet which contained 1.2M high resolution images, making it possible to train deep models with large scale training data. With the development of computing resources on parallel computing systems (such as GPU clusters), in 2012 Krizhevsky et al. [33] trained a large deep convolutional model with ImageNet dataset and showed significant improvement on Large Scale Visual Recognition Challenge (ILSVRC) compared to all other approaches. After the success of applying DCNN for classification, deep learning techniques were quickly adapted to other vision tasks and showed promising results compared to the traditional methods.

In contrast to hand-crafted descriptors used in traditional detectors, deep convolutional neural networks generate hierarchical feature representations from raw pixels to high level semantic information, which is learned automatically from the training data and shows more discriminative expression capability in complex contexts. Furthermore, benefiting from the powerful learning capacity, a deep convolutional neural network can obtain a better feature representation with a larger dataset, while the learning capacity of traditional visual descriptors are fixed, and can not improve when more data becomes available. These properties made it possible to design object detection algorithms based on deep convolutional neural networks which could be optimized in an end-to-end manner, with more powerful feature representation capability.

Currently, deep learning based object detection frameworks can be primarily divided into two families: (i) two-stage detectors, such as Region-based CNN (R-CNN) [2] and its variants [34,38,39] and (ii) one-stage detectors, such as YOLO [40] and its variants [41,42]. Two-stage detectors first use a proposal generator to generate a sparse set of proposals and extract features from each proposal, followed by region classifiers which predict the category of the proposed region. One-stage detectors directly make categorical prediction of objects on each location of the feature maps without the cascaded region classification step. Two-stage detectors commonly achieve better detection performance and report state-of-the-art results on public benchmarks, while one-stage detectors are significantly more time-efficient and have greater applicability to real-time object detection. Fig. 2 also illustrates the major developments and milestones of deep learning based object detection techniques after 2012. We will cover basic ideas of these key techniques and analyze them in a systematic manner in the survey.

The goal of this survey is to present a comprehensive understanding of deep learning based object detection algorithms. Fig. 3 shows a taxonomy of key methodologies to be covered in this survey. We review various contributions in deep learning based object detection and categorize them into three groups: detection components, learning strategies, and applications & benchmarks. For detection components, we first introduce two detection settings: bounding box level (bbox-level) and pixel mask level (mask-level) localization. Bbox-level algorithms require to localize objects by rectangle bounding boxes, while more precise pixel-wise masks are required to segment objects in mask-level algorithms. Next, we summarize the representative frameworks of two detection families: two-stage detection and one-stage detection. Then we give a detailed survey of each detection component, including backbone architecture, proposal generation and feature learning. For learning strategies, we first highlight the importance of learning strategy of

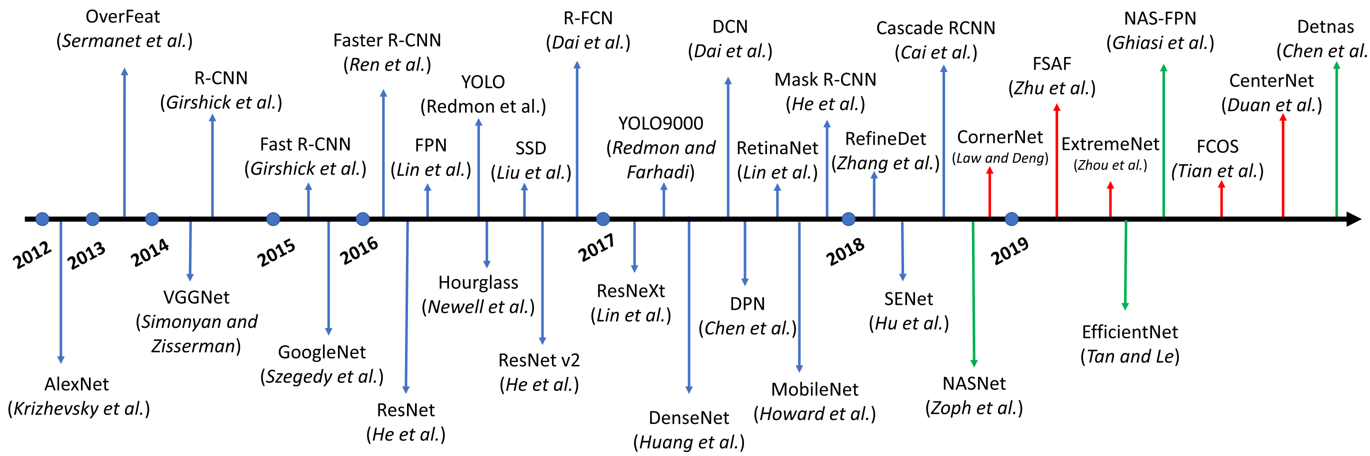


Fig. 2. Major milestone in object detection research based on deep convolution neural networks since 2012. The trend in the last year has been designing object detectors based on anchor-free (in red) and AutoML (in green) techniques, which are potentially two important research directions in the future. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.) Q3

Object Detection				
Detection Components			**Learning Strategy**	**Applications & Benchmarks**
Detection Settings	**Detection Paradigms**	**Backbone Architecture**	**Training Stage**	**Applications**
Bounding Box	Two-Stage Detectors	VGG16,ResNet,DenseNet	Data Augmentation; Imbalance Sampling	Face Detection
Pixel Mask	One-Stage Detectors	MobileNet, ResNeXt	Localization Refinement	Pedestrian Detection
		DetNet, Hourglass Net	Cascade Learning; Others	Others
Proposal Generation		**Feature Representation**	**Testing Stage**	**Public Benchmarks**
Traditional Computer Vision Methods		Multi-scale Feature Learning	Duplicate Removal	MSCOCO, Pascal VOC, Open Images
Anchor-based Methods		Region Feature Encoding		
Keypoint-based Methods		Contextual Reasoning	Model Acceleration	FDDB, WIDER FACE
Other Methods		Deformable Feature Learning	Others	KITTI, ETH, CityPersons

Fig. 3. Taxonomy of key methodologies in this survey. We categorize various contributions for deep learning based object detection into three major categories: Detection Components, Learning Strategies, Applications and Benchmarks. We review each of these categories in detail.

detection due to the difficulty of training detectors, and then introduce the optimization techniques for both training and testing stages in detail. Finally, we review some real-world object detection based applications including face detection, pedestrian detection, logo detection and video analysis. We also discuss publicly available and commonly used benchmarks and evaluation metrics for these detection tasks. Finally we show the state-of-the-art results of generic detection on public benchmarks over the recent years.

We hope our survey can provide a timely review for researchers and practitioners to further catalyze research on detection systems. The rest of the paper are organized as follows: in Section 2, we give a standard problem setting of object detection. The details of detector components are listed in Section 3. Then the learning strategies are presented in Section 4. Detection algorithms for real-world applications and benchmarks are provided in Sections 5 and 6. State-of-the-art results of generic detection, face detection and pedestrian detection are listed in Section 7. Finally, we conclude and discuss future directions in Section 9. The code is available at https://github.com/XiongweiWu/Awesome-Object-Detection.

2. Problem settings

In this section, we present the formal problem setting for object detection based on deep learning. Object detection involves both recognition (e.g., "object classification") and localization (e.g., "location regression") tasks. An object detector needs to distinguish objects of certain target classes from backgrounds in the image with precise localization and correct categorical label prediction to each object instance. Bounding boxes or pixel masks are predicted to localize these target object instances.

More formally, assume we are given a collection of N annotated images $\{x_1, x_2, \ldots, x_N\}$, and for ith image x_i, there are M_i objects belonging to C categories with annotations:

$$y_i = \left\{(c_1^i, b_1^i), (c_2^i, b_2^i), \ldots, (c_{M_i}^i, b_{M_i}^i)\right\} \tag{1}$$

where $c^i_j (c^i_j \in C)$ and b^i_j (bounding box or pixel mask of the object) denote categorical and spatial labels of jth object in x_i respectively. The detector is f parameterized by θ. For x_i, the prediction y^i_{pred} shares the same format as y_i:

$$y^i_{\text{pred}} = \left\{ (c^i_{\text{pred}_1}, b^i_{\text{pred}_1}), (c^i_{\text{pred}_2}, b^i_{\text{pred}_2}), \ldots \right\} \tag{2}$$

Finally a loss function ℓ is set to optimize detector as:

$$\ell(x, \theta) = \frac{1}{N} \sum_{i=1}^{N} \ell(y^i_{\text{pred}}, x_i, y_i; \theta) + \frac{\lambda}{2} \|\theta\|_2^2 \tag{3}$$

where the second term is a regularizer, with trade-off parameter λ. Different loss functions such as softmax loss [38] and focal loss [43] impact the final detection performance, and we will discuss these functions in Section 4.

At the time of evaluation, a metric called intersection-over-union (IoU) between objects and predictions is used to evaluate the quality of localization (we omit index i here):

$$\text{IoU}(b_{\text{pred}}, b_{\text{gt}}) = \frac{\text{Area}(b_{\text{pred}} \bigcap b_{\text{gt}})}{\text{Area}(b_{\text{pred}} \bigcup b_{\text{gt}})} \tag{4}$$

Here, b_{gt} refers to the ground truth bbox or mask. An IoU threshold Ω is set to determine whether a prediction *tightly* covers the object or not (i.e. IoU $\geq \Omega$; commonly researchers set $\Omega = 0.5$). For object detection, a prediction with correct categorical label as well as successful localization prediction (meeting the IoU criteria) is considered as positive, otherwise it's a negative prediction:

$$\text{Prediction} = \begin{cases} \text{Positive} & c_{\text{pred}} = c_{\text{gt}} \text{ and } \text{IoU}(b_{\text{pred}}, b_{\text{gt}}) > \Omega \\ \text{Negative} & \text{otherwise} \end{cases} \tag{5}$$

For generic object detection problem evaluation, mean average precision (mAP) over C classes is used for evaluation, and in real world scenarios such as pedestrian detection, different evaluation metrics are used. The details of evaluation metric for different detection tasks will be discussed in Section 6. In addition to detection accuracy, inference speed is also an important metric to evaluate object detection algorithms. Specifically, if we wish to detect objects in a video stream (real-time detection), it is imperative to have a detector that can process this information quickly. Thus, the detector efficiency is also evaluated on Frame per second (FPS), i.e., how many images it can process per second. Commonly a detector that can achieve an inference speed of 20 FPS, is considered to be a real-time detector.

3. Detection components

In this section, we introduce different components of object detection. The first is about the choice of object detection paradigm. We first introduce the concepts of two detection settings: bbox-level and mask-level algorithms. Then, We introduce two major object detection paradigms: two-stage detectors and one-stage detectors. Under these paradigms, detectors can use a variety of deep learning backbone architectures, proposal generators, and feature representation modules.

3.1. Detection settings

There are two settings in object detection: (i) vanilla object detection (bbox-level localization) and (ii) instance segmentation (pixel-level or mask-level localization). Vanilla object detection has been more extensively studied and is considered as the traditional detection setting, where the goal is to localize objects by rectangle bounding boxes. In vanilla object detection algorithms, only bbox annotations are required, and in evaluation, the IoU between predicted bounding box with the ground truth is calculated to measure the performance. Instance segmentation is a relatively new setting and is based on traditional detection setting. Instance segmentation requires to segment each object by a pixel-wise mask instead of a rough rectangle bounding box. Due to more precise pixel-level prediction, instance segmentation is more sensitive to spatial misalignment, and thus has higher requirement to process the spatial information. The evaluation metric of instance segmentation is almost identical to the bbox-level detection, except that the IoU computation is performed on mask predictions. Though the two detection settings are slightly different, the main components introduced later can mostly be shared by the two settings.

3.2. Detection paradigms

Current state-of-the-art object detectors with deep learning can be mainly divided into two major categories: two-stage detectors and one-stage detectors. For a two-stage detector, in the first stage, a sparse set of proposals is generated; and in the second stage, the feature vectors of generated proposals are encoded by deep convolutional neural networks followed by making the object class predictions. An one-stage detector does not have a separate stage for proposal generation (or learning a proposal generation). They typically consider all positions on the image as potential objects, and try to classify each region of interest as either background or a target object. Two-stage detectors often reported state-of-the-art results on many public benchmark datasets. However, they generally fall short in terms of lower inference speeds. One-stage detectors are much faster and more desired for real-time object detection applications, but have a relatively poor performance compared to the two-stage detectors.

3.2.1. Two-stage detectors

Two-stage detectors split the detection task into two stages: (i) proposal generation; and (ii) making predictions for these proposals. During the proposal generation phase, the detector will try to identify regions in the image which may potentially be objects. The idea is to propose regions with a high recall, such that all objects in the image belong to at least one of these proposed region. In the second stage, a deep-learning based model is used to classify these proposals with the right categorical labels. The region may either be a background, or an object from one of the predefined class labels. Additionally, the model may refine the original localization suggested by the proposal generator. Next, we review some of the most influential efforts among two-stage detectors.

R-CNN [2] is a pioneering two-stage object detector proposed by Girshick et al. in 2014. Compared to the previous state-of-the-art methods based on a traditional detection framework SegDPM [44] with 40.4% mAP on Pascal VOC2010, R-CNN significantly improved the detection performance and obtained 53.7% mAP. The pipeline of R-CNN can be divided into three components: (i) proposal generation, (ii) feature extraction and (iii) region classification. For each image, R-CNN generates a sparse set of proposals (around 2000 proposals) via Selective Search [45], which is designed to reject regions that can easily be identified as background regions. Then, each proposal is cropped and resized into a fixed-size region and is encoded into a (e.g. 4096 dimensional) feature vector by a deep convolutional neural network, followed by a one-vs-all SVM classifier. Finally the bounding box regressors are learned using the extracted features as input in order to make the original proposals tightly bound the objects. Compared to traditional hand-crafted feature descriptors, deep neural networks generate hierarchical features and capture different scale information in different layers, and finally produce robust and discriminative features for classification. utilize the power of transfer learning, R-CNN adopts weights of convolutional networks pre-trained on ImageNet. The last fully connected layer (FC layer) is re-initialized for the detection task. The whole detector is then finetuned on the

Please cite this article as: X. Wu, D. Sahoo and S.C.H. Hoi, Recent advances in deep learning for object detection, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.01.085

pre-trained model. This transfer of knowledge from the Imagenet dataset offers significant performance gains. In addition, R-CNN rejects huge number of easy negatives before training, which helps improve learning speed and reduce false positives.

However, R-CNN faces some critical shortcomings: (i) the features of each proposal were extracted by deep convolutional networks *separately* (i.e., computation was not shared), which led to heavily duplicated computations. Thus, R-CNN was extremely time-consuming for training and testing; (ii) the three steps of R-CNN (proposal generation, feature extraction and region classification) were independent components and the whole detection framework could not be optimized in an end-to-end manner, making it difficult to obtain global optimal solution; and (iii) Selective Search relied on low-level visual cues and thus struggled to generate high quality proposals in complex contexts. Moreover, it is unable to enjoy the benefits of GPU acceleration.

Inspired by the idea of spatial pyramid matching (SPM) [46], He et al. proposed **SPP-net** [47] to accelerate R-CNN as well as learn more discriminative features. Instead of cropping proposal regions and feeding into CNN model separately, SPP-net computes the feature map from the whole image using a deep convolutional network and extracts fixed-length feature vectors on the feature map by a Spatial Pyramid Pooling (SPP) layer. SPP partitions the feature map into an $N \times N$ grid, for multiple values of N (thus allowing obtaining information at different scales), and performs pooling on each cell of the grid, to give a feature vector. The feature vectors obtained from each $N \times N$ grid are concatenated to give the representation for the region. The extracted features are fed into region SVM classifiers and bounding box regressors. In contrast to RCNN, SPP-layer can also work on images/regions at various scales and aspect ratios without resizing them. Thus, it does not suffer from information loss and unwanted geometric distortion.

SPP-net achieved better results and had a significantly faster inference speed compared to R-CNN. However, the training of SPP-net was still multi-stage and thus it could not be optimized end-to-end (and required extra cache memory to store extracted features). In addition, SPP layer did not back-propagate gradients to convolutional kernels and thus all the parameters before the SPP layer were frozen. This significantly limited the learning capability of deep backbone architectures. Girshick et al. proposed **Fast R-CNN** [38], a multi-task learning detector which addressed these two limitations of SPP-net. Fast R-CNN (like SPP-Net) also computed a feature map for the whole image and extracted fixed-length region features on the feature map. Different from SPP-net, Fast R-CNN used ROI Pooling layer to extract region features. *ROI pooling* layer is a special case of SPP which only takes a single scale (i.e., only one value of N for the $N \times N$ grid) to partition the proposal into fixed number of divisions, and also backpropagated error signals to the convolution kernels. After feature extraction, feature vectors were fed into a sequence of fully connected layers before two sibling output layers: classification layer (cls) and regression layer (reg). Classification layer was responsible for generating softmax probabilities over C+1 classes (C classes plus one background class), while regression layer encoded 4 real-valued parameters to refine bounding boxes. In Fast RCNN, the feature extraction, region classification and bounding box regression steps can all be optimized end-to-end, without extra cache space to store features (unlike SPP Net). Fast R-CNN achieved a much better detection accuracy than R-CNN and SPP-net, and had a better training and inference speed.

Despite the progress in learning detectors, the proposal generation step still relied on traditional methods such as Selective Search [45] or Edge Boxes [48], which were based on low-level visual cues and could not be learned in a data-driven manner. To address this issue, **Faster R-CNN** [34] was developed which relied on a novel proposal generator: Region Proposal Network (RPN). This proposal generator could be learned via supervised learning methods. RPN is a fully convolutional network which takes an image of arbitrary size and generates a set of object proposals on each position of the feature map. The network slid over the feature map using an $n \times n$ sliding window, and generated a feature vector for each position. The feature vector was then fed into two sibling output branches, object classification layer (which classified whether the proposal was an object or not) and bounding box regression layer. These results were then fed into the final layer for the actual object classification and bounding box localization. RPN could be inserted into Fast R-CNN and thus the whole framework could be optimized in an end-to-end manner on training data. This way RPN enabled proposal generation in a data driven manner, and was also able to enjoy the discriminative power of deep backbone networks. Faster R-CNN was able to make predictions at 5FPS on GPU and achieved state-of-the-art results on many public benchmark datasets, such as Pascal VOC 2007, 2012 and MSCOCO. Currently, there are huge number of detector variants based on Faster R-CNN for different usage [39,49–51].

Faster R-CNN computed feature map of the input image and extracted region features on the feature map, which shared feature extraction computation across different regions. However, the computation was not shared in the region classification step, where each feature vector still needed to go through a sequence of FC layers separately. Such extra computation could be extremely large as each image may have hundreds of proposals. Simply removing the fully connected layers would result in the drastic decline of detection performance, as the deep network would have reduced the spatial information of proposals. Dai et al. [52] proposed Region-based Fully Convolutional Networks (**R-FCN**) which shared the computation cost in the region classification step. R-FCN generated a Position Sensitive Score Map which encoded relative position information of different classes, and used a Position Sensitive ROI Pooling layer (PSROI Pooling) to extract spatial-aware region features by encoding each relative position of the target regions. The extracted feature vectors maintained spatial information and thus the detector achieved competitive results compared to Faster R-CNN without region-wise fully connected layer operations.

Another issue with Faster R-CNN was that it used a single deep layer feature map to make the final prediction. This made it difficult to detect objects at different scales. In particular, it was difficult to detect small objects. In DCNN feature representations, deep layer features are semantically-strong but spatially-weak, while shallow layer features are semantically-weak but spatially-strong. Lin et al. [39] exploited this property and proposed Feature Pyramid Networks (**FPN**) which combined deep layer features with shallow layer features to enable object detection in feature maps at different scales. The main idea was to strengthen the spatially strong shallow layer features with rich semantic information from the deeper layers. FPN achieved significant progress in detecting multi-scale objects and has been widely used in many other domains such as video detection [53,54] and human pose recognition [55,56].

Most instance segmentation algorithms are extended from vanilla object detection algorithms. Early methods [57–59] commonly generated segment proposals, followed by Fast RCNN for segments classification. Later, Dai et al. [59] proposed a multi-stage algorithm named "MNC" which divided the whole detection framework into multiple stages and predicted segmentation masks from the learned bounding box proposals, which were later categorized by region classifiers. These early works performed bbox and mask prediction in multiple stages. To make the whole process more flexible, He et al. [3] proposed **Mask R-CNN**, which predicted bounding boxes and segmentation masks in parallel based on the proposals and reported state-of-the-art results. Based on Mask R-CNN, Huang et al. [60] proposed a mask-quality aware framework,

Please cite this article as: X. Wu, D. Sahoo and S.C.H. Hoi, Recent advances in deep learning for object detection, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.01.085

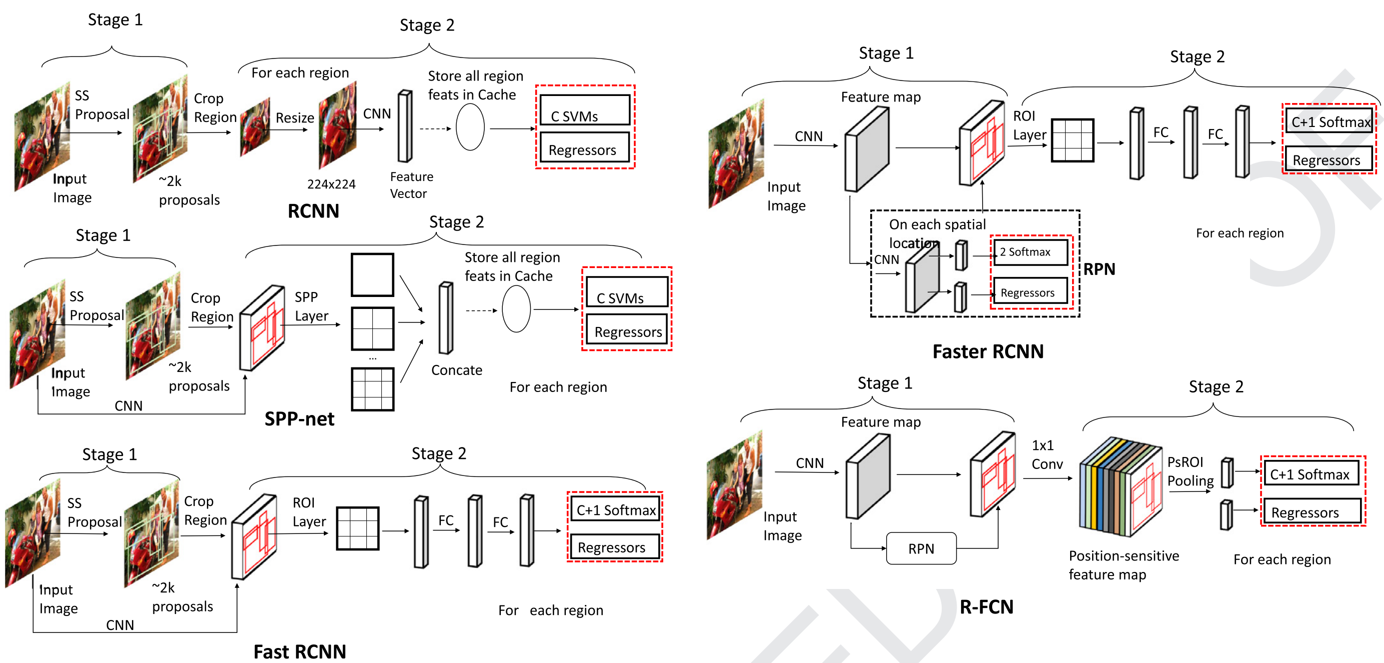


Fig. 4. Overview of different two-stage detection frameworks for generic object detection. Red dotted rectangles denote the outputs that define the loss functions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

named Mask Scoring R-CNN, which learned the quality of the predicted masks and calibrated the misalignment between mask quality and mask confidence score.

Fig. 4 gives an overview of the detection frameworks for several representative two-stage detectors.

3.2.2. One-stage detectors

Different from two-stage detection algorithms which divide the detection pipeline into two parts: proposal generation and region classification; one-stage detectors do not have a separate stage for proposal generation (or learning a proposal generation). They typically consider all positions on the image as potential objects, and try to classify each region of interest as either background or a target object.

One of the early successful one-stage detectors based on deep learning was developed by Sermanet et al. [61] named **OverFeat**. OverFeat performed object detection by casting DCNN classifier into a fully convolutional object detector. Object detection can be viewed as a "multi-region classification" problem, and thus OverFeat extended the original classifier into detector by viewing the last FC layers as 1x1 convolutional layers to allow arbitrary input. The classification network output a grid of predictions on each region of the input to indicate the presence of an object. After identifying the objects, bounding box regressors were learned to refine the predicted regions based on the same DCNN features of classifier. In order to detect multi-scale objects, the input image was resized into multiple scales which were fed into the network. Finally, the predictions across all the scales were merged together. OverFeat showed significant speed strength compared with RCNN by sharing the computation of overlapping regions using convolutional layers, and only a single pass forward through the network was required. However, the training of classifiers and regressors were separated without being jointly optimized.

Later, Redmon etal. [40] developed a real-time detector called **YOLO** (You Only Look Once). YOLO considered object detection as a regression problem and spatially divided the whole image into fixed number of grid cells (e.g. using a 7×7 grid). Each cell was considered as a proposal to detect the presence of one or more objects. In the original implementation, each cell was considered to contain the center of (upto) two objects. For each cell, a prediction was made which comprised the following information: whether that location had an object, the bounding box coordinates and size (width and height), and the class of the object. The whole framework was a single network and it omitted proposal generation step which could be optimized in an end-to-end manner. Based on a carefully designed lightweight architecture, YOLO could make prediction at 45 FPS, and reach 155 FPS with a more simplified backbone. However, YOLO faced some challenges: (i) it could detect upto only two objects at a given location, which made it difficult to detect small objects and crowded objects [40]. (ii) only the last feature map was used for prediction, which was not suitable for predicting objects at multiple scales and aspect ratios.

In 2016, Liu etal. proposed another one-stage detector Single-Shot Mulibox Detector (**SSD**) [42] which addressed the limitations of YOLO. SSD also divided images into grid cells, but in each grid cell, a set of anchors with multiple scales and aspect-ratios were generated to discretize the output space of bounding boxes (unlike predicting from fixed grid cells adopted in YOLO). Each anchor was refined by 4-value offsets learned by the regressors and was assigned (C+1) categorical probabilities by the classifiers. In addition, SSD predicted objects on multiple feature maps, and each of these feature maps was responsible for detecting a certain scale of objects according to its receptive fields. In order to detect large objects and increase receptive fields, several extra convolutional feature maps were added to the original backbone architecture. The whole network was optimized with a weighted sum of localization loss and classification loss over all prediction maps via an end-to-end training scheme. The final prediction was made by merging all detection results from different feature maps. In order to avoid huge number of negative proposals dominating training gradients, hard negative mining was used to train the detector. Intensive data augmentation was also applied to improve detection accuracy. SSD achieved comparable detection accuracy with Faster R-CNN but enjoyed the ability to do real-time inference.

Without proposal generation to filter easy negative samples, the class imbalance between foreground and background is a severe

Please cite this article as: X. Wu, D. Sahoo and S.C.H. Hoi, Recent advances in deep learning for object detection, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.01.085

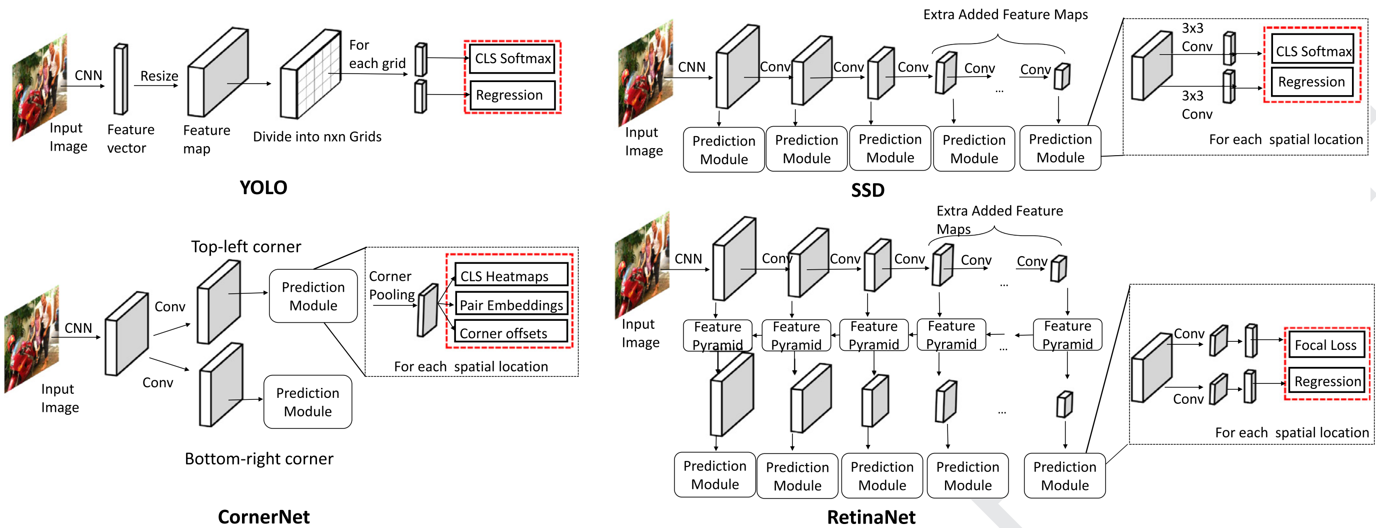


Fig. 5. Overview of different one-stage detection frameworks for generic object detection. Red rectangles denotes the outputs that define the objective functions.

problem in one-stage detector. Lin et al. [43] proposed a one-stage detector **RetinaNet** which addressed class imbalance problem in a more flexible manner. RetinaNet used focal loss which suppressed the gradients of easy negative samples instead of simply discarding them. Further, they used feature pyramid networks to detect multi-scale objects at different levels of feature maps. Their proposed focal loss outperformed naive hard negative mining strategy by large margins.

Redmon et al. proposed an improved YOLO version, **YOLOv2** [41] which significantly improved detection performance but still maintained real-time inference speed. YOLOv2 adopted a more powerful deep convolutional backbone architecture which was pre-trained on higher resolution images from ImageNet (from 224×224 to 448×448), and thus the weights learned were more sensitive to capturing fine-grained information. In addition, inspired by the anchor strategy used in SSD, YOLOv2 defined better anchor priors by k-means clustering from the training data (instead of setting manually). This helped in reducing optimizing difficulties in localization. Finally integrating with Batch Normalization layers [62] and multi-scale training techniques, YOLOv2 achieved state-of-the-art detection results at that time.

The previous approaches required designing anchor boxes manually to train a detector. Later a series of anchor-free object detectors were developed, where the goal was to predict keypoints of the bounding box, instead of trying to fit an object to an anchor. Law and Deng proposed a novel anchor-free framework **CornerNet** [63] which detected objects as a pair of corners. On each position of the feature map, class heatmaps, pair embeddings and corner offsets were predicted. Class heatmaps calculated the probabilities of being corners, and corner offsets were used to regress the corner location. And the pair embeddings served to group a pair of corners which belong to the same objects. Without relying on manually designed anchors to match objects, CornerNet obtained significant improvement on MSCOCO datasets. Later there were several other variants of keypoint detection based one-stage detectors [64,65].

Fig. 5 gives an overview of different detection frameworks for several representative one-stage detectors.

3.3. Backbone architecture

R-CNN [2] showed adopting convolutional weights from models pre-trained on large scale image classification problem could provide richer semantic information to train detectors and enhanced the detection performance. During the later years, this approach had become the default strategy for most object detectors. In this section, we will first briefly introduce the basic concept of deep convolutional neural networks and then review some architectures which are widely used for detection.

3.3.1. Basic architecture of a CNN

Deep convolutional neural network (DCNN) is a typical deep neural network and has proven extremely effective in visual understanding [33,36]. Deep convolutional neural networks are commonly composed of a sequence of convolutional layers, pooling layers, nonlinear activation layers and fully connected layers (FC layers). Convolutional layer takes an image input and convolves over it by $n \times n$ kernels to generate a feature map. The generated feature map can be regarded as a multi-channel image and each channel represents different information about the image. Each pixel in the feature map (named neuron) is connected to a small portion of adjacent neurons from the previous map, which is called the receptive field. After generating feature maps, a non-linear activation layer is applied. Pooling layers are used to summarize the signals within the receptive fields, to enlarge receptive fields as well as reduce computation cost,.

With the combination of a sequence of convolutional layers, pooling layers and non-linear activation layers, the deep convolutional neural network is built. The whole network can be optimized via a defined loss function by gradient-based optimization method (stochastic gradient descent [66], Adam [67], etc.). A typical convolutional neural network is AlexNet [33], which contains five convolutional layers, three max-pooling layers and three fully connected layers. Each convolutional layer is followed by a ReLU [68] non-linear activation layer.

3.3.2. CNN Backbone for object detection

In this section, we will review some architectures which are widely used in object detection tasks with state-of-the-art results, such as VGG16 [34,38], ResNet [1,52], ResNeXt [43] and Hourglass [63].

VGG16 [69] was developed based on AlexNet. VGG16 is composed of five groups of convolutional layers and three FC layers. There are two convolutional layers in the first two groups and three convolutional layers in the next three groups. Between each group, a Max Pooling layer is applied to decrease spatial dimension. VGG16 showed that increasing depth of networks by stacking

Please cite this article as: X. Wu, D. Sahoo and S.C.H. Hoi, Recent advances in deep learning for object detection, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.01.085

convolutional layers could increase the model's expression capability, and led to a better performance. However, increasing model depth to 20 layers by simply stacking convolutional layers led to optimization challenges with SGD. The performance declined significantly and was inferior to shallower models, even during the training stages. Based on this observation, He et al. [1] proposed ResNet which reduced optimization difficulties by introducing shortcut connections. Here, a layer could skip the nonlinear transformation and directly pass the values to the next layer as is (thus giving us an implicit identity layer). This is given as:

$$x_{l+1} = x_l + f_{l+1}(x_l, \theta) \tag{6}$$

where x_l is the input feature in l-th layer and f_{l+1} denotes operations on input x_l such as convolution, normalization or non-linear activation. $f_{l+1}(x_l, \theta)$ is the residual function to x_l, so the feature map of any deep layer can be viewed as the sum of the activation of shallow layer and the residual function. Shortcut connection creates a highway which directly propagates the gradients from deep layers to shallow units and thus, significantly reduces training difficulty. With residual blocks effectively training networks, the model depth could be increased (e.g. from 16 to 152), allowing us to train very high capacity models. Later, He et al. [70] proposed a pre-activation variant of ResNet, named ResNet-v2. Their experiments showed appropriate ordering of the Batch Normalization [62] could further perform better than original ResNet. This simple but effective modification of ResNet made it possible to successfully train a network with more than 1000 layers, and still enjoyed improved performance due to the increase in depth. Huang et al. argued that although ResNet reduced the training difficulty via shortcut connection, it did not fully utilize features from previous layers. The original features in shallow layers were missing in element-wise operation and thus could not be directly used later. They proposed DenseNet [71], which retained the shallow layer features, and improved information flow, by concatenating the input with the residual output instead of element-wise addition:

$$x_{l+1} = x_l \circ f_{l+1}(x_l, \theta) \tag{7}$$

where $\circ$ denotes concatenation. Chen [72] et al. argued that in DenseNet, the majority of new exploited features from shallow layers were duplicated and incurred high computation cost. Integrating the advantages of both ResNet and DenseNet, they propose a Dual Path Network (DPN) which divides x_l channels into two parts: x_l^d and x_l^r. x_l^d was used for dense connection computation and x_l^r was used for element-wise summation, with unshared residual learning branch f_{l+1}^d and f_{l+1}^r. The final result was the concatenated output of the two branches:

$$x_{l+1} = (x_l^r + f_{l+1}^r(x_l^r, \theta^r)) \circ (x_l^d \circ f_{l+1}^d(x_l^d, \theta^d)) \tag{8}$$

Based on ResNet, Xie et al. [73] proposed ResNeXt which considerably reduced computation and memory cost while maintaining comparable classification accuracy. ResNeXt adopted group convolution layers [33] which sparsely connects feature map channels to reduce computation cost. By increasing group number to keep computation cost consistent to the original ResNet, ResNeXt captures richer semantic feature representation from the training data and thus improves backbone accuracy. Later, Howard et al. [74] set the coordinates equal to number of channels of each feature map and developed MobileNet. MobileNet significantly reduced computation cost as well as number of parameters without significant loss in classification accuracy. This model was specifically designed for usage on a mobile platform.

In addition to increasing model depth, some efforts explored benefits from increasing model width to improve the learning capacity. Szegedy et al. proposed GoogleNet with an inception module [75] which applied different scale convolution kernels (1×1, 3×3 and 5×5) on the same feature map in a given layer. This way it captured multi-scale features and summarized these features together as an output feature map. Better versions of this model were developed later with different design of choice of convolution kernels [76], and introducing residual blocks [77].

The network structures introduced above were all designed for image classification. Typically these models trained on ImageNet are adopted as initialization of the model used for object detection. However, directly applying this pre-trained model from classification to detection is sub-optimal due to a potential conflict between classification and detection tasks. Specifically, (i) classification requires large receptive fields and wants to maintain spatial invariance. Thus multiple downsampling operation (such as pooling layer) are applied to decrease feature map resolution. The feature maps generated are low-resolution and spatially invariant and have large receptive fields. However, in detection, high-resolution spatial information is required to correctly localize objects; and (ii) classification makes predictions on a single feature map, while detection requires feature maps with multiple representations to detect objects at multiple scales. To bridge the difficulties between the two tasks, Li et al. introduced DetNet [78] which was designed specifically for detection. DetNet kept high resolution feature maps for prediction with dilated convolutions to increase receptive fields. In addition, DetNet detected objects on multi-scale feature maps, which provided richer information. DetNet was pre-trained on large scale classification dataset while the network structure was designed for detection.

Hourglass Network [79] is another architecture, which was not designed specifically for image classification. Hourglass Network first appeared in human pose recognition task [79], and was a fully convolutional structure with a sequence of hourglass modules. Hourglass module first downsampled the input image via a sequence of convolutional layer or pooling layer, and upsampled the feature map via deconvolutional operation. To avoid information loss in downsampling stage, skip connection were used between downsampling and upsampling features. Hourglass module could capture both local and global information and thus was very suitable for object detection. Currently Hourglass Network is widely used in state-of-the-art detection frameworks [63–65].

3.4. Proposal generation

Proposal generation plays a very important role in the object detection framework. A proposal generator generates a set of rectangle bounding boxes, which are potentially objects. These proposals are then used for classification and localization refinement. We categorize proposal generation methods into four categories: traditional computer vision methods, anchor-based supervised learning methods, keypoint based methods and other methods. Notably, both one-stage detectors and two-stage detectors generate proposals, the main difference is two-stage detectors generates a sparse set of proposals with only foreground or background information, while one-stage detectors consider each region in the image as a potential proposal, and accordingly estimates the class and bounding box coordinates of potential objects at each location.

3.4.1. Traditional computer vision methods

These methods generate proposals in images using traditional computer vision methods based on low-level cues, such as edges, corners, color, etc. These techniques can be categorized into three principles: (i) computing the 'objectness score' of a candidate box; (ii) merging super-pixels from original images; (iii) generating multiple foreground and background segments;

Objectness Score based methods predict an objectness score of each candidate box measuring how likely it may contain an object. Arbelaez et al. [80] assigned objectness score to proposals by classification based on visual cues such as color contrast, edge

Please cite this article as: X. Wu, D. Sahoo and S.C.H. Hoi, Recent advances in deep learning for object detection, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.01.085

density and saliency. Rahtu et al. [81] revisited the idea of Arbelaez et al. [80] and introduced a more efficient cascaded learning method to rank the objectness score of candidate proposals.

Superpixels merging is based on merging superpixels generated from segmentation results. Selective Search [45] was a proposal generation algorithm based on merging super-pixels. It computed the multiple hierarchical segments generated by segmentation method [82], which were merged according to their visual factors (color, areas, etc.), and finally bounding boxes were placed on the merged segments. Manen et al. [83] proposed a similar idea to merge superpixels. The difference was that the weight of the merging function was learned and the merging process was randomized. Selective Search is widely used in many detection frameworks due to its efficiency and high recall compared to other traditional methods.

Seed segmentation starts with multiple seed regions, and for each seed, foreground and background segments are generated. To avoid building up hierarchical segmentation, CPMC [84] generated a set of overlapping segments initialized with diverse seeds. Each proposal segment was the solution of a binary (foreground or background) segmentation problem. Enreds and Hoiem [85] combined the idea of Selective Search [45] and CPMC [84]. It started with super-pixels and merged them with new designed features. These merged segments were used as seeds to generate larger segments, which was similar to CPMC. However, producing high quality segmentation masks is very time-consuming and it's not applicable to large scale datasets.

The primary advantage of these traditional computer vision methods is that they are very simple and can generate proposals with high recall (e.g. on medium scale datasets such as Pascal VOC). However, these methods are mainly based on low level visual cues such as color or edges. They cannot be jointly optimized with the whole detection pipeline. Thus they are unable to exploit the power of large scale datasets to improve representation learning. On challenging datasets such as MSCOCO [86], traditional computer vision methods struggled to generate high quality proposals due to these limitations.

3.4.2. Anchor-based methods

One large family of supervised proposal generators is anchor-based methods. They generate proposals based on pre-defined anchors. Ren et al. proposed Region Proposal Network (RPN) [34] to generate proposals in a supervised way based on deep convolutional feature maps. The network slid over the entire feature map using 3×3 convolution filters. For each position, k anchors (or initial estimates of bounding boxes) of varying size and aspect ratios were considered. These sizes and ratios allowed for matching objects at different scales in the entire image. Based on the ground truth bonding boxes, the object locations were matched with the most appropriate anchors to obtain the supervision signal for the anchor estimation. A 256−dimensional feature vector was extracted from each anchor and was fed into two sibling branches - classification layer and regression layer. Classification branch was responsible for modeling objectness score while regression branch encoded four real-values to refine location of the bounding box from the original anchor estimation. Based on the ground truth, each anchor was predicted to either be an object, or just background by the classification branch (See Fig. 6). Later, SSD [42] adopted a similar idea of anchors in RPN by using multi-scale anchors to match objects. The main difference was that SSD assigned categorical probabilities to each anchor proposal, while RPN first evaluated whether the anchor proposal was foreground or background and performed the categorical classification in the next stage.

Despite promising performance, the anchor priors are manually designed with multiple scales and aspect ratios in a heuristic manner. These design choices may not be optimal, and different datasets would require different anchor design strategies. Many efforts have been made to improve the design choice of anchors. Zhang et al. proposed Single Shot Scale-invariant Face Detector (S3FD) [87] based on SSD with carefully designed anchors to match the objects. According to the effective receptive field [88] of different feature maps, different anchor priors were designed. Zhu et al. [89] introduced an anchor design method for matching small objects by enlarging input image size and reducing anchor strides. Xie et al. proposed Dimension-Decomposition Region Proposal Network (DeRPN) [90] which decomposed the dimension of anchor boxes based on RPN. DeRPN used an anchor string mechanism to independently match objects width and height. This helped match objects with large scale variance and reduced the searching space.

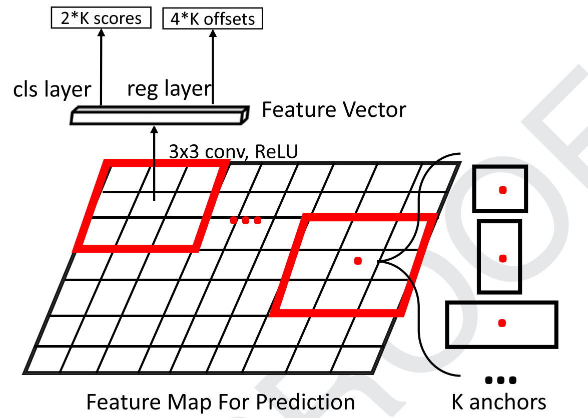


Fig. 6. Diagram of RPN [34]. Each position of the feature map connects with a sliding windows, followed with two sibling branches.

Ghodrati et al. developed DeepProposals [91] which predicted proposals on the low-resolution deeper layer feature map. These were then projected back onto the high-resolution shallow layer feature maps, where they are further refined. Redmon et al. [41] designed anchor priors by learning priors from the training data using k-means clustering. Later, Zhang et al. introduced Single-Shot Refinement Neural Network (RefineDet) [92] which refined the manually defined anchors in two steps. In the first step, RefineDet learned a set of localization offsets based on the original hand-designed anchors and these anchors were refined by the learned offsets. In the second stage, a new set of localization offsets were learned based on the refined anchors from the first step for further refinement. This cascaded optimization framework significantly improved the anchor quality and final prediction accuracy in a data-driven manner. Cai et al. proposed Cascade R-CNN [49] which adopted a similar idea as RefineDet by refining proposals in a cascaded way. Yang et al. [93] modeled anchors as functions implemented by neural networks which was computed from customized anchors. Their method MetaAnchor showed comprehensive improvement compared to other manually defined methods but the customized anchors were still designed manually.

3.4.3. Keypoints-based methods

Another proposal generation approach is based on keypoint detection, which can be divided into two families: corner-based methods and center-based methods. *Corner-based methods* predict bounding boxes by merging pairs of corners learned from the feature map. Denet [94] reformulated the object detection problem in a probabilistic way. For each point on the feature map, Denet modeled the distribution of being one of the 4 corner types of objects (top-left, top-right, bottom-left, bottom-right), and applied a naive bayesian classifiers over each corner of the objects to estimate the confidence score of a bounding box. This corner-based algorithm eliminated the design of anchors and became a more effective method to produce high quality proposals. Later based on Denet, Law and Deng proposed CornerNet [63] which directly modeled categorical information on corners. CornerNet modeled information of top-left and bottom-right corners with novel feature embedding methods and corner pooling layer to correctly match keypoints belonging to the same objects, obtaining state-of-the-art results on public benchmarks. For *center-based methods*, the probability of being the center of the objects is predicted on each position of the feature map, and the height and width are directly regressed without any anchor priors. Zhu et al. [95] presented a feature-selection-anchor-free (FSAF) framework which could be plugged into one-stage detectors with FPN structure. In FSAF, an online feature selection block is applied to train multi-level center-based branches attached in each level of the feature pyramid. During training, FSAF dynamically assigned each object to the most suitable feature level to train the center-based branch. Similar to FSAF, Zhou et al. proposed a new center-based framework [64] based on a single Hourglass network [63] without FPN structure. Furthermore, they applied center-based method into higher-level problems such as 3D-detection and human pose recognition, and all achieved state-of-the-art results. Duan et al. [65] proposed CenterNet, which combined the idea of center-based methods and corner-based methods. CenterNet first predicted bounding boxes by pairs of corners, and then predicted center probabilities of the initial prediction to reject easy negatives. CenterNet obtained significant improvements compared with baselines. These anchor-free methods form a promising research direction in the future.

3.4.4. Other methods

There are some other proposal generation algorithms which are not based on keypoints or anchors but also offer competitive performances. Lu et al. proposed AZnet [96] which automatically focused on regions of high interest. AZnet adopted a search strategy that adaptively directed computation resources to sub-regions which were likely contain objects. For each region, AZnet predicted two values: zoom indicator and adjacency scores. Zoom indicator determined whether to further divide this region which may contain smaller objects and adjacency scores denoted its objectness. The starting point was the entire image and each divided sub-region is recursively processed in this way until the zoom indicator is too small. AZnet was better at matching sparse and small objects compared to RPN's anchor-object matching approach.

3.5. Feature representation learning

Feature Representation Learning is a critical component in the whole detection framework. Target objects lie in complex environments and have large variance in scale and aspect ratios. There is a need to train a robust and discriminative feature embedding of objects to obtain a good detection performance. In this section, we introduce feature representation learning strategies for object detection. Specifically, we identify three categories: multi-scale feature learning, contextual reasoning, and deformable feature learning.

3.5.1. Multi-scale feature learning

Typical object detection algorithms based on deep convolutional networks such as Fast R-CNN [38] and Faster R-CNN [34] use only a single layer's feature map to detect objects. However, detecting objects across large range of scales and aspect ratios is quite challenging on a single feature map. Deep convolutional networks learn hierarchical features in different layers which capture different scale information. Specifically, shallow layer features with spatial-rich information have higher resolution and smaller receptive fields and thus are more suitable for detecting small objects, while semantic-rich features in deep layers are more robust to illumination, translation and have larger receptive fields (but coarse resolutions), and are more suitable for detecting large objects. When detecting small objects, high resolution representations are required and the representation of these objects may not even be available in the deep layer features, making small object detection difficult. Some techniques such as dilated/atrous convolutions [52,97] were proposed to avoid downsampling, and used the high resolution information even in the deeper layers. At the same time, detecting large objects in shallow layers are also non-optimal without large enough receptive fields. Thus, handling feature scale issues has become a fundamental research problem within object detection. There are four main paradigms addressing multi-scale feature learning problem: Image Pyramid, Prediction Pyramid, Integrated Features and Feature Pyramid. These are briefly illustrated in the Fig. 7.

Image pyramid: An intuitive idea is to resize input images into a number of different scales (Image Pyramid) and to train multiple detectors, each of which is responsible for a certain range of scales [98–101]. During testing, images are resized to different scales followed by multiple detectors and the detection results are merged. This can be computationally expensive. Liu et al. [101] first learned a light-weight scale-aware network to resize images such that all objects were in a similar scale. This was followed by learning a single scale detector. Singh et. al. [98] conducted comprehensive experiments on small object detection. They argued that learning a single scale-robust detector to handle all scale objects was much more difficult than learning scale-dependent detectors with image pyramids. In their work, they proposed a novel framework Scale Normalization for Image Pyramids (SNIP) [98] which trained multiple scale-dependent detectors and each of them was responsible for a certain scale objects.

Integrated features: Another approach is to construct a single feature map by combining features in multiple layers and making final predictions based on the new constructed map [50,51,102–105]. By fusing spatially rich shallow layer features and semantic-rich deep layer features, the new constructed features contain rich information and thus can detect objects at different scales. These combinations are commonly achieved by using skip connections [1]. Feature normalization is required as feature norms of different layers have a high variance. Bell et al. proposed Inside-Outside Network (ION) [51] which cropped region features from different layers via ROI Pooling [38], and combined these multi-scale region features for the final prediction. Kong et. al. proposed HyperNet [50] which adopted a similar idea as IoN. They carefully designed high resolution hyper feature maps by integrating intermediate and shallow layer features to generate proposals and detect objects. Deconvolutional layers were used to up-sample deep layer feature maps and batch normalization layers were used to normalize input blobs in their work. The constructed hyper feature maps could also implicitly encode contextual information from different layers. Inspired by fine-grained classification algorithms which integrate high-order representation instead of exploiting simple first-order representations of object proposals, Wang et al. proposed a novel framework Multi-scale Location-aware Kernel Representation (MLKP) [103] which captured high-order statistics of proposal features and generated more discriminative feature representations efficiently. The combined feature representation was more descriptive and provides both semantic and spatial information for both classification and localization.

Prediction pyramid: Liu et al.'s SSD [42] combined coarse and fine features from multiple layers together. In SSD, predictions were made from multiple layers, where each layer was responsible for a certain scale of objects. Later, many efforts [106–108] followed this principle to detect multi-scale objects.

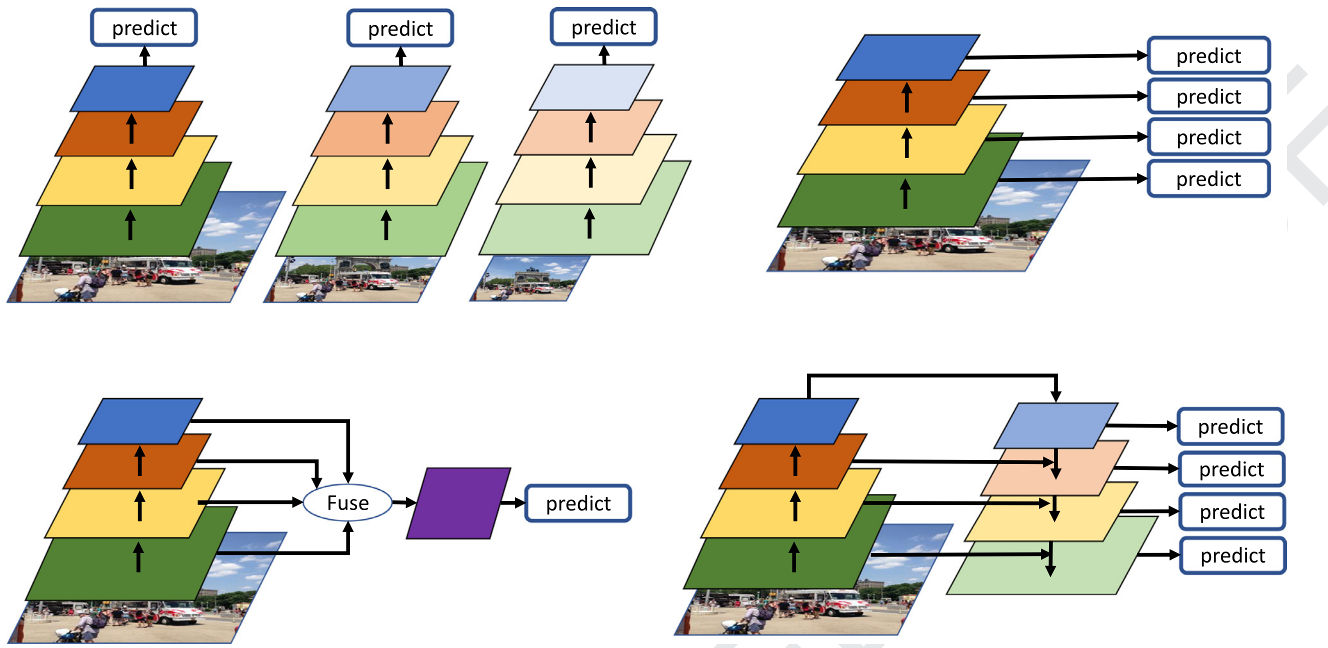


Fig. 7. Four paradigms for multi-scale feature learning. Top Left: *Image Pyramid*, which learns multiple detectors from different scale images; Top Right: *Prediction Pyramid*, which predicts on multiple feature maps; Bottom Left: *Integrated Features*, which predicts on single feature map generated from multiple features; Bottom Right: *Feature Pyramid* which combines the structure of *Prediction Pyramid* and *Integrated Features*.

Yang et al. [100] also exploited appropriate feature maps to generate certain scale of object proposals and these feature maps were fed into multiple scale-dependent classifiers to predict objects. In their work, cascaded rejection classifiers were learned to reject easy background proposals in early stages to accelerate detection speed. Multi-scale Deep Convolutional Neural Network (MSCNN) [106] applied deconvolutional layers on multiple feature maps to improve their resolutions, and later these refined feature maps were used to make predictions. Liu et al. proposed a Receptive Field Block Net (RFBNet) [108] to enhance the robustness and receptive fields via a receptive field block (RFB block). RFB block adopted similar ideas as the inception module [75] which captured features from multiple scale and receptive fields via multiple branches with different convolution kernels and finally merged them together.

Feature pyramid: To combine the advantage of Integrated Features and Prediction Pyramid, Lin et al. proposed Feature Pyramid Network (FPN) [39] which integrated different scale features with lateral connections in a top-down fashion to build a set of scale invariant feature maps, and multiple scale-dependent classifiers were learned on these feature pyramids. Specifically, the deep semantic-rich features were used to strengthen the shallow spatially-rich features. These top-down and lateral features were combined by element-wise summation or concatenation, with small convolutions reducing the dimensions. FPN showed significant improvement in object detection, as well as other applications, and achieved state-of-the art results in learning multi-scale features. Many variants of FPN were later developed [92,109,109–119], with modifications to the feature pyramid block (see Fig. 8). Kong et al. [120] and Zhang et. al. [92] built scale invariant feature maps with lateral connections. Different from FPN which generated region proposals followed by categorical classifiers, their methods omitted proposal generation and thus were more efficient than original FPN. Ren et al. [109] and Jeong et al. [110] developed a novel structure which gradually and selectively encoded contextual information between different layer features. Inspired by super resolution tasks [121,122], Zhou et al. [111] developed high resolution feature maps using a novel transform block which explicitly explored the inter-scale consistency nature across multiple detection scales.

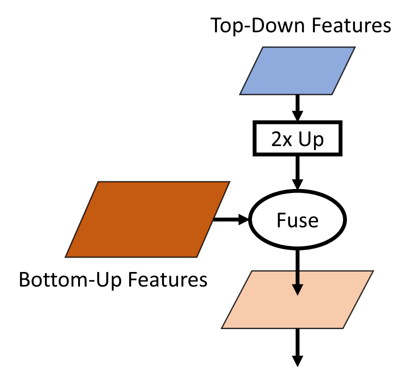


Fig. 8. General framework for feature combination. Top-down features are 2 times up-sampled and fuse with bottom-up features. The fuse methods can be element-wise sum, multiplication, concatenation and so on. Convolution and normalization layers can be inserted in to this general framework to enhance semantic information and reduce memory cost.

3.5.2. Region feature encoding

For two-stage detectors, region feature encoding is a critical step to extract features from proposals into fixed length feature vectors. In R-CNN, Girshick et al. [2] cropped region proposals from the whole image and resized the cropped regions into fixed sized patches (224×224) via bilinear interpolation, followed by a deep convolution feature extractor. Their method encoded high resolution region features but the computation was expensive.

Later Girshick et al. [38] and Ren [34] proposed ROI Pooling layer to encode region features. ROI Pooling divided each region into $n \times n$ cells (e.g. 7×7 by default) and only the neuron with the maximum signal would go ahead in the feedforward stage. This is similar to max-pooling, but across (potentially) different sized regions. ROI Pooling extracted features from the down-sampled feature map and as a result struggled to handle small objects. Dai [59] proposed ROI Warping layer which encoded region features via bilinear interpolation. Due to the downsampling operation in DCNN, there can be a misalignment of the object position in the original image and the downsampled feature maps, which RoI Pooling and RoI Warping layers are not able to handle. Instead of quantizing grids border as ROI Warping and ROI Pooling do, He et al. [3] proposed ROI Align layer which addressed the quantization issue by bilinear interpolation at fractionally sampled positions within each grid. Based on ROI Align, Jiang et al. [123] presented Precise ROI Pooing (PrROI Pooling), which avoided any quantization of coordinates and had a continuous gradient on bounding box coordinates.

In order to enhance spatial information of the downsampled region features, Dai et al. [52] proposed Position Sensitive ROI Pooing (PSROI Pooling) which kept relative spatial information of downsampled features. Each channel of generated region feature map only corresponded to a subset channels of input region according to its relative spatial position. Based on PSROI Pooling, Zhai et al. [124] presented feature selective networks to learn robust region features by exploiting disparities among sub-region and aspect ratios. The proposed network encoded sub-region and aspect ratio information which were selectively pooled to refine initial region features by a light-weight head.

Later, more algorithms were proposed to well encode region features from different viewpoints. Zhu et al. proposed CoupleNet [125] which extracted region features by combining outputs generated from both ROI Pooling layer and PSROI Pooling layer. ROI Pooling layer extracted global region information but struggled for objects with high occlusion while PSROI Pooling layer focused more on local information. CoupleNet enhanced features generated from ROI Pooling and PSROI Pooling by element-wise summation and generated more powerful features. Later Dai et al. proposed Deformable ROI Pooling [97] which generalized aligned RoI pooling by learning an offset for each grid and adding it to the grid center. The sub-grid start with a regular ROI Pooling layer to extract initial region features and the extracted features were used to regress offset by an auxiliary network. Deformable ROI Pooling can automatically model the image content without being constrained by fixed receptive fields.

3.5.3. *Contextual reasoning*

Contextual information plays an important role in object detection. Objects often tend to appear in specific environments and sometimes also coexist with other objects. For each example, birds commonly fly in the sky. Effectively using contextual information can help improve detection performance, especially for detecting objects with insufficient cues (small object, occlusion etc.) Learning the relationship between objects with their surrounding context can improve detector's ability to understand the scenario. For traditional object detection algorithms, there have been several efforts exploring context [126], but for object detection based on deep learning, context has not been extensively explored. This is because convolutional networks implicitly already capture contextual information from hierarchical feature representations. However, some recent efforts [1,3,3,59,106,127–131] still try to exploit contextual information. Some works [132] have even shown that in some cases context information may even harm the detection performance. In this section we review contextual reasoning for object detection from two aspects: *global context* and *region context*.

Global context reasoning refers to learning from the context in the whole image. Unlike traditional detectors which attempt to classify specific regions in the image as objects, the idea here is to use the contextual information (i.e., information from the rest of the image) to classify a particular region of interest. For example, detecting a baseball ball from an image can be challenging for a traditional detector (as it may be confused with balls from other sports); but if the contextual information from the rest of the image is used (e.g. baseball field, players, bat), it becomes easier to identify the baseball ball object.

Some representative efforts include ION [51], DeepId [127] and improved version of Faster R-CNN [1]. In ION, Bell et al. used recurrent neural network to encode contextual information across the whole image from four directions. Ouyang et al. [127] learned a categorical score for each image which is used as contextual features concatenated with the object detection results. He et al. [1] extracted feature embedding of the entire image and concatenate it with region features to improve detection results. In addition, some methods [3,59,129,133–136] exploit global contextual information via semantic segmentation. Due to precise pixel-level annotation, segmentation feature maps capture strong spatial information. He et al. [3] and Dai et al. [59] learn unified instance segmentation framework and optimize the detector with pixel-level supervision. They jointly optimized detection and segmentation objectives as a multi-task optimization. Though segmentation can significantly improve detection performance, obtaining the pixel-level annotation is very expensive. Zhao et al. [133] optimized detectors with pseudo segmentation annotation and showed promising results. Zhang et al.'s work Detection with Enriched Semantics (DES) [134], introduced contextual information by learning a segmentation mask without segemtation annotations. It also jointly optimized object detection and segmentation objectives and enriched original feature map with a more discriminative feature map.

Region Context Reasoning encodes contextual information surrounding regions and learns interactions between the objects with their surrounding area. Directly modeling different locations and categories objects relations with the contextual is very challenging. Chen et al. proposed Spatial Memory Network (SMN) [130] which introduced a spatial memory based module. The spatial memory module captured instance-level contexts by assembling object instances back into a pseudo "image" representations which were later used for object relations reasoning. Liu et al. proposed Structure Inference Net (SIN) [137] which formulated object detection as a graph inference problem by considering scene contextual information and object relationships. In SIN, each object was treated as a graph node and the relationship between different objects were regarded as graph edges. Hu et al. [138] proposed a lightweight framework relation network which formulated the interaction between different objects between their appearance and image locations. The new proposed framework did not need additional annotation and showed improvements in object detection performance. Based on Hu et al., Gu et al. [139] proposed a fully learnable object detector which proposed a general viewpoint that unified existing region feature extraction methods. Their proposed method removed heuristic choices in ROI pooling methods and automatically select the most significant parts, including contexts beyond proposals. Another method to encode contextual information is to implicitly encode region features by adding image features surrounding region proposals and a large number of approaches have been proposed based on this idea [106,131,140–143]. In addition to encode features from region proposals, Gidaris et al. [131] extracted features from a number of different sub-regions of the original object proposals (border regions, central regions, contextual regions etc.) and concatenated these features with the original region features. Similar to their method, [106] extracted local contexts by enlarging the proposal window size and concatenating these features with the original ones. Zeng et al. [142] proposed Gated Bi-Directional CNN (GBDNet) which extracted features from multi-scale subregions. Notably, GBDNet learned a gated function to control the transmission of different region in-

Please cite this article as: X. Wu, D. Sahoo and S.C.H. Hoi, Recent advances in deep learning for object detection, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.01.085

formation because not all contextual information is helpful for detection.

3.5.4. Deformable feature learning

A good detector should be robust to nonrigid deformation of objects. Before the deep learning era, Deformable Part based Models (DPMs) [28] had been successfully used for object detection. DPMs represented objects by multiple component parts using a deformable coding method, making the detector robust to nonrigid object transformation. In order to enable detectors based on deep learning to model deformations of object parts, many researchers have developed detection frameworks to explicitly model object parts [97,127,144,145]. DeepIDNet [127] developed a deformable-aware pooling layer to encode the deformation information across different object categories. Dai et al. [97] and Zhu et al. [144] designed deformable convolutional layers which automatically learned the auxiliary position offsets to augment information sampled in regular sampling locations of the feature map.

4. Learning strategy

In contrast to image classification, object detection requires optimizing both localization and classification tasks, which makes it more difficult to train robust detectors. In addition, there are several issues that need to be addressed, such as imbalance sampling, localization, acceleration etc. Thus there is a need to develop innovative learning strategies to train effective and efficient detectors. In this section, we review some of the learning strategies for object detection.

4.1. Training stage

In this section, we review the learning strategies for training object detectors. Specifically we discuss, data augmentation, imbalance sampling, cascade learning, localization refinement and some other learning strategies.

4.1.1. Data augmentation.

Data augmentation is important for nearly all deep learning methods as they are often data-hungry and more training data leads to better results. In object detection, in order to increase training data as well as generate training patches with multiple visual properties, Horizontal flips of training images is used in training Faster R-CNN detector [38]. A more intensive data augmentation strategy is used in one-stage detectors including rotation, random crops, expanding and color jittering [42,106,146]. This data augmentation strategy has shown significant improvement in detection accuracy.

4.1.2. Imbalance sampling

In object detection, imbalance of negative and positive samples is a critical issue. That is, most of the regions of interest estimated as proposals are in fact just background images. Very few of them are positive instances (or objects). This results in problem of imbalance while training detectors. Specifically, two issues arise, which need to be addressed: class imbalance and difficulty imbalance. The class imbalance issue is that most candidate proposals belong to the background and only a few of proposals contain objects. This results in the background proposals dominating the gradients during training. The difficulty imbalance is closely related to the first issue, where due to the class imbalance, it becomes much easier to classify most of the background proposals easily, while the objects become harder to classify. A variety of strategies have been developed to tackle the class imbalance issue. Two-stage detectors such as R-CNN and Fast R-CNN will first reject majority of negative samples and keep 2000 proposals for further classification. In Fast R-CNN [38], negative samples were randomly sampled from these 2k proposals and the ratio of positive and negative was fixed as 1:3 in each mini-batch, to further reduce the adverse effects of class imbalance. Random sample can address class imbalance issue but are not able to fully utilize information from negative proposals. Some negative proposals may contain rich context information about the images, and some hard proposals can help to improve detection accuracy. To address this, Liu et al. [42] proposed hard negative sampling strategy which fixed the foreground and background ratio but sampled most difficult negative proposals for updating the model. Specifically, negative proposals with higher classification loss were selected for training.

To address difficulty imbalance, most sampling strategies are based on carefully designed loss functions. For obejct detection, a *multi-class* classifier is learned over C+1 categories (C target categories plus one background category). Assume the region is labeled with ground truth class u, and p is the output discrete probability distribution over C+1 classes ($p = \{p_0, \dots, p_C\}$). The loss function is given by:

$$L_{\mathrm{cls}}(p, u) = -\log p_u \tag{9}$$

Lin et al. proposed a novel focal loss [43] which suppressed signals from easy samples. Instead of discarding all easy samples, they assigned an importance weight to each sample w.r.t its loss value as:

$$L_{\mathrm{FL}} = -\alpha (1 - p_u)^{\gamma} \log(p_u) \tag{10}$$

where α and γ were parameters to control the importance weight. The gradient signals of easy samples got suppressed which led the training process to focus more on hard proposals. Li et al. [147] adopt a similar idea from focal loss and propose a novel gradient harmonizing mechanism (GHM). The new proposed GHM not only suppressed easy proposals but also avoided negative impact of outliers. Shrivastava et al. [148] proposed an online hard example mining strategy which was based on a similar principle as Liu et al.'s SSD [42] to automatically select hard examples for training. Different from Liu et al., online hard negative mining only considered difficulty information but ignored categorical information, which meant the ratio of foreground and background was not fixed in each mini-batch. They argued that difficult samples played a more important role than class imbalance in object detection task.

4.1.3. Localization refinement

An object detector must provide a tight localization prediction (bbox or mask) for each object. To do this, many efforts refine the preliminary proposal prediction to improve the localization. Precise localization is challenging because predictions are commonly focused on the most discriminative part of the objects, and not necessarily the region containing the object. In some scenarios, the detection algorithms are required to make high quality predictions (high IoU threshold) See Fig. 9 for an illustration of how a detector may fail in a high IoU threshold regime. A general approach for localization refinement is to generate high quality proposals (See Section 3.4). In this section, we will review some other methods for localization refinement. In R-CNN framework, the L-2 auxiliary bounding box regressors were learned to refine localizations, and in Fast R-CNN, the smooth L1 regressors were learned via an end-to-end training scheme as:

$$L_{\mathrm{reg}}(t^c, v) = \sum_{i \in \{x,y,w,h\}} \mathrm{SmoothL1}(t_i^c - v_i) \tag{11}$$

$$\mathrm{SmoothL1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{12}$$

where the predicted offset is given by $t^c = (t_x^c, t_y^c, t_w^c, t_h^c)$ for each target class, and v denotes ground truth of object bounding

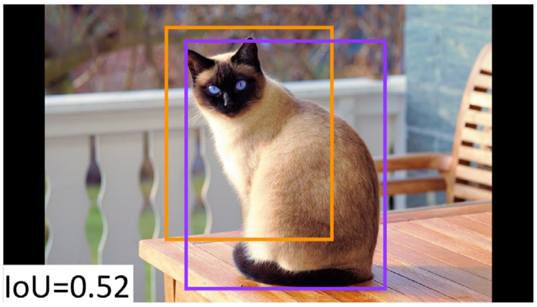


Fig. 9. Example of failure case of detection in high IoU threshold. Purple box is ground truth and yellow box is prediction. In low IoU requirement scenario, this prediction is correct while in high IoU threshold, it's a false positive due to insufficient overlap with objects. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

boxes($v = (v_x, v_y, v_w, v_h)$). x, y, w, h denote bounding box center, width and height respectively.

Beyond the default localization refinement, some methods learn auxiliary models to further refine localizations. Gidaris et al. [131] introduced an iterative bounding box regression method, where an R-CNN was applied to refine learned predictions. Here the predictions were refined multiple times. Gidaris et al. [149] proposed LocNet which modeled the distribution of each bounding box and refined the learned predictions. Both these approaches required a separate component in the detection pipeline, and prevent joint optimization.

Some other efforts [150,151] focus on designing a unified framework with modified objective functions. In MultiPath Network, Zagoruyko et al. [150] developed an ensemble of classifiers which were optimized with an integral loss targeting various quality metrics. Each classifier was optimized for a specific IoU threshold and the final prediction results were merged from these classifiers. Tychsen et al. proposed Fitness-NMS [152] which learned novel fitness score function of IoU between proposals and objects. They argued that existing detectors aimed to find *qualified* predictions instead of *best* predictions and thus highly quality and low quality proposals received equal importance. Fitness-IoU assigned higher importance to highly overlapped proposals. They also derived a bounding box regression loss based on a set of IoU upper bounds to maximum the IoU of predictions with objects. Inspired by CornerNet [63] and DeNet [94], Lu et al. [151] proposed a Grid R-CNN which replaced linear bounding box regressor with the principle of locating corner keypoints corner-based mechanism.

4.1.4. Cascade learning

Cascade learning is a coarse-to-fine learning strategy which collects information from the output of the given classifiers to build stronger classifiers in a cascaded manner. Cascade learning strategy was first used by Viola and Jones [17] to train the robust face detectors. In their models, a lightweight detector first rejects the majority easy negatives and feeds hard proposals to train detectors in next stage. For deep learning based detection algorithms, Yang et al. [153] proposed CRAFT (Cascade Region-proposal-network And FasT-rcnn) which learned RPN and region classifiers with a cascaded learning strategy. CRAFTS first learned a standard RPN followed by a two-class Fast RCNN which rejected the majority easy negatives. The remaining samples were used to build the cascade region classifiers which consisted of two Fast RCNNs. Yang et al. [100] introduced layer-wise cascade classifiers for different scale objects in different layers. Multiple classifiers were placed on different feature maps and classifiers on shallower layers would reject easy negatives. The remaining samples would be fed into deeper layers for classification. RefineDet [92] and Cascade R-CNN [49] utilized cascade learning methods in refining object locations. They built multi-stage bounding box regressors and bounding box predictions were refined in each stage trained with different quality metrics. Cheng et al. [132] observed the failure cases of Faster RCNN, and noticed that even though the localization of objects was good, there were several classification errors. They attributed this to sub-optimal feature representation due to sharing of features and joint multi-task optimization, for classification and regression; and they also argued that the large receptive field of Faster RCNN induce too much noise in the detection process. They found that vanilla RCNN was robust to these issues. Thus, they built a cascade detection system based on Faster RCNN and RCNN to complement each other. Specifically, A set of initial predictions were obtained from a well trained Faster RCNN, and these predictions were used to train RCNN to refine the results.

4.1.5. Others

There are some other learning strategies which offer interesting directions, but have not yet been extensively explored. We split these approaches into four categories: adversarial learning, training from scratch and knowledge distillation.

Adversarial learning. Adversarial learning has shown significant advances in generative models. The most famous work applying adversarial learning is generative adversarial network (GAN) [154] where a generator is competing with a discriminator. The generator tries to model data distribution by generating fake images using a noise vector input and use these fake images to confuse the discriminator, while the discriminator competes with the generator to identify the real images from fake images. GAN and its variants [155–157] have shown effectiveness in many domains and have also found applications in object detection. Li et al. [158] proposed a new framework Perceptual GAN for small object detection. The learnable generator learned high-resolution feature representations of small objects via an adversarial scheme. Specifically, its generator learned to transfer low-resolution small region features into high-resolution features and competed with the discriminator which identified real high-resolution features. Finally the generator learned to generate high quality features for small objects. Wang et al. [159] proposed A-Fast-R-CNN which was trained by generated adversarial examples. They argued the difficult samples were on long tail so they introduced two novel blocks which automatically generated features with occlusion and deformation. Specifically, a learned mask was generated on region features followed by region classifiers. In this case, the detectors could receive more adversarial examples and thus become more robust.

Training from scratch. Modern object detectors heavily rely on pre-trained classification models on ImageNet, however, the bias of loss functions and data distribution between classification and detection can have an adversarial impact on the performance. Finetuning on detection task can relieve this issue, but cannot fully get rid of the bias. Besides, transferring a classification model for detection in a new domain can lead to more challenges (from RGB to MRI data etc.). Due to these reasons, there is a need to train detectors from scratch, instead of relying on pretrained models. The main difficulty of training detectors from scratch is the training data of object detection is often insufficient and may lead to overfitting. Different from image classification, object detection requires bounding box level annotations and thus, annotating a large scale detection dataset requires much more effort and time (ImageNet has 1000 categories for image classification while only 200 of them have detection annotations).

There are some works [107,160,161] exploring training object detectors from scratch. Shen et al. [107] first proposed a novel framework DSOD (Deeply Supervised Object Detectors) to train detectors from scratch. They argued deep supervision with a densely connected network structure could significantly reduce op-

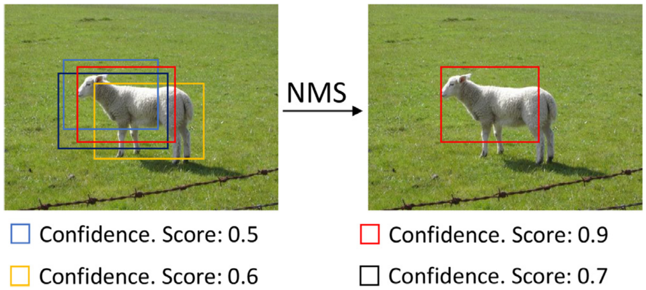


Fig. 10. Duplicate predictions are eliminated by NMS operation. The most-confident box is kept, and all other boxes surrounding it will be removed.

timization difficulties. Based on DSOD, Shen et al. [162] proposed a gated recurrent feature pyramid which dynamically adjusted supervision intensities of intermediate layers for objects with different scales. They defined a recurrent feature pyramid structure to squeeze both spatial and semantic information into a single prediction layer, which further reduced parameter numbers leading to faster convergence. In addition, the gate-control structure on feature pyramids adaptively adjusted the supervision at different scales based on the size of objects. Their method was more powerful than original DSOD. However, later He et al. [160] validated the difficulty of training detectors from scratch on MSCOCO and found that the vanilla detectors could obtain a competitive performance with at least 10K annotated images. Their findings proved no specific structure was required for training from scratch which contradicted the previous work.

Knowledge distillation. Knowledge distillation is a training strategy which distills the knowledge in an ensemble of models into a single model via teacher-student training scheme. This learning strategy was first used in image classification [163]. In object detection, some works [132,164] also investigate this training scheme to improve detection performance. Li et al. [164] proposed a light weight detector whose optimization was carefully guided by a heavy but powerful detector. This light detector could achieve comparable detection accuracy by distilling knowledge from the heavy one, meanwhile having faster inference speed. Cheng et al. [132] proposed a Faster R-CNN based detector which was optimized via teacher-student training scheme. An R-CNN model is used as teacher network to guide the training process. Their framework showed improvement in detection accuracy compared with traditional single model optimization strategy.

4.2. Testing stage

Object detection algorithms make a dense set of predictions and thus these predictions cannot be directly used for evaluation due to heavy duplication. In addition, some other learning strategies are required to further improve the detection accuracy. These strategies improve the quality of prediction or accelerate the inference speed. In this section, we introduce these strategies in testing stage including duplicate removal, model acceleration and other effective techniques.

4.2.1. Duplicate removal

Non maximum suppression (NMS) is an integral part of object detection to remove duplicate false positive predictions (See Fig. 10). Object detection algorithms make a dense set of predictions with several duplicate predictions. For one-stage detection algorithms which generate a dense set of candidate proposals such as SSD [42] or DSSD (Deconvolutional Single Shot Detector) [112], the proposals surrounding the same object may have similar confidence scores, leading to false positives. For two-stage detection algorithms which generates a sparse set of proposals, the bounding box regressors will pull these proposals close to the same object and thus lead to the same problem. The duplicate predictions are regarded as false positives and will receive penalties in evaluation, so NMS is needed to remove these duplicate predictions. Specifically, for each category, the prediction boxes are sorted according to the confidence score and the box with highest score is selected. This box is denoted as M. Then IoU of other boxes with M is calculated, and if the IoU value is larger than a predefined threshold Ω_{test}, these boxes will are removed. This process is repeated for all remaining predictions. More formally, the confidence score of box B which overlaps with M larger than Ω_{test} will be set to zero:

$$\text{Score}_B = \begin{cases} \text{Score}_B & \text{IoU}(B, M) < \Omega_{\text{test}} \\ 0 & \text{IoU}(B, M) \geq \Omega_{\text{test}} \end{cases} \tag{13}$$

However, if an object just lies within Ω_{test} of M, NMS will result in a missing prediction, and this scenario is very common in clustered object detection. Navaneeth et al. [165] introduced a new algorithm Soft-NMS to address this issue. Instead of directly eliminating the prediction B, Soft-NMS decayed the confidence score of B as a continuous function F(F can be linear function or Gaussian function) of its overlaps with M. This is given by:

$$\text{Score}_B = \begin{cases} \text{Score}_B & \text{IoU}(B, M) < \Omega_{\text{test}} \\ F(\text{IoU}(B, M)) & \text{IoU}(B, M) \geq \Omega_{\text{test}} \end{cases} \tag{14}$$

Soft-NMS avoided eliminating prediction of clustered objects and showed improvement in many common benchmarks. Hosong et al [166]. introduced a network architecture designed to perform NMS based on confidence scores and bounding boxes, which was optimized separately from detector training in a supervised way. They argued the reason for duplicate predictions was that the detector deliberately encouraged multiple high score detections per object instead of rewarding one high score. Based on this, they designed the network following two motivations: (i) a loss penalizing double detections to push detectors to predict exactly one precise detection per object; (ii) joint processing of detections nearby to give the detector information whether an object is detected more than once. The new proposed model did not discard detections but instead reformulated NMS as a re-scoring task that sought to decrease the score of detections that cover objects that already have been detected.

4.2.2. Model acceleration

Application of object detection for real world application requires the algorithms to function in an efficient manner. Thus, evaluating detectors on efficiency metrics is important. Although current state-of-the-art algorithms [1,167] can achieve very strong results on public datasets, their inference speeds make it difficult to apply them into real applications. In this section we review several works on accelerating detectors. Two-stage detectors are usually slower than one-stage detectors because they have two stages - one proposal generation and one region classification, which makes them computationally more time consuming than one-stage detectors which directly use one network for both proposal generation and region classification. R-FCN [52] built spatially-sensitive feature maps and extracted features with position sensitive ROI Pooling to share computation costs. However, the number of channels of spatially-sensitive feature maps significantly increased with the number of categories. Li et al. [168] proposed a new framework Light Head R-CNN which significantly reduced the number of channels in the final feature map (from 1024 to 16) instead of sharing all computation. Thus, though computation was not shared across regions, but the cost could be neglected.

From the aspect of backbone architecture, a major computation cost in object detection is feature extraction [34]. A simple idea to accelerate detection speed is to replace the detection backbone with a more efficient backbone, e.g., MobileNet [74,169] was an efficient CNN model with depth-wise convolution layers which was also adopted into many works such as [170] and [171]. PVANet [104] was proposed as a new network structure with CReLu [172] layer to reduce non-linear computation and accelerated inference speed. Another approach is to optimize models offline, such as model compression and quantization [173–179] on the learned models. Finally, NVIDIA Corporation[2] released an acceleration toolkit TensorRT[3] which optimized the computation of learned models for deployment and thus significantly sped up the inference.

4.2.3. Others

Other learning strategies in testing stage mainly comprise the transformation of input image to improve the detection accuracy. Image pyramids [1,92] are a widely used technique to improve detection results, which build a hierarchical image set at different scales and make predictions on all of these images. The final detection results are merged from the predictions of each image. Zhang et al. [87,92] used a more extensive image pyramid structure to handle different scale objects. They resized the testing image to different scales and each scale was responsible for a certain scale range of objects. Horizontal Flipping [3,92] was also used in the testing stage and also showed improvement. These learning strategies largely improved the capability of detector to handle different scale objects and thus were widely used in public detection competitions. However, they also increase computation cost and thus were not suitable for real world applications.

5. Applications

Object detection is a fundamental computer vision task and there are many real world applications based on this task. Different from generic object detection, these real world applications commonly have their own specific properties and thus carefully-designed detection algorithms are required. In this section, we will introduce several real world applications such as face detection and pedestrian detection.

5.1. Face detection

Face detection is a classical computer vision problem to detect human faces in the images, which is often the first step towards many real-world applications with human beings, such as face verification, face alignment and face recognition. There are some critical differences between face detection and generic detection: (i) the range of scale for objects in face detection is much larger than objects in generic detection. Moreover occlusion and blurred cases are more common in face detection; (ii) Face objects contain strong structural information, and there is only one target category in face detection. Considering these properties of face detection, directly applying generic detection algorithms is not an optimal solution as there could be some priors that can exploited to improve face detection.

In early stages of research before the deep learning era, face detection [20,180–182] was mainly based on sliding windows, and dense image grids were encoded by hand-crafted features followed by training classifiers to find and locate objects. Notably, Viola and Jones [20] proposed a pioneering cascaded classifiers using AdaBoost with Haar features for face detection and obtained excellent performance with high real time prediction speed. After the progresses of deep learning in image classification, face detectors based on deep learning significantly outperformed traditional face detectors [183–187].

Current face detection algorithms based on deep learning are mainly extended from generic detection frameworks such as Fast R-CNN and SSD. These algorithms focus more on learning robust feature representations. In order to handle extreme scale variance, multi-scale feature learning methods discussed before have been widely used in face detection. Sun et al. [183] proposed a Fast R-CNN based framework which integrated multi-scale features for prediction and converted the resulting detection bounding boxes into ellipses as the regions of human faces are more elliptical than rectangular. Zhang et al. [87] proposed one-stage S3FD which found faces on different feature maps to detect faces at a large range of scales. They made predictions on larger feature maps to capture small-scale face information. Notably, a set of anchors were carefully designed according to empirical receptive fields and thus provided a better match to the faces. Based on S3FD, Zhang et al. [188] proposed a novel network structure to capture multi-scale features in different stages. The new proposed feature agglomerate structure integrated features at different scales in a hierarchical way. Moreover, a hierarchical loss was proposed to reduce the training difficulties. Single Stage Headless Face Detector (SSH) [189] was another one-stage face detector which combined different scale features for prediction. Hu et al. [99] gave a detailed analysis of small face detection and proposed a light weight face detector consisting of multiple RPNs, each of which was responsible for a certain range of scales. Their method could effectively handle face scale variance but it was slow for real world usage. Unlike this method, Hao et al. [190] proposed a Scale Aware Face network which addresses scale issues without incurring significant computation costs. They learned a scale aware network which modeled the scale distribution of faces in a given image and guided zoom-in or zoom-out operations to make sure that the faces were in desirable scale. The resized image was fed into a single scale light weight face detector. Wang et al. [191] followed RetinaNet [43] and utilized more dense anchors to handle faces in a large range of scales. Moreover, they proposed an attention function to account for context information, and to highlight the discriminative features. Zhang et al. [192] proposed a deep cascaded multi-task face detector with cascaded structure (MTCNN). MTCNN had three stages of carefully designed CNN models to predict faces in a coarse-to-fine style. Further, they also proposed a new online hard negative mining strategy to improve the result. Samangouei et al. [193] proposed a Face MegNet which allowed information flow of small faces without any skip connections by placing a set of deconvolution layers before RPN and ROI Pooling to build up finer face representations.

In addition to multi-scale feature learning, some frameworks were focused on contextual information. Face objects have strong physical relationships with the surrounding contexts (commonly appearing with human bodies) and thus encoding contextual information became an effective way to improve detection accuracy. Zhang et al. [194] proposed FDNet based on ResNet with larger deformable convolutional kernels to capture image context. Zhu et al. [195] proposed a Contextual Multi-Scale Region-based Convolution Neural Network (CMS-RCNN) in which multi-scale information was grouped both in region proposal and ROI detection to deal with faces at various range of scale. In addition, contextual information around faces is also considered in training detectors. Notably, Tang et al. [185] proposed a state-of-the-art context assisted single shot face detector, named PyramidBox to handle the hard face detection problem. Observing the importance of the context, they improved the utilization of contextual information in the following three aspects: (i) first, a novel context anchor was designed to supervise high-level contextual feature learning by a semi-supervised method, dubbed as PyramidAnchors; (ii) the Low-level Feature Pyramid Network was developed to combine adequate high-level context semantic features and low-level facial

[2] https://www.nvidia.com/en-us/.
[3] https://developer.nvidia.com/tensorrt.

Please cite this article as: X. Wu, D. Sahoo and S.C.H. Hoi, Recent advances in deep learning for object detection, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.01.085

features together, which also allowed the PyramidBox to predict faces at all scales in a single shot; and (iii) they introduced a context sensitive structure to increase the capacity of prediction network to improve the final accuracy of output. In addition, they used the method of data-anchor-sampling to augment the training samples across different scales, which increased the diversity of training data for smaller faces. Yu et al. [196] introduced a context pyramid maxout mechanism to explore image contexts and devised an efficient anchor based cascade framework for face detection which optimized anchor-based detector in cascaded manner. Zhang et al. [197] proposed a two-stream contextual CNN to adaptively capture body part information. In addition, they proposed to filter easy non-face regions in the shallow layers and leave difficult samples to deeper layers.

Beyond efforts on designing scale-robust or context-assistant detectors, Wang et al. [191] developed a framework from the perspective of loss function design. Based on vanilla Faster R-CNN framework, they replaced original softmax loss with a center loss which encouraged detectors to reduce the large intra-class variance in face detection. They explored multiple technologies in improving Faster R-CNN such as fixed-ratio online hard negative mining, multi-scale training and multi-scale testing, which made vanilla Faster R-CNN adaptable to face detection. Later, Wang et al. [198] proposed Face R-FCN which was based on vanilla R-FCN. Face R-FCN distinguished the contribution of different facial parts and introduced a novel position-sensitive average pooling to re-weight the response on final score maps. This method achieved state-of-the-art results on many public benchmarks such as FDDB [199] and WIDER FACE [200].

5.2. Pedestrian detection

Pedestrian detection is an essential and significant task in any intelligent video surveillance system. Different from generic object detection, there are some properties of pedestrian detection different from generic object detection: (i) Pedestrian objects are well structured objects with nearly fixed aspect ratios (about 1.5), but they also lie at a large range of scales; (ii) Pedestrian detection is a real world application, and hence the challenges such as crowding, occlusion and blurring are commonly exhibited. For example, in the CityPersons dataset, there are a total of 3157 pedestrian annotations in the validation subset, among which 48.8% overlap with another annotated pedestrian with Intersection over Union (IoU) above 0.1. Moreover, 26.4% of all pedestrians have considerable overlap with another annotated pedestrian with IoU above 0.3. The highly frequent crowd occlusion harms the performance of pedestrian detectors; (iii) There are more hard negative samples (such as traffic light, Mailbox etc.) in pedestrian detection due to complicated contexts.

Before the deep learning era, pedestrian detection algorithms [19,201–204] were mainly extended from Viola Jones frameworks [20] by exploiting Integral Channel Features with a sliding window strategy to locate objects, followed by region classifiers such as SVMs. The early works were mainly focused on designing robust feature descriptors for classification. For example, Dalal and Triggs [19] proposed the histograms of oriented gradient (HOG) descriptors, while Paisitkriangkrai et al. [204] designed a feature descriptor based on low-level visual cues and spatial pooling features. These methods show promising results on pedestrian detection benchmarks but were mainly based on hand-crafted features.

Deep learning based methods for pedestrian detection [8–10,205–211] showed excellent performance and achieved state-of-the-art results on public benchmarks. Angelova et al [10] proposed a real-time pedestrian detection framework using a cascade of deep convolutional networks. In their work, a large number of easy negatives were rejected by a tiny model and the remaining hard proposals were then classified by a large deep networks. Zhang et al. [212] proposed a decision tree based framework. In their method, multiscale feature maps were used to extract pedestrian features, which were later fed into boosted decision trees for classification. In contrast to the FC layers, boosted decision trees applied a bootstrapping strategy for mining hard negative samples and achieved a better performance. Also to reduce the impact of large variance in scales, Li et al. [8] proposed Scale-aware Fast R-CNN (SAF RCNN) which inserted multiple built-in networks into the whole detection framework. The proposed SAF RCNN detected different scale pedestrian instances using different subnets. Further, Yang et al. [100] inserted Scale Dependent Pooling (SDP) and Cascaded Rejection Classifiers (CRC) into Fast RCNN to handle pedestrians at different scales. According to the height of the instances, SDP extracted region features from a suitable scale feature map, while CRC rejected easy negative samples in shallower layers. Wang et al. [213] proposed a novel Repulsion Loss to detect pedestrians in a crowd. They argued that detecting a pedestrian in a crowd made it very sensitive to the NMS threshold, which led to more false positives and missing objects. The new proposed repulsion loss pushed the proposals into their target objects but also pulled them away from other objects and their target proposals. Based on their idea, Zhang et al. [214] proposed an Occlusion-aware R-CNN (OR-CNN) which was optimized by an Aggression Loss. The new loss function encouraged the proposals to be close to the objects and other proposals with the same targeted proposals. Mao et al. [215] claimed that properly aggregating extra features into pedestrian detector could boost the detection accuracy. In their paper, they explored different kinds of extra features useful in improving accuracy and proposed a new method to use these features. The new proposed component - HyperLearner aggregated extra features into a vanilla DCNN detector in a jointly optimized fashion and no extra input was required for the inference stage.

For pedestrian detection, one of the most significant challenges is to handle occlusion [214,216–226]. A straightforward method is to use part-based models which learn a series of part detectors and integrate the results of part detectors to locate and classify objects. Tian et al. [216] proposed DeepParts which consisted of multiple part-based detectors. During training, the important pedestrian parts were automatically selected from a part pool which was composed of parts of the human body (at different scales), and for each selected part, a detector was learned to handle occlusions. To integrate the inaccurate scores of part-based models, Ouyang and Wang [223] proposed a framework which modeled visible parts as hidden variables in training the models. In their work, the visible relationship of overlapping parts were learned by discriminative deep models, instead of being manually defined or even being assumed independent. Later, Ouyang et al. [225] addressed this issue from another aspect. They proposed a mixture network to capture unique visual information which was formed by crowded pedestrians. To enhance the final predictions of single-pedestrian detectors, a probabilistic framework was learned to model the relationship between the configurations estimated by single-pedestrian and multi-pedestrian detectors. Zhang et al. [214] proposed an occlusion-aware ROI Pooling layer which integrated the prior structure information of pedestrian with visibility prediction into the final feature representations. The original region was divided into five parts and for each part, a sub-network enhanced the original region feature via a learned visibility score for better representations. Zhou et al. [222] proposed Bi-box which simultaneously estimated pedestrian detection as well as visible parts by regressing two bounding boxes, one for the full body and the other for visible part. In addition, a new positive-instance sampling criterion was proposed to bias positive training examples with large visible area, which showed effectiveness in training occlusion-aware detectors.

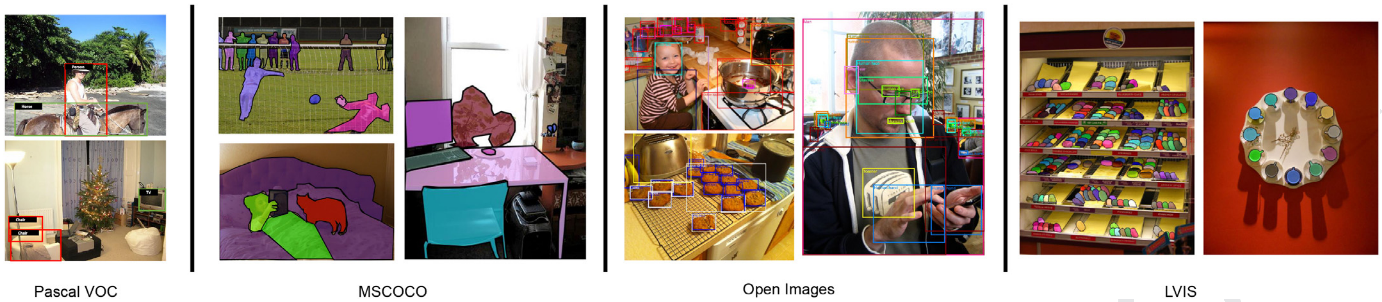


Fig. 11. Some examples of Pascal VOC, MSCOCO, Open Images and LVIS.

5.3. Others

There are some other real applications with object detection techniques, such as logo detection and video object detection.

Logo detection is an important research topic in e-commerce systems. Compared to generic detection, logo instance is much smaller with strong non-rigid transformation. Further, there are few logo detection baselines available. To address this issue, Su et al. [15] adopted the learning principle of webly data learning which automatically mined information from noisy web images and learns models with limited annotated data. Su et al. [14] described an image synthesising method to successfully learn a detector with limited logo instances. Hoi et al. [13] collected a large scale logo dataset from an e-commerce website and conducted a comprehensive analysis on the problem logo detection.

Existing detection algorithms are mainly designed for still images and are suboptimal for directly applying in videos for object detection. To detect objects in videos, there are two major differences from generic detection: temporal and contextual information. The location and appearance of objects in video should be temporally consistent between adjacent frames. Moreover, a video consists of hundreds of frames and thus contains far richer contextual information compared to a single still image. Han et al. [54] proposed a Seq-NMS which associates detection results of still images into sequences. Boxes of the same sequence are re-scored to the average score across frames, and other boxes along the sequence are suppressed by NMS. Kang et al. proposed Tubelets with Convolutional Neural Networks (T-CNN) [53] which was extended from Faster RCNN and incorporated the temporal and contextual information from tubelets (box sequence over time). T-CNN propagated the detection results to the adjacent frames by optical flow, and generated tubelets by applying tracking algorithms from high-confidence bounding boxes. The boxes along the tubelets were re-scored based on tubelets classification.

There are also many other real-world applications based on object detection such as vehicle detection [227–229], traffic-sign detection [230,231] and skeleton detection [232,233].

6. Detection benchmarks

In this section we will show some common benchmarks of generic object detection, face detection and pedestrian detection. We will first present some widely used datasets for each task and then introduce the evaluation metrics.

6.1. Generic detection benchmarks

Pascal VOC2007 [29] is a mid scale dataset for object detection with 20 categories. There are three image splits in VOC2007: training, validation and test with 2501, 2510 and 5011 images respectively.

Pascal VOC2012 [29] is a mid scale dataset for object detection which shares the same 20 categories with Pascal VOC2007. There are three image splits in VOC2012: training, validation and test with 5717, 5823 and 10991 images respectively. The annotation information of VOC2012 test set is not available.

MSCOCO [86] is a large scale dataset for with 80 categories. There are three image splits in MSCOCO: training, validation and test with 118287, 5000 and 40,670 images respectively. The annotation information of MSCOCO test set is not available.

Open Images [234] contains 1.9M images with 15M objects of 600 categories. The 500 most frequent categories are used to evaluate detection benchmarks, and more than 70% of these categories have over 1000 training samples.

LVIS [235] is a new collected benchmark with 164,000 images and 1000+ categories. It is a new dataset without any existing results so we leave the details of LVIS in future work section (Section 9).

ImageNet [37] is also a important dataset with 200 categories. However, the scale of ImageNet is huge and the object scale range is similar to VOC datasets, so it is not a commonly used benchmarks for detection algorithms.

Evaluation metrics: The details of evaluation metrics are shown in Tab. 1, both detection accuracy and inference speed are used to evaluate detection algorithms. For detection accuracy, mean Average Precision (mAP) is used as evaluation metric for all these challenges. The mAP is the mean value of AP, which is calculated separately for each class based on recall and precision. Assume the detector returns a set of predictions, we sample top γ predictions by confidence in decreasing order, which is denoted as D_γ. Next we calculate the number of true positive (TP_γ) and false positive (FP_γ) from sampled D_γ by the metric introduced in Section 2. Based on TP_γ and FP_γ, recall (R_γ) and precision (P_γ) are easy to obtain. AP is the region area under the curve of precision and recall, which is also easy to compute by varying the value of parameter γ. Finally mAP is computed by averaging the value of AP across all classes. For VOC2012, VOC2007 and ImageNet, IoU threshold of mAP is set to 0.5, and for MSCOCO, more comprehensive evaluation metrics are applied. There are six evaluation scores which demonstrates different capability of detection algorithms, including performance on different IoU thresholds and on different scale objects. Some examples of listed datasets (Pascal VOC, MSCOCO, Open Images and LVIS) are shown in Fig. 11.

6.2. Face detection benchmarks

In this section, we introduce several widely used face detection datasets (WIDER FACE, FDDB and Pascal Face) and the commonly used evaluation metrics.

WIDER FACE [200]. WIDER FACE has totally 32,203 images with about 400 k faces for a large range of scales. It has three subsets: 40% for training, 10% for validation, and 50% for test. The annotations of training and validation sets are online available. According

Please cite this article as: X. Wu, D. Sahoo and S.C.H. Hoi, Recent advances in deep learning for object detection, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.01.085

Table 1
Summary of common evaluation metrics for various detection tasks including generic object detection, face detection and pedestrian detection.

Alias	Meaning	Definition and description	
FPS	Frame per second	The number of images processed per second.	
Ω	IoU threshold	The IoU threshold to evaluate localization.	
D_γ	All Predictions	Top γ predictions returned by the detectors by confidence in decreasing order.	
TP_γ	True Positive	Correct predictions from sampled predictions D_γ.	
FP_γ	False Positive	False predictions from sampled predictions D_γ.	
P_γ	Precision	The fraction of TP_γ out of D_γ.	
R_γ	Recall	The fraction of TP_γ out of all positive samples.	
AP	Average Precision	Region area under curve of R_γ and P_γ by varying the value of parameter γ.	
mAP	mean AP	Average score of AP across all classes.	
TPR	True Positive Rate	The fraction of positive rate over false positives.	
FPPI	FP Per Image	The fraction of false positive for each image.	
MR	log-average missing rate	Average miss rate over different FPPI rates evenly spaced in log-space	
Generic Object Detection			
mAP	mean Average Precision	VOC2007	mAP at 0.50 IoU threshold over all 20 classes.
		VOC2012	mAP at 0.50 IoU threshold over all 20 classes.
		OpenImages	mAP at 0.50 IoU threshold over 500 most frequent classes.
		MSCOCO	• AP_{coco}: mAP averaged over ten Ω: {0.5: 0.05: 0.95};
			• AP_{50}: mAP at 0.50 IoU threshold;
			• AP_{75}: mAP at 0.75 IoU threshold;
			• AP_S: AP_{coco} for small objects of area smaller than 32^2;
			• AP_M: AP_{coco} for objects of area between 32^2 and 96^2;
			• AP_L: AP_{coco} for large objects of area bigger than 96^2;
Face detection			
mAP	mean Average Precision	Pascal Face	mAP at 0.50 IoU threshold.
		AFW	mAP at 0.50 IoU threshold.
		WIDER FACE	• mAP_{easy}: mAP for easy level faces;
			• mAP_{mid}: mAP for mid level faces;
			• mAP_{hard}: mAP for hard level faces;
TPR	True Positive Rate	FDDB	• TPR_{dis} with 1k FP at 0.50 IoU threshold, with bbox level.
			• TPR_{cont} with 1k FP at 0.50 IoU threshold, with eclipse level.
Pedestrian Detection			
mAP	mean Average Precision	KITTI	• mAP_{easy}: mAP for easy level pedestrians;
			• mAP_{mid}: mAP for mid level pedestrians;
			• mAP_{hard}: mAP for hard level pedestrians;
MR	log-average miss rate	CityPersons	MR: ranging from $1e^{-2}$ to 100 FPPI
		Caltech	MR: ranging from $1e^{-2}$ to $1e^0$ FPPI
		ETH	MR: ranging from $1e^{-2}$ to $1e^0$ FPPI
		INRIA	MR: ranging from $1e^{-2}$ to $1e^0$ FPPI

to the difficulty of detection tasks, it has three splits: Easy, Medium and Hard.

FDDB [199]. The Face Detection Data set and Benchmark (FDDB) is a well-known benchmark with 5171 faces in 2845 images. Commonly face detectors will first be trained on a large scale dataset (WIDERFACE etc.) and tested on FDDB.

PASCAL FACE [29]. This dataset was collected from PASCAL person layout test set, with 1335 labeled faces in 851 images. Similar to FDDB, it's commonly used as test set only.

Evaluation Metrics. As Table 1 shown, the evaluation metric for WIDER FACE and PASCAL FACE is mean average precision (mAP) with IoU threshold as 0.5, and for WIDER FACE the results of each difficulty level will be reported. For FDDB, true positive rate (TPR) at 1k false positives are used for evaluation. There are two annotation types to evaluate FDDB dataset: bounding box level and eclipse level.

6.3. Pedestrian detection benchmarks

In this section we will first introduce five widely used datasets (Caltech, ETH, INRIA, CityPersons and KITTI) for pedestrian object detection and then introduce their evaluation metrics.

CityPersons [257] is a new and challenging pedestrian detection dataset on top of the semantic segmentation dataset CityScapes [258], of which 5000 images are captured in several cities in Germany. A total of 35,000 persons with an additional 13,000 ignored regions, both bounding box annotation of all persons and annotation of visible parts are provided.

Caltech [259] is a popular and challenging datasets for pedestrian detection, which comes from approximately 10 h 30 Hz VGA video recorded by a car traversing the streets in the greater Los Angeles metropolitan area. The training and testing sets contains 42,782 and 4024 frames, respectively.

ETH [260] contains 1804 frames in three video clips and commonly it's used as test set to evaluate performance of the models trained on the large scale datasets (CityPersons dataset etc.).

INRIA [19] contains images of high resolution pedestrians collected mostly from holiday photos, which consists of 2120 images, including 1832 images for training and 288 images. Specifically, there are 614 positive images and 1218 negative images in the training set.

KITTI [261] contains 7481 labeled images of resolution 1250×375 and another 7518 images for testing. The person class in KITTI is divided into two subclasses: pedestrian and cyclist, both evaluated by mAP method. KITTI contains three evaluation metrics: easy, moderate and hard, with difference in the min. bounding box height, max. occlusion level, etc.

Evaluation Metrics. For CityPersons, INRIA and ETH, the log-average miss rate (MR) over 9 points ranging from $1e^{-2}$ to $1e^0$ FPPI (False Positive Per Image) is used to evaluate the performance of the detectors (lower is better). For KITTI, standard mean average precision is used as evaluation metric with 0.5 IoU threshold.

Please cite this article as: X. Wu, D. Sahoo and S.C.H. Hoi, Recent advances in deep learning for object detection, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.01.085

Table 2
Detection results on PASCAL VOC dataset. For VOC2007, the models are trained on VOC2007 and VOC2012 `trainval` sets and tested on VOC2007 `test` set. For VOC2012, the models are trained on VOC2007 and VOC2012 `trainval` sets plus VOC2007 `test` set and tested on VOC2012 `test` set by default. Since Pascal VOC datasets are well tuned and thus the number of detection frameworks for VOC reduces in recent years.

Method	Backbone	Proposed Year	Input size(Test)	mAP (%)	
				VOC2007	VOC2012
Two-stage Detectors:					
R-CNN [2]	VGG-16	2014	Arbitrary	66.0[a]	62.4[b]
SPP-net [2]	VGG-16	2014	$\sim 600 \times 1000$	63.1[a]	–
Fast R-CNN [38]	VGG-16	2015	$\sim 600 \times 1000$	70.0	68.4
Faster R-CNN [34]	VGG-16	2015	$\sim 600 \times 1000$	73.2	70.4
MR-CNN [131]	VGG-16	2015	Multi-Scale	78.2	73.9
Faster R-CNN [1]	ResNet-101	2016	$\sim 600 \times 1000$	76.4	73.8
R-FCN [52]	ResNet-101	2016	$\sim 600 \times 1000$	80.5	77.6
OHEM [148]	VGG-16	2016	$\sim 600 \times 1000$	74.6	71.9
HyperNet [50]	VGG-16	2016	$\sim 600 \times 1000$	76.3	71.4
ION [51]	VGG-16	2016	$\sim 600 \times 1000$	79.2	76.4
CRAFT [153]	VGG-16	2016	$\sim 600 \times 1000$	75.7	71.3[b]
LocNet [149]	VGG-16	2016	$\sim 600 \times 1000$	78.4	74.8[b]
R-FCN w DCN [97]	ResNet-101	2017	$\sim 600 \times 1000$	82.6	–
CoupleNet [125]	ResNet-101	2017	$\sim 600 \times 1000$	82.7	80.4
DeNet512(wide) [94]	ResNet-101	2017	$\sim 512 \times 512$	77.1	73.9
FPN-Reconfig [115]	ResNet-101	2018	$\sim 600 \times 1000$	82.4	81.1
DeepRegionLet [140]	ResNet-101	2018	$\sim 600 \times 1000$	83.3	**81.3**
DCN+R-CNN [132]	ResNet-101+ResNet-152	2018	Arbitrary	**84.0**	81.2
One-stage Detectors:					
YOLOv1 [40]	VGG16	2016	448×448	66.4	57.9
SSD512 [42]	VGG-16	2016	512×512	79.8	78.5
YOLOv2 [41]	Darknet	2017	544×544	78.6	73.5
DSSD513 [112]	ResNet-101	2017	513×513	81.5	80.0
DSOD300 [107]	DS/64-192-48-1	2017	300×300	77.7	76.3
RON384 [120]	VGG-16	2017	384×384	75.4	73.0
STDN513 [111]	DenseNet-169	2018	513×513	80.9	–
RefineDet512 [92]	VGG-16	2018	512×512	81.8	80.1
RFBNet512 [108]	VGG16	2018	512×512	**82.2**	–
CenterNet [64]	ResNet101	2019	512×512	78.7	-
CenterNet [64]	DLA [64]	2019	512×512	80.7	-

[a] This entry reports the model is trained with VOC2007 `trainval` sets only.
[b] This entry reports the model are trained with VOC2012 `trainval` sets only.

7. State-of-the-art for object detection

Generic object detection: Pascal VOC2007, VOC2007 and MSCOCO are three most commonly used datasets for evaluating detection algorithms. Pascal VOC2012 and VOC2007 are mid scale datasets with 2 or 3 objects per image and the range of object size in VOC dataset is not large. For MSCOCO, there are nearly 10 objects per image and the majority objects are small objects with large scale ranges, which leads to a very challenge task for detection algorithms. In Tables 2 and 3 we give the benchmarks of VOC2007, VOC2012 and MSCOCO over the recent few years.

Face detection: WIDER FACE is currently the most commonly used benchmark for evaluating face detection algorithms. High variance of face scales and large number of faces per image make WIDER FACE the hardest benchmark for face detection, with three evaluation metrics: easy, medium and hard. In Table 4 we give the benchmarks of WIDER FACE over the recent few years.

Pedestrian detection: CityPersons is a new but challenging benchmark for pedestrian detection. The dataset is split into different subsets according to the height and visibility level of the objects, and thus it's able to evaluate the detectors in a more comprehensive manner. The results are listed in Tab. 5, where MR is used for evaluation (lower is better).

8. Related surveys

There are some other surveys which is parallel to our work [265–269]. Sultana et al. [267] review the existing deep learning based detectors and their training settings. Agarwal et al. [268] review the connection between deep learning and detection algorithms proposed in recent years and explore the potential leads by introducing some relevant topics such as few-shot detection and life-long detection. Zhao et al. [269] review the existing deep learning based detectors and also provide the benchmarks of generic detection and real applications. Jiao et al. [266] cover a series of general detection algorithms and introduce the state-of-the-art methods to explore novel solutions and directions to develop the new detectors.

Compared with these surveys, our work not only reviews the existing representative detectors, but also makes comprehensive analysis on general components and learning strategy of different detectors. We aim to fully explore the factors which impact detection tasks, which are not covered in most existing surveys. Liu et al. [265] also give a comprehensive understanding of generic object detection as well as the analysis of detector components and learning strategies. However, their work only focus on generic detection but ignore the importance of detection in real-world applications. In our survey, we also give a comprehensive understanding of the limitations and strategies to adapt generic detection algorithms into real-world applications. Furthermore, we organize the state-of-the-art algorithms for both generic detection and real-world applications to facilitate the future research. Finally, based on the tendency of the latest work proposed within the past one year, we discuss the future direction of object detection.

9. Concluding remarks and future directions

Object detection has been actively investigated and new state-of-the-art results have been reported almost every few months.

Please cite this article as: X. Wu, D. Sahoo and S.C.H. Hoi, Recent advances in deep learning for object detection, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.01.085

Table 3
Detection performance on the MS COCO `test-dev` data set. "++" denotes applying inference strategy such as multi scale test, horizontal flip, etc.

Method	Backbone	Year	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-Stage Detectors:								
Fast R-CNN [38]	VGG-16	2015	19.7	35.9	–	–	–	–
Faster R-CNN [34]	VGG-16	2015	21.9	42.7	–	–	–	–
OHEM [148]	VGG-16	2016	22.6	42.5	22.2	5.0	23.7	37.9
ION [51]	VGG-16	2016	23.6	43.2	23.6	6.4	24.1	38.3
OHEM++ [148]	VGG-16	2016	25.5	45.9	26.1	7.4	27.7	40.3
R-FCN [52]	ResNet-101	2016	29.9	51.9	–	10.8	32.8	45.0
Faster R-CNN+++ [1]	ResNet-101	2016	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [39]	ResNet-101	2016	36.2	59.1	39.0	18.2	39.0	48.2
DeNet-101(wide) [94]	ResNet-101	2017	33.8	53.4	36.1	12.3	36.1	50.8
CoupleNet [125]	ResNet-101	2017	34.4	54.8	37.2	13.4	38.1	50.8
Faster R-CNN by G-RMI [167]	Inception-ResNet-v2	2017	34.7	55.5	36.7	13.5	38.1	52.0
Deformable R-FCN [52]	Aligned-Inception-ResNet	2017	37.5	58.0	40.8	19.4	40.1	52.5
Mask-RCNN [3]	ResNeXt-101	2017	39.8	62.3	43.4	22.1	43.2	51.2
umd_det [236]	ResNet-101	2017	40.8	62.4	44.9	23.0	43.4	53.2
Fitness-NMS [152]	ResNet-101	2017	41.8	60.9	44.9	21.5	45.0	57.5
DCN w Relation Net [138]	ResNet-101	2018	39.0	58.6	42.9	–	–	–
DeepRegionlets [140]	ResNet-101	2018	39.3	59.8	–	21.7	43.7	50.9
C-Mask RCNN [141]	ResNet-101	2018	42.0	62.9	46.4	23.4	44.7	53.8
Group Norm [237]	ResNet-101	2018	42.3	62.8	46.2	–	–	–
DCN+R-CNN [132]	ResNet-101+ResNet-152	2018	42.6	65.3	46.5	26.4	46.1	56.4
Cascade R-CNN [49]	ResNet-101	2018	42.8	62.1	46.3	23.7	45.5	55.2
SNIP++ [98]	DPN-98	2018	45.7	67.3	51.1	29.3	48.8	57.1
SNIPER++ [146]	ResNet-101	2018	46.1	67.0	51.6	29.6	48.9	58.1
PANet++ [238]	ResNeXt-101	2018	47.4	67.2	51.8	30.1	51.7	60.0
Grid R-CNN [151]	ResNeXt-101	2019	43.2	63.0	46.6	25.1	46.5	55.2
DCN-v2 [144]	ResNet-101	2019	44.8	66.3	48.8	24.4	48.1	59.6
DCN-v2++ [144]	ResNet-101	2019	46.0	67.9	50.8	27.8	49.1	59.5
TridentNet [239]	ResNet-101	2019	42.7	63.6	46.5	23.9	46.6	56.6
TridentNet [239]	ResNet-101-Deformable	2019	**48.4**	**69.7**	**53.5**	31.8	51.3	60.3
Single-Stage Detectors:								
SSD512 [42]	VGG-16	2016	28.8	48.5	30.3	10.9	31.8	43.5
RON384++ [120]	VGG-16	2017	27.4	49.5	27.1	–	–	–
YOLOv2 [41]	DarkNet-19	2017	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [112]	ResNet-101	2017	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [112]	ResNet-101	2017	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet800++ [43]	ResNet-101	2017	39.1	59.1	42.3	21.8	42.7	50.2
STDN513 [111]	DenseNet-169	2018	31.8	51.0	33.6	14.4	36.1	43.4
FPN-Reconfig [115]	ResNet-101	2018	34.6	54.3	37.3	–	–	–
RefineDet512 [92]	ResNet-101	2018	36.4	57.5	39.5	16.6	39.9	51.4
RefineDet512++ [92]	ResNet-101	2018	41.8	62.9	45.7	25.6	45.1	54.1
GHM SSD [147]	ResNeXt-101	2018	41.6	62.8	44.2	22.3	45.1	55.3
CornerNet511 [63]	Hourglass-104	2018	40.5	56.5	43.1	19.4	42.7	53.9
CornerNet511++ [63]	Hourglass-104	2018	42.1	57.8	45.3	20.8	44.8	56.7
M2Det800 [116]	VGG-16	2019	41.0	59.7	45.0	22.1	46.5	53.8
M2Det800++ [116]	VGG-16	2019	44.2	64.6	49.3	29.2	47.9	55.1
ExtremeNet [240]	Hourglass-104	2019	40.2	55.5	43.2	20.4	43.2	53.1
CenterNet-HG [64]	Hourglass-104	2019	42.1	61.1	45.9	24.1	45.5	52.8
FCOS [241]	ResNeXt-101	2019	42.1	62.1	45.2	25.6	44.9	52.0
FSAF [95]	ResNeXt-101	2019	42.9	63.8	46.3	26.6	46.2	52.7
CenterNet511 [65]	Hourglass-104	2019	44.9	62.4	48.1	25.6	47.4	57.4
CenterNet511++ [65]	Hourglass-104	2019	**47.0**	**64.5**	**50.7**	28.9	49.9	58.9

However, there are still many open challenges. Below we discuss several open challenges and future directions.

(i) Scalable proposal generation strategy. As claimed in Section 3.4, currently most detectors are anchor-based methods, and there are some critical shortcomings which limit the detection accuracy. Current anchor priors are mainly manually designed which is difficult to match multi-scale objects and the matching strategy based on IoU is also heuristic. Although some methods have been proposed to transform anchor-based methods into anchor-free methods (e.g. methods based on keypoints), there are still some limitations (high computation cost etc.) with large space to improve. From Fig. 2, developing anchor-free methods becomes a very hot topic in object detection [63,65,95,240,241], and thus designing an efficient and effective proposal generation strategy is potentially a very important research direction in the future.

(ii) Effective encoding of contextual information. Contexts can contribute or impede visual object detection results, as objects in the visual world have strong relationships, and contexts are critical to better understand the visual worlds. However, little effort has been focused on how to correctly use contextual information. How to incorporate contexts for object detection effectively can be a promising future direction.

(iii) Detection based on Auto Machine Learning (AutoML). To design an optimal backbone architecture for a certain task can significantly improve the results but also requires huge engineering effort. Thus to learn backbone architecture directly on the datasets is a very interesting and important research direction. From Fig. 2, inspired by the pioneering AutoML work on image classification [270,271], more relevant work has been proposed to address detection problems via AutoML [272,273], such as learning FPN structure [273] and learning data augmentation policies [274],

Table 4
Detection results on WIDER FACE dataset. The models are trained on WIDER FACE `training` sets and tested on WIDER FACE `test` set.

Method	Year	mAP (%)		
		Easy	Medium	Hard
ACF-WIDER[242]	2014	69.5	58.8	29.0
Faceness [243]	2015	71.6	60.4	31.5
Two-stage CNN [200]	2016	65.7	58.9	30.4
LDCF+ [244]	2016	79.7	77.2	56.4
CMS-CNN [195]	2016	90.2	87.4	64.3
MSCNN [106]	2016	91.7	90.3	80.9
ScaleFace [245]	2017	86.7	86.6	76.4
HR [99]	2017	92.3	91.0	81.9
SHH [189]	2017	92.7	91.5	84.4
Face R-CNN [191]	2017	93.2	91.6	82.7
S3FD [87]	2017	93.5	92.1	85.8
Face R-FCN [198]	2017	94.3	93.1	87.6
FAN [246]	2017	94.6	93.6	88.5
FANet [188]	2017	94.7	93.9	88.7
FDNet [247]	2018	95.0	93.9	87.8
PyramidBox [185]	2018	95.6	94.6	88.7
SRN [186]	2018	95.9	94.8	89.6
DSFD [187]	2018	96.0	95.3	90.0
DFS [248]	2018	96.3	95.4	90.7
SFDet [249]	2019	94.8	94.0	88.3
CSP [250]	2019	94.9	94.4	89.9
PyramidBox++ [251]	2019	95.6	95.2	90.9
VIM-FD [252]	2019	96.2	95.3	90.2
ISRN [253]	2019	96.3	95.4	90.3
RetinaFace [254]	2019	96.3	95.6	91.4
AlnnoFace [255]	2019	96.5	95.7	91.2
RefineFace [256]	2019	**96.6**	**95.8**	**91.4**

Table 5
Detection results on CityPersons dataset. The models are trained on CityPersons `training` sets and tested on CityPersons `test` set. There are four evaluation metrics: Reasonable (R.), Small (S.), Heavy (H.) and All (A.), which are related to the height and visibility level of the objects.

Method	Year	R.	S.	H.	A.
FRCNN [38]	2015	12.97	37.24	50.47	43.86
MS-CNN [106]	2016	13.32	15.86	51.88	39.94
RepLoss [213]	2017	11.48	15.67	52.59	39.17
Ada-FRCN [257]	2017	12.97	37.24	50.47	43.86
OR-CNN [214]	2018	11.32	14.19	51.43	40.19
HBAN [262]	2019	11.26	15.68	39.54	38.77
MGAN [263]	2019	9.29	11.38	40.97	38.86
APD [264]	2019	**8.27**	**11.03**	**35.45**	**35.65**

which show significant improvement over the baselines. However, the required computation resource for AutoML is unaffordable to most researchers (more than 100 GPU cards to train a single model). Thus, developing a low-computation framework shall have a large impact for object detection. Further, new structure policies (such as proposal generation and region encoding) of detection task can be explored in the future.

(iv) Emerging benchmarks for object detection. Currently MSCOCO is the most commonly used detection benchmark testbed. However, MSCOCO has only 80 categories, which is still too small to understand more complicated scenes in real world. Recently, a new benchmark dataset LVIS [235] has been proposed in order to collect richer categorical information. LVIS contains 164,000 images with 1000+ categories, and there are total of 2.2 million high-quality instance segmentation masks. Further, LVIS simulates the real-world low-shot scenario where a large number of categories are present but per-category data is sometimes scarce. LVIS will open a new benchmark for more challenging detection, segmentation and low-shot learning tasks in near future.

(v) Low-shot object detection. Training detectors with limited labeled data is dubbed as Low-shot detection. Deep learning based detectors often have huge amount of parameters and thus are data-hungry, which require large amount of labeled data to achieve satisfactory performance. However, labeling objects in images with bounding box level annotation is very time-consuming. Low-shot learning has been actively studied for classification tasks, but only a few studies are focused on detection tasks. For example, Multi-modal Self-Paced Learning for Detection (MSPLD) [275] addresses the low-shot detection problem in a semi-supervised learning setting where a large-scale unlabeled dataset is available. RepMet [276] adopts a Deep Metric Learning (DML) structure, which jointly learns feature embedding space and data distribution of training set categories. However, RepMet was only tested on datasets with similar concepts (animals). Low-Shot Transfer Detector (LSTD) [277] addresses low-shot detection based on transfer learning which transfers the knowledge form large annotated external datasets to the target set by knowledge regularization. LSTD still suffers from overfitting. There is still a large room to improve the low-shot detection tasks.

(vi) Backbone architecture for detection task. It has become a common practice to adopt weights of classification models pretrained on a large scale dataset for detection. However, there still exists conflicts between classification and detection tasks [78], and thus directly adopting a pretrained network may not result in the optimal solution. From Table 3, most state-of-the-art detection algorithms are based on classification backbones, and only a few of them try different selections (such as CornerNet based on Hourglass Net). Thus, developing a detection-aware backbone architecture is also an important research direction for the future.

(vii) Other research issues. In addition, there are some other open research issues, such as large batch learning [278] and incremental learning [279]. Batch size is a key factor in DCNN training but has not been well studied for detection. For incremental learning, detection algorithms still suffer from catastrophic forgetting if adapted to a new task without initial training data. These open and fundamental research issues also deserve more attention for future work.

In this survey, we give a comprehensive survey of recent advances in deep learning techniques for object detection tasks. The main contents of this survey are divided into three major categories: object detector components, machine learning strategies, real-world applications and benchmark evaluations. We have reviewed a large body of representative articles in recent literature, and presented the contributions on this important topic in a structured and systematic manner. We hope this survey can give readers a comprehensive understanding of object detection with deep learning and potentially spur more research work on object detection techniques and their applications.

Declaration of Competing Interest

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

CRediT authorship contribution statement

Xiongwei Wu: Conceptualization, Methodology, Software, Investigation, Writing - original draft, Writing - review & editing. **Doyen Sahoo:** Validation, Investigation, Writing - review & editing. **Steven C.H. Hoi:** Supervision.

References

[1] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the CVPR, 2016.

Please cite this article as: X. Wu, D. Sahoo and S.C.H. Hoi, Recent advances in deep learning for object detection, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.01.085

[2] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the CVPR, 2014.
[3] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: Proceedings of the ICCV, 2017.
[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, Semantic image segmentation with deep convolutional nets and fully connected CRFS, 2014. arXiv: 1412.7062.
[5] Y. Sun, D. Liang, X. Wang, X. Tang, Deepid3: face recognition with very deep neural networks, 2015. arXiv: 1502.00873.
[6] Y. Sun, Y. Chen, X. Wang, X. Tang, Deep learning face representation by joint identification-verification, in: Proceedings of the NeurIPS, 2014.
[7] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, L. Song, Sphereface: deep hypersphere embedding for face recognition, in: Proceedings of the CVPR, 2017.
[8] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, S. Yan, Scale-aware fast R-CNN for pedestrian detection, in: Proceedings of the IEEE Transactions on Multimedia, 2018.
[9] J. Hosang, M. Omran, R. Benenson, B. Schiele, Taking a deeper look at pedestrians, in: Proceedings of the CVPR, 2015.
[10] A. Angelova, A. Krizhevsky, V. Vanhoucke, A.S. Ogale, D. Ferguson, Real-time pedestrian detection with deep network cascades., in: Proceedings of the BMVC, 2015.
[11] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: Proceedings of the CVPR, 2014.
[12] H. Mobahi, R. Collobert, J. Weston, Deep learning from temporal coherence in video, in: Proceedings of the Annual International Conference on Machine Learning, 2009.
[13] S.C. Hoi, X. Wu, H. Liu, Y. Wu, H. Wang, H. Xue, Q. Wu, Logo-net: large-scale deep logo detection and brand recognition with deep region-based convolutional networks, 2015. arXiv: 1511.02462.
[14] H. Su, X. Zhu, S. Gong, Deep learning logo detection with data expansion by synthesising context, in: Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), 2017.
[15] H. Su, S. Gong, X. Zhu, Scalable deep learning logo detection, 2018. arXiv: 1803.11417.
[16] A. Vedaldi, V. Gulshan, M. Varma, A. Zisserman, Multiple kernels for object detection, in: Proceedings of the ICCV, 2009.
[17] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Proceedings of the CVPR, 2001.
[18] H. Harzallah, F. Jurie, C. Schmid, Combining efficient object localization and image classification, in: Proceedings of the ICCV, 2009.
[19] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of the CVPR, 2005.
[20] P. Viola, M.J. Jones, Robust real-time face detection, in: Proceedings of the IJCV, 2004.
[21] D.G. Lowe, Object recognition from local scale-invariant features, in: Proceedings of the ICCV, 1999.
[22] R. Lienhart, J. Maydt, An extended set of haar-like features for rapid object detection, in: Proceedings of the International Conference on Image Processing, 2002.
[23] H. Bay, T. Tuytelaars, L. Van Gool, Surf: speeded up robust features, in: Proceedings of the ECCV, 2006.
[24] M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, B. Scholkopf, Support vector machines, in: Proceedings of the IEEE Intelligent Systems and their applications, 1998.
[25] D. Opitz, R. Maclin, Popular ensemble methods: an empirical study, in: Proceedings of the Artificial Intelligence Research, 1999.
[26] Y. Freund, R.E. Schapire, et al., Experiments with a new boosting algorithm, in: Proceedings of the ICML, 1996.
[27] Y. Yu, J. Zhang, Y. Huang, S. Zheng, W. Ren, C. Wang, K. Huang, T. Tan, Object detection by context and boosted hog-LBP, in: Proceedings of the PASCAL VOC Challenge, 2010.
[28] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Discriminatively trained mixtures of deformable part models, in: Proceedings of the PASCAL VOC Challenge, 2008.
[29] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, in: Proceedings of the IJCV, 2010.
[30] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, in: Proceedings of the TPAMI, 2010.
[31] D.G. Lowe, Distinctive image features from scale-invariant keypoints, in: Proceedings of the IJCV, 2004.
[32] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, in: Proceedings of the TPAMI, 2002.
[33] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the NeurIPS, 2012.
[34] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, in: Proceedings of the NeurIPS, 2015.
[35] K. Fukushima, S. Miyake, Neocognitron: a self-organizing neural network model for a mechanism of visual pattern recognition, in: Proceedings of the Competition and Cooperation in Neural Nets, 1982.
[36] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, in: Proceedings of the IEEE, 1998.
[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, in: Proceedings of the CVPR, 2009.
[38] R. Girshick, Fast R-CNN, in: Proceedings of the ICCV, 2015.
[39] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the CVPR, 2017.
[40] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in: Proceedings of the CVPR, 2016.
[41] J. Redmon, A. Farhadi, Yolo9000: better, faster, stronger, in: Proceedings of the CVPR, 2017.
[42] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: Single shot multibox detector, in: Proceedings of the ECCV, 2016.
[43] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the ICCV, 2017.
[44] S. Fidler, R. Mottaghi, A. Yuille, R. Urtasun, Bottom-up segmentation for top-down detection, in: Proceedings of the CVPR, 2013.
[45] J.R. Uijlings, K.E. Van De Sande, T. Gevers, A.W. Smeulders, Selective search for object recognition, in: Proceedings of the IJCV, 2013.
[46] J. Kleban, X. Xie, W.-Y. Ma, Spatial pyramid mining for logo detection in natural scenes, in: Proceedings of the 2008 IEEE International Conference on Multimedia and Expo, 2008.
[47] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, in: Proceedings of the ECCV, 2014.
[48] C.L. Zitnick, P. Dollár, Edge boxes: locating object proposals from edges, in: Proceedings of the ECCV, 2014.
[49] Z. Cai, N. Vasconcelos, Cascade R-CNN: delving into high quality object detection, in: Proceedings of the CVPR, 2018.
[50] T. Kong, A. Yao, Y. Chen, F. Sun, Hypernet: towards accurate region proposal generation and joint object detection, in: Proceedings of the CVPR, 2016.
[51] S. Bell, C. Lawrence Zitnick, K. Bala, R. Girshick, Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks, in: Proceedings of the CVPR, 2016.
[52] J. Dai, Y. Li, K. He, J. Sun, R-FCN: object detection via region-based fully convolutional networks, in: Proceedings of the NeurIPS, 2016.
[53] K. Kang, W. Ouyang, H. Li, X. Wang, Object detection from video tubelets with convolutional neural networks, in: Proceedings of the CVPR, 2016.
[54] W. Han, P. Khorrami, T.L. Paine, P. Ramachandran, M. Babaeizadeh, H. Shi, J. Li, S. Yan, T.S. Huang, SEQ-NMS for video object detection, 2016. arXiv: 1602.08465.
[55] M. Rayat Imtiaz Hossain, J. Little, Exploiting temporal information for 3D human pose estimation, in: Proceedings of the ECCV, 2018.
[56] G. Pavlakos, X. Zhou, K.G. Derpanis, K. Daniilidis, Coarse-to-fine volumetric prediction for single-image 3D human pose, in: Proceedings of the CVPR, 2017.
[57] P.O. Pinheiro, T.-Y. Lin, R. Collobert, P. Dollár, Learning to refine object segments, in: Proceedings of the ECCV, 2016.
[58] P.O. Pinheiro, R. Collobert, P. Dollár, Learning to segment object candidates, in: Proceedings of the NeurIPS, 2015.
[59] J. Dai, K. He, J. Sun, Instance-aware semantic segmentation via multi-task network cascades, in: Proceedings of the CVPR, 2016.
[60] Z. Huang, L. Huang, Y. Gong, C. Huang, X. Wang, Mask scoring R-CNN, in: Proceedings of the CVPR, 2019.
[61] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun, Overfeat: integrated recognition, localization and detection using convolutional networks, 2013. arXiv: 1312.6229.
[62] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proceedings of the ICML, 2015.
[63] H. Law, J. Deng, Cornernet: detecting objects as paired keypoints, in: Proceedings of the ECCV, 2018.
[64] X. Zhou, D. Wang, P. Krähenbühl, Objects as points, 2019. arXiv: 1904.07850.
[65] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, Q. Tian, Centernet: keypoint triplets for object detection, 2019. arXiv: 1904.08189.
[66] H. Robbins, S. Monro, A stochastic approximation method, The Annals of Mathematical Statistics, 1951.
[67] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, 2014. arXiv: 1412.6980.
[68] V. Nair, G.E. Hinton, Rectified linear units improve restricted Boltzmann machines, in: Proceedings of the ICML, 2010.
[69] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014. arXiv: 1409.1556.
[70] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: Proceedings of the ECCV, Springer, 2016.
[71] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks., in: Proceedings of the CVPR, 2017.
[72] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, J. Feng, Dual path networks, in: Proceedings of the NeurIPS, 2017, pp. 4467–4475.
[73] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: Proceedings of the CVPR, 2017.
[74] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: efficient convolutional neural networks for mobile vision applications, 2017. arXiv: 1704.04861.
[75] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the CVPR, 2015.
[76] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the CVPR, 2016.
[77] C. Szegedy, S. Ioffe, V. Vanhoucke, A.A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning., in: Proceedings of the AAAI, 2017.

Please cite this article as: X. Wu, D. Sahoo and S.C.H. Hoi, Recent advances in deep learning for object detection, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.01.085

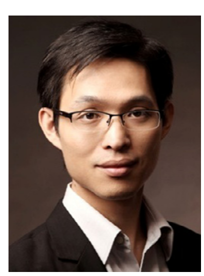
[78] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, J. Sun, Detnet: a backbone network for object detection, in: Proceedings of the ECCV, 2018.
[79] A. Newell, K. Yang, J. Deng, Stacked hourglass networks for human pose estimation, in: Proceedings of the ECCV, 2016.
[80] B. Alexe, T. Deselaers, V. Ferrari, Measuring the objectness of image windows, in: Proceedings of the TPAMI, 2012.
[81] E. Rahtu, J. Kannala, M. Blaschko, Learning a category independent object detection cascade, in: Proceedings of the ICCV, 2011.
[82] P.F. Felzenszwalb, D.P. Huttenlocher, Efficient graph-based image segmentation, in: Proceedings of the IJCV, 2004.
[83] S. Manen, M. Guillaumin, L. Van Gool, Prime object proposals with randomized prim's algorithm, in: Proceedings of the CVPR, 2013.
[84] J. Carreira, C. Sminchisescu, CPMC: automatic object segmentation using constrained parametric min-cuts, in: Proceedings of the TPAMI, 2011.
[85] I. Endres, D. Hoiem, Category-independent object proposals with diverse ranking, in: Proceedings of the TPAMI, 2014.
[86] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: common objects in context, in: Proceedings of the ECCV, 2014.
[87] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, S.Z. Li, S3FD: single shot scale-invariant face detector, in: Proceedings of the ICCV, 2017.
[88] W. Luo, Y. Li, R. Urtasun, R. Zemel, Understanding the effective receptive field in deep convolutional neural networks, in: Proceedings of the NeurIPS, 2016.
[89] C. Zhu, R. Tao, K. Luu, M. Savvides, Seeing small faces from robust anchor perspective, in: Proceedings of the CVPR, 2018.
[90] L.J.Z.X. Lele Xie, Y. Liu, DERPN: taking a further step toward more general object detection, in: Proceedings of the AAAI, 2019.
[91] A. Ghodrati, A. Diba, M. Pedersoli, T. Tuytelaars, L. Van Gool, Deepproposal: hunting objects by cascading deep convolutional layers, in: Proceedings of the ICCV, 2015.
[92] S. Zhang, L. Wen, X. Bian, Z. Lei, S.Z. Li, Single-shot refinement neural network for object detection, in: Proceedings of the CVPR, 2018.
[93] T. Yang, X. Zhang, Z. Li, W. Zhang, J. Sun, Metaanchor: learning to detect objects with customized anchors, in: Proceedings of the NeurIPS, 2018.
[94] L. Tychsen-Smith, L. Petersson, Denet: scalable real-time object detection with directed sparse sampling, in: Proceedings of the ICCV, 2017.
[95] C. Zhu, Y. He, M. Savvides, Feature selective anchor-free module for single-shot object detection, in: Proceedings of the CVPR, 2019.
[96] Y. Lu, T. Javidi, S. Lazebnik, Adaptive object detection using adjacency and zoom prediction, in: Proceedings of the CVPR, 2016.
[97] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, in: Proceedings of the ICCV, 2017.
[98] B. Singh, L.S. Davis, An analysis of scale invariance in object detection–snip, in: Proceedings of the CVPR, 2018.
[99] P. Hu, D. Ramanan, Finding tiny faces, in: Proceedings of the CVPR, 2017.
[100] F. Yang, W. Choi, Y. Lin, Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers, in: Proceedings of the CVPR, 2016.
[101] Y. Liu, H. Li, J. Yan, F. Wei, X. Wang, X. Tang, Recurrent scale approximation for object detection in CNN, in: Proceedings of the ICCV, 2017.
[102] A. Shrivastava, R. Sukthankar, J. Malik, A. Gupta, Beyond skip connections: top-down modulation for object detection, 2016. arXiv: 1612.06851.
[103] H. Wang, Q. Wang, M. Gao, P. Li, W. Zuo, Multi-scale location-aware kernel representation for object detection, in: Proceedings of the CVPR, 2018.
[104] K.-H. Kim, S. Hong, B. Roh, Y. Cheon, M. Park, PVANET: deep but lightweight neural networks for real-time object detection, 2016. arXiv: 1608.08021.
[105] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Proceedings of the International Conference on Medical image Computing and Computer-Assisted Intervention, 2015.
[106] Z. Cai, Q. Fan, R.S. Feris, N. Vasconcelos, A unified multi-scale deep convolutional neural network for fast object detection, in: Proceedings of the ECCV, 2016.
[107] Z. Shen, Z. Liu, J. Li, Y.-G. Jiang, Y. Chen, X. Xue, Dsod: learning deeply supervised object detectors from scratch, in: Proceedings of the ICCV, 2017.
[108] S. Liu, D. Huang, Y. Wang, Receptive field block net for accurate and fast object detection, in: Proceedings of the ECCV, 2018.
[109] J. Ren, X. Chen, J. Liu, W. Sun, J. Pang, Q. Yan, Y.-W. Tai, L. Xu, Accurate single stage detector using recurrent rolling convolution, in: Proceedings of the CVPR, 2017.
[110] J. Jeong, H. Park, N. Kwak, Enhancement of SSD by concatenating feature maps for object detection, 2017. arXiv: 1705.09587.
[111] P. Zhou, B. Ni, C. Geng, J. Hu, Y. Xu, Scale-transferrable object detection, in: Proceedings of the CVPR, 2018.
[112] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, A.C. Berg, DSSD: deconvolutional single shot detector, 2017. arXiv: 1701.06659.
[113] S. Woo, S. Hwang, I.S. Kweon, Stairnet: Top-down semantic aggregation for accurate one shot detection, in: Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 2018.
[114] H. Li, Y. Liu, W. Ouyang, X. Wang, Zoom out-and-in network with recursive training for object proposal, 2017. arXiv: 1702.05711.
[115] T. Kong, F. Sun, W. Huang, H. Liu, Deep feature pyramid reconfiguration for object detection, in: Proceedings of the ECCV, 2018.
[116] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, H. Ling, M2DET: a single-shot object detector based on multi-level feature pyramid network, in: Proceedings of the AAAI, 2019.
[117] Z. Li, F. Zhou, FSSD: feature fusion single shot multibox detector, 2017. arXiv: 1712.00960.
[118] K. Lee, J. Choi, J. Jeong, N. Kwak, Residual features and unified prediction network for single stage detection, 2017. arXiv: 1707.05031.
[119] L. Cui, MDSSD: multi-scale deconvolutional single shot detector for small objects, 2018. arXiv: 1805.07009.
[120] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, Y. Chen, RON: reverse connection with objectness prior networks for object detection, in: Proceedings of the CVPR, 2017.
[121] B. Lim, S. Son, H. Kim, S. Nah, K. Mu Lee, Enhanced deep residual networks for single image super-resolution, in: Proceedings of the CVPR Workshops, 2017.
[122] W. Shi, J. Caballero, F. Huszár, J. Totz, A.P. Aitken, R. Bishop, D. Rueckert, Z. Wang, Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, in: Proceedings of the CVPR, 2016.
[123] B. Jiang, R. Luo, J. Mao, T. Xiao, Y. Jiang, Acquisition of localization confidence for accurate object detection, in: Proceedings of the ECCV, 2018.
[124] Y. Zhai, J. Fu, Y. Lu, H. Li, Feature selective networks for object detection, in: Proceedings of the CVPR, 2018.
[125] Y. Zhu, C. Zhao, J. Wang, X. Zhao, Y. Wu, H. Lu, Couplenet: coupling global structure with local parts for object detection, in: Proceedings of the ICCV, 2017.
[126] C. Galleguillos, S. Belongie, Context based object categorization:a critical survey, in: Proceedings of the Computer Vision and Image Understanding, 2010.
[127] W. Ouyang, X. Wang, X. Zeng, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, C.-C. Loy, et al., Deepid-net: deformable deep convolutional neural networks for object detection, in: Proceedings of the CVPR, 2015.
[128] W. Chu, D. Cai, Deep feature based contextual model for object detection, in: Proceedings of the Neurocomputing, 2018.
[129] Y. Zhu, R. Urtasun, R. Salakhutdinov, S. Fidler, SEGDEEPM: exploiting segmentation and context in deep neural networks for object detection, in: Proceedings of the CVPR, 2015.
[130] X. Chen, A. Gupta, Spatial memory for context reasoning in object detection, in: Proceedings of the ICCV, 2017.
[131] S. Gidaris, N. Komodakis, Object detection via a multi-region and semantic segmentation-aware CNN model, in: Proceedings of the ICCV, 2015.
[132] B. Cheng, Y. Wei, H. Shi, R. Feris, J. Xiong, T. Huang, Revisiting RCNN: on awakening the classification power of faster RCNN, in: Proceedings of the ECCV, 2018.
[133] X. Zhao, S. Liang, Y. Wei, Pseudo mask augmented object detection, in: Proceedings of the CVPR, 2018.
[134] Z. Zhang, S. Qiao, C. Xie, W. Shen, B. Wang, A.L. Yuille, Single-Shot Object Detection with Enriched Semantics, Technical Report, 2018. Q4
[135] A. Shrivastava, A. Gupta, Contextual priming and feedback for faster R-CNN, in: Proceedings of the ECCV, 2016.
[136] B. Li, T. Wu, L. Zhang, R. Chu, Auto-context R-CNN, 2018. arXiv: 1807.02842.
[137] Y. Liu, R. Wang, S. Shan, X. Chen, Structure inference net: object detection using scene-level context and instance-level relationships, in: Proceedings of the CVPR, 2018.
[138] H. Hu, J. Gu, Z. Zhang, J. Dai, Y. Wei, Relation networks for object detection, in: Proceedings of the CVPR, 2018.
[139] J. Gu, H. Hu, L. Wang, Y. Wei, J. Dai, Learning region features for object detection, in: Proceedings of the ECCV, 2018.
[140] H. Xu, X. Lv, X. Wang, Z. Ren, R. Chellappa, Deep regionlets for object detection, in: Proceedings of the ECCV, 2018.
[141] Z. Chen, S. Huang, D. Tao, Context refinement for object detection, in: Proceedings of the ECCV, 2018.
[142] X. Zeng, W. Ouyang, B. Yang, J. Yan, X. Wang, Gated bi-directional CNN for object detection, in: Proceedings of the ECCV, 2016.
[143] J. Li, Y. Wei, X. Liang, J. Dong, T. Xu, J. Feng, S. Yan, Attentive contexts for object detection, in: Proceedings of the IEEE Transactions on Multimedia, 2017.
[144] X. Zhu, S. Lin, H. Hu, J. Dai, Deformable convnets v2: more deformable, better results, in: Proceedings of the CVPR, 2019.
[145] R. Girshick, F. Iandola, T. Darrell, J. Malik, Deformable part models are convolutional neural networks, in: Proceedings of the CVPR, 2015.
[146] B. Singh, M. Najibi, L.S. Davis, Sniper: Efficient multi-scale training, in: Proceedings of the NeurIPS, 2018.
[147] Y.L. Buyu Li, X. Wang, Gradient harmonized single-stage detector, in: Proceedings of the AAAI, 2019.
[148] A. Shrivastava, A. Gupta, R. Girshick, Training region-based object detectors with online hard example mining, in: Proceedings of the CVPR, 2016.
[149] S. Gidaris, N. Komodakis, Locnet: Improving localization accuracy for object detection, in: Proceedings of the CVPR, 2016.
[150] S. Zagoruyko, A. Lerer, T.-Y. Lin, P.O. Pinheiro, S. Gross, S. Chintala, P. Dollár, A multipath network for object detection, in: Proceedings of the BMVC, 2016.
[151] X. Lu, B. Li, Y. Yue, Q. Li, J. Yan, Grid R-CNN, in: Proceedings of the CVPR, 2019.
[152] L. Tychsen-Smith, L. Petersson, Improving object localization with fitness NMS and bounded IOU loss, 2017. arXiv: 1711.00164.
[153] B. Yang, J. Yan, Z. Lei, S.Z. Li, Craft objects from images, in: Proceedings of the CVPR, 2016.
[154] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Proceedings of the NeurIPS, 2014.
[155] J.-Y. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proceedings of the ICCV, 2017.

[156] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, 2015. arXiv: 1511.06434.
[157] A. Brock, J. Donahue, K. Simonyan, Large scale GAN training for high fidelity natural image synthesis, 2018. arXiv: 1809.11096.
[158] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, S. Yan, Perceptual generative adversarial networks for small object detection, in: Proceedings of the CVPR, 2017.
[159] X. Wang, A. Shrivastava, A. Gupta, A-fast-RCNN: hard positive generation via adversary for object detection, in: Proceedings of the CVPR, 2017.
[160] R.G. Kaiming He, P. DollÃ‘ro, Rethinking imagenet pre-training, 2018. arXiv: 1811.08883.
[161] R. Zhu, S. Zhang, X. Wang, L. Wen, H. Shi, L. Bo, T. Mei, Scratchdet: exploring to train single-shot object detectors from scratch, in: Proceedings of the CVPR, 2019.
[162] Z. Shen, H. Shi, R. Feris, L. Cao, S. Yan, D. Liu, X. Wang, X. Xue, T.S. Huang, Learning object detectors from scratch with gated recurrent feature pyramids, 2017. arXiv: 1712.00886.
[163] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, 2015. arXiv: 1503.02531.
[164] Q. Li, S. Jin, J. Yan, Mimicking very efficient network for object detection, in: Proceedings of the CVPR, 2017.
[165] N. Bodla, B. Singh, R. Chellappa, L.S. Davis, Soft-NMS – improving object detection with one line of code, in: Proceedings of the ICCV, 2017.
[166] J. Hosang, R. Benenson, B. Schiele, Learning non-maximum suppression, in: Proceedings of the CVPR, 2017.
[167] 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: Proceedings of the CVPR, 2017.
[168] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, J. Sun, Light-head R-CNN:in defense of two-stage object detector, 2017. arXiv: 1711.07264.
[169] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Inverted residuals and linear bottlenecks: mobile networks for classification, detection and segmentation, 2018. arXiv: 1801.04381.
[170] A. Wong, M.J. Shafiee, F. Li, B. Chwyl, Tiny SSD: A tiny single-shot detection deep convolutional neural network for real-time embedded object detection, 2018. arXiv: 1802.06488.
[171] Y. Li, J. Li, W. Lin, J. Li, Tiny-DSOD: lightweight object detection for resource-restricted usages, 2018. arXiv: 1807.11013.
[172] D. Almeida, H. Lee, K. Sohn, D. Shang, Understanding and improving convolutional neural networks via concatenated rectified linear units, in: Proceedings of the ICML, 2016.
[173] Y.D. Kim, E. Park, S. Yoo, T. Choi, L. Yang, D. Shin, Compression of deep convolutional neural networks for fast and low power mobile applications, Computer Science, 2015.
[174] Y. He, X. Zhang, J. Sun, Channel pruning for accelerating very deep neural networks, in: Proceedings of the ICCV, 2017.
[175] Y. Gong, L. Liu, M. Yang, L. Bourdev, Compressing deep convolutional networks using vector quantization, Computer Science, 2014.
[176] Y. Lin, S. Han, H. Mao, Y. Wang, W.J. Dally, Deep gradient compression: Reducing the communication bandwidth for distributed training, 2017. arXiv: 1712.01887.
[177] J. Wu, L. Cong, Y. Wang, Q. Hu, J. Cheng, Quantized convolutional neural networks for mobile devices, in: Proceedings of the CVPR, 2016.
[178] S. Han, H. Mao, W.J. Dally, Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding, in: Proceedings of the Fiber, 2015.
[179] S. Han, J. Pool, J. Tran, W. Dally, Learning both weights and connections for efficient neural network, in: Proceedings of the NeurIPS, 2015.
[180] E. Osuna, R. Freund, F. Girosit, Training support vector machines: an application to face detection, in: Proceedings of the CVPR, 1997.
[181] M. Rätsch, S. Romdhani, T. Vetter, Efficient face detection by a cascaded support vector machine using haar-like features, in: Proceedings of the Joint Pattern Recognition Symposium, 2004.
[182] S. Romdhani, P. Torr, B. Scholkopf, A. Blake, Computationally efficient face detection, in: Proceedings of the ICCV, 2001.
[183] X. Sun, P. Wu, S.C. Hoi, Face detection using deep learning: an improved faster RCNN approach, in: Proceedings of the Neurocomputing, 2018.
[184] Y. Liu, M.D. Levine, Multi-path region-based convolutional neural network for accurate detection of unconstrained" hard faces", in: Proceedings of the 14th Conference on Computer and Robot Vision (CRV), 2017, 2017.
[185] X. Tang, D.K. Du, Z. He, J. Liu, Pyramidbox: a context-assisted single shot face detector, in: Proceedings of the ECCV, 2018.
[186] C. Chi, S. Zhang, J. Xing, Z. Lei, S.Z. Li, X. Zou, Selective refinement network for high performance face detection, 2018. arXiv: 1809.02693.
[187] J. Li, Y. Wang, C. Wang, Y. Tai, J. Qian, J. Yang, C. Wang, J. Li, F. Huang, Dsfd: Dual shot face detector, in: Proceedings of the CVPR, 2019.
[188] J. Zhang, X. Wu, J. Zhu, S.C. Hoi, Feature agglomeration networks for single stage face detection, 2017. arXiv: 1712.00721.
[189] M. Najibi, P. Samangouei, R. Chellappa, L. Davis, SSH: single stage headless face detector, in: Proceedings of the ICCV, 2017.
[190] Z. Hao, Y. Liu, H. Qin, J. Yan, X. Li, X. Hu, Scale-aware face detection, in: Proceedings of the CVPR, 2017.
[191] H. Wang, Z. Li, X. Ji, Y. Wang, Face R-CNN, 2017. arXiv: 1706.01061.
[192] K. Zhang, Z. Zhang, Z. Li, Y. Qiao, Joint face detection and alignment using multi-task cascaded convolutional networks, in: Proceedings of the IEEE Signal Processing Letters, 2016.
[193] P. Samangouei, M. Najibi, L. Davis, R. Chellappa, Face-magnet: magnifying feature maps to detect small faces, 2018. arXiv: 1803.05258.
[194] C. Zhang, X. Xu, D. Tu, Face detection using improved faster RCNN, 2018. arXiv: 1802.02142.
[195] C. Zhu, Y. Zheng, K. Luu, M. Savvides, CMS-RCNN: contextual multi-scale region-based CNN for unconstrained face detection, in: Proceedings of the Deep Learning for Biometrics, 2017.
[196] B. Yu, D. Tao, Anchor cascade for efficient face detection, 2018. arXiv: 1805.03363.
[197] K. Zhang, Z. Zhang, H. Wang, Z. Li, Y. Qiao, W. Liu, Detecting faces using inside cascaded contextual CNN, in: Proceedings of the ICCV, 2017.
[198] Y. Wang, X. Ji, Z. Zhou, H. Wang, Z. Li, Detecting faces using region-based fully convolutional networks, 2017. arXiv: 1709.05256.
[199] V. Jain, E. Learned-Miller, FDDB: A Benchmark for Face Detection in Unconstrained Settings, Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010.
[200] S. Yang, P. Luo, C.-C. Loy, X. Tang, Wwider face: a face detection benchmark, in: Proceedings of the CVPR, 2016.
[201] J. Han, W. Nam, P. Dollar, Local decorrelation for improved detection, in: Proceedings of the NeurIPS, 2014.
[202] P. Dollár, Z. Tu, P. Perona, S. Belongie, Integral channel features, in: Proceedings of the BMVC, 2009.
[203] P. Dollár, R. Appel, S. Belongie, P. Perona, Fast feature pyramids for object detection, in: Proceedings of the TPAMI, 2014.
[204] C.P. Papageorgiou, M. Oren, T. Poggio, A general framework for object detection, in: Proceedings of the ICCV, 1998.
[205] G. Brazil, X. Yin, X. Liu, Illuminating pedestrians via simultaneous detection & segmentation, 2017. arXiv: 1706.08564.
[206] X. Du, M. El-Khamy, J. Lee, L. Davis, Fused DNN: a deep neural network fusion approach to fast and robust pedestrian detection, in: Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), 2017.
[207] S. Wang, J. Cheng, H. Liu, M. Tang, PCN: part and context information for pedestrian detection with CNNS, 2018. arXiv: 1804.04483.
[208] D. Xu, W. Ouyang, E. Ricci, X. Wang, N. Sebe, Learning cross-modal deep representations for robust pedestrian detection, in: Proceedings of the CVPR, 2017.
[209] R. Benenson, M. Omran, J. Hosang, B. Schiele, Ten years of pedestrian detection, what have we learned? in: Proceedings of the ECCV, 2014.
[210] Z. Cai, M. Saberian, N. Vasconcelos, Learning complexity-aware cascades for deep pedestrian detection, in: Proceedings of the ICCV, 2015.
[211] P. Sermanet, K. Kavukcuoglu, S. Chintala, Y. LeCun, Pedestrian detection with unsupervised multi-stage feature learning, in: Proceedings of the CVPR, 2013.
[212] L. Zhang, L. Lin, X. Liang, K. He, Is faster R-CNN doing well for pedestrian detection? in: Proceedings of the ECCV, 2016.
[213] X. Wang, T. Xiao, Y. Jiang, S. Shao, J. Sun, C. Shen, Repulsion loss: detecting pedestrians in a crowd, in: Proceedings of the CVPR, 2018.
[214] S. Zhang, L. Wen, X. Bian, Z. Lei, S.Z. Li, Occlusion-aware R-CNN: detecting pedestrians in a crowd, in: Proceedings of the ECCV, 2018.
[215] J. Mao, T. Xiao, Y. Jiang, Z. Cao, What can help pedestrian detection? in: Proceedings of the CVPR, 2017.
[216] Y. Tian, P. Luo, X. Wang, X. Tang, Deep learning strong parts for pedestrian detection, in: Proceedings of the CVPR, 2015.
[217] W. Ouyang, X. Wang, Joint deep learning for pedestrian detection, in: Proceedings of the ICCV, 2013.
[218] M. Mathias, R. Benenson, R. Timofte, L. Van Gool, Handling occlusions with Franken-classifiers, in: Proceedings of the ICCV, 2013.
[219] W. Ouyang, X. Zeng, X. Wang, Modeling mutual visibility relationship in pedestrian detection, in: Proceedings of the CVPR, 2013.
[220] G. Duan, H. Ai, S. Lao, A structural filter approach to human detection, in: Proceedings of the ECCV, 2010.
[221] M. Enzweiler, A. Eigenstetter, B. Schiele, D.M. Gavrila, Multi-cue pedestrian classification with partial occlusion handling, in: Proceedings of the CVPR, 2010.
[222] C. Zhou, J. Yuan, Bi-box regression for pedestrian detection and occlusion estimation, in: Proceedings of the ECCV, 2018.
[223] W. Ouyang, X. Wang, A discriminative deep model for pedestrian detection with occlusion handling, in: Proceedings of the CVPR, 2012.
[224] S. Tang, M. Andriluka, B. Schiele, Detection and tracking of occluded people, in: Proceedings of the IJCV, 2014.
[225] W. Ouyang, X. Wang, Single-pedestrian detection aided by multi-pedestrian detection, in: Proceedings of the CVPR, 2013.
[226] V.D. Shet, J. Neumann, V. Ramesh, L.S. Davis, Bilattice-based logical reasoning for human detection, in: Proceedings of the CVPR, 2007.
[227] Y. Zhou, L. Liu, L. Shao, M. Mellor, Dave: a unified framework for fast vehicle detection and annotation, in: Proceedings of the ECCV, 2016.
[228] T. Gebru, J. Krause, Y. Wang, D. Chen, J. Deng, L. Fei-Fei, Fine-grained car detection for visual census estimation, in: Proceedings of the AAAI, 2017.
[229] S. Majid Azimi, Shuffledet: real-time vehicle detection network in on-board embedded UAV imagery, in: Proceedings of the ECCV, 2018.
[230] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, S. Hu, Traffic-sign detection and classification in the wild, in: Proceedings of the CVPR, 2016.
[231] A. Pon, O. Adrienko, A. Harakeh, S.L. Waslander, A hierarchical deep architecture and mini-batch selection method for joint traffic sign and light detection, in: Proceedings of the Conference on Computer and Robot Vision (CRV), 2018.
[232] W. Ke, J. Chen, J. Jiao, G. Zhao, Q. Ye, Srn: side-output residual network for object symmetry detection in the wild, in: Proceedings of the CVPR, 2017.

[233] W. Shen, K. Zhao, Y. Jiang, Y. Wang, Z. Zhang, X. Bai, Object skeleton extraction in natural images by fusing scale-associated deep side outputs, in: Proceedings of the CVPR, 2016.
[234] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, T. Duerig, et al., The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale, 2018. arXiv: 1811.00982.
[235] A. Gupta, P. Dollar, R. Girshick, LVIS: a dataset for large vocabulary instance segmentation, in: Proceedings of the CVPR, 2019.
[236] N. Bodla, B. Singh, R. Chellappa, L.S. Davis, Soft-NMS–improving object detection with one line of code, in: Proceedings of the ICCV, 2017.
[237] Y. Wu, K. He, Group normalization, in: Proceedings of the ECCV, 2018.
[238] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, in: Proceedings of the CVPR, 2018.
[239] Y. Li, Y. Chen, N. Wang, Z. Zhang, Scale-aware trident networks for object detection, 2019. arXiv: 1901.01892.
[240] X. Zhou, J. Zhuo, P. Krahenbuhl, Bottom-up object detection by grouping extreme and center points, in: Proceedings of the CVPR, 2019.
[241] Z. Tian, C. Shen, H. Chen, T. He, FCOS: fully convolutional one-stage object detection, 2019. arXiv: 1904.01355.
[242] B. Yang, J. Yan, Z. Lei, S.Z. Li, Aggregate channel features for multi-view face detection, in: Proceedings of the Biometrics (IJCB), 2014.
[243] S. Yang, P. Luo, C.C. Loy, X. Tang, From facial parts responses to face detection: a deep learning approach, in: Proceedings of the ICCV, 2015.
[244] E. Ohn-Bar, M.M. Trivedi, To boost or not to boost? On the limits of boosted trees for object detection, in: Proceedings of the ICPR, 2016.
[245] S. Yang, Y. Xiong, C.C. Loy, X. Tang, Face detection through scale-friendly deep convolutional networks, 2017. arXiv: 1706.02863.
[246] Y.Y. J. Wang, G. Yu, Face attention network: an effective face detector for the occluded faces, 2017. arXiv: 1711.07246.
[247] X. Xu, C. Zhang, D. Tu, Face detection using improved faster RCNN, 2018. arXiv: 1802.02142.
[248] W. Tian, H. Shen, W. Deng, Z. Wang, Learning better features for face detection with feature fusion and segmentation supervision, 2018. arXiv: 1811.08557.
[249] S. Zhang, L. Wen, H. Shi, Z. Lei, Single-shot scale-aware network for real-time face detection, in: Proceedings of the IJCV, 2019.
[250] W. Liu, S. Liao, W. Ren, W. Hu, High-level semantic feature detection: a new perspective for pedestrian detection, in: Proceedings of the CVPR, 2019.
[251] Z. Li, X. Tang, J. Han, J. Liu, Pyramidbox++: high performance detector for finding tiny face, 2019. arXiv: 1904.00386.
[252] Y. Zhang, X. Xu, X. Liu, Robust and high performance face detector, 2019. arXiv: 1901.02350.
[253] S. Zhang, R. Zhu, X. Wang, H. Shi, T. Fu, S. Wang, T. Mei, S.Z. Li, Improved selective refinement network for face detection, 2019. arXiv: 1901.06651.
[254] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, S. Zafeiriou, Retinaface: single-stage dense face localisation in the wild, 2019. arXiv: 1905.00641.
[255] F. Zhang, X. Fan, G. Ai, J. Song, Y. Qin, J. Wu, Accurate face detection for high performance, 2019. arXiv: 1905.00641.
[256] S. Zhang, C. Chi, Z. Lei, S.Z. Li, Refineface: refinement neural network for high performance face detection, 2019. arXiv: 1909.04376.
[257] S. Zhang, R. Benenson, B. Schiele, Citypersons: a diverse dataset for pedestrian detection, in: Proceedings of the CVPR, 2017.
[258] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the CVPR, 2016.
[259] P. Dollar, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: an evaluation of the state of the art, in: Proceedings of the TPAMI, 2012.
[260] A. Ess, B. Leibe, L. Van Gool, Depth and appearance for mobile scene analysis, in: Proceedings of the ICCV, 2007.
[261] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: the kitti dataset, 2013.
[262] H.M. Ruiqi Lu, Semantic head enhanced pedestrian detection in a crowd, 2019. arXiv: 1911.11985.
[263] Y. Pang, J. Xie, M.H. Khan, R.M. Anwar, F.S. Khan, L. Shao, Mask-guided attention network for occluded pedestrian detection, in: Proceedings of the ICCV, 2019.
[264] Y. Hu, J. Xie, J. Zhang, L. Lin, Y. Li, S.C. Hoi, Attribute-aware pedestrian detection in a crowd, 2019. arXiv: 1910.09188.
[265] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, M. Pietikäinen, Deep learning for generic object detection: a survey, 2018. arXiv: 1809.02165.
[266] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, R. Qu, A survey of deep learning-based object detection,
[267] F. Sultana, A. Sufian, P. Dutta, A review of object detection models based on convolutional neural network, 2019. arXiv: 1905.01614.
[268] S. Agarwal, J.O.D. Terrail, F. Jurie, Recent advances in object detection in the age of deep convolutional neural networks, 2018. arXiv: 1809.03193.
[269] Z.-Q. Zhao, P. Zheng, S.-t. Xu, X. Wu, Object detection with deep learning: a review, in: Proceedings of the IEEE Transactions on Neural Networks and Learnzing systems, 2019.
[270] B. Zoph, V. Vasudevan, J. Shlens, Q.V. Le, Learning transferable architectures for scalable image recognition, in: Proceedings of the CVPR, 2018.
[271] M. Tan, Q.V. Le, Efficientnet: rethinking model scaling for convolutional neural networks, 2019. arXiv: 1905.11946.
[272] Y. Chen, T. Yang, X. Zhang, G. Meng, C. Pan, J. Sun, Detnas: neural architecture search on object detection, 2019. arXiv: 1903.10979.
[273] G. Ghiasi, T.-Y. Lin, Q.V. Le, NAS-FPN: learning scalable feature pyramid architecture for object detection, in: Proceedings of the CVPR, 2019.
[274] B. Zoph, E.D. Cubuk, G. Ghiasi, T.-Y. Lin, J. Shlens, Q.V. Le, Learning data augmentation strategies for object detection, 2019. arXiv: 1906.11172.
[275] X. Dong, L. Zheng, F. Ma, Y. Yang, D. Meng, Few-example object detection with model communication, in: Proceedings of the TPAMI, 2018.
[276] E. Schwartz, L. Karlinsky, J. Shtok, S. Harary, M. Marder, S. Pankanti, R. Feris, A. Kumar, R. Giries, A.M. Bronstein, Repmet: representative-based metric learning for classification and one-shot object detection, in: Proceedings of the CVPR, 2019.
[277] H. Chen, Y. Wang, G. Wang, Y. Qiao, LSTD: a low-shot transfer detector for object detection, in: Proceedings of the AAAI, 2018.
[278] C. Peng, T. Xiao, Z. Li, Y. Jiang, X. Zhang, K. Jia, G. Yu, J. Sun, Megdet: a large mini-batch object detector, in: Proceedings of the CVPR, 2018.
[279] K. Shmelkov, C. Schmid, K. Alahari, Incremental learning of object detectors without catastrophic forgetting, in: Proceedings of the ICCV, 2017.

Xiongwei WU received the bachelor's degree in computer science from Zhejiang University, Zhejiang, P.R. China. He is currently the Ph.D. student in the School of Information Systems, Singapore Management University, Singapore, supervised by Prof. Steven Hoi. His research directions mainly focus on object detection and deep learning.

Doyen SAHOO is a Research Scientist at Salesforce Research Asia. Prior to this, he was serving as Adjunct faculty in Singapore Management University, and was also a Research Fellow at the Living Analytics Research Center. He works on Online Learning, Deep Learning and related machine learning applications. He obtained his Ph.D. from Singapore Management University, and B.Eng from Nanyang Technological University.



Prof. Steven C.H. HOI is currently Managing Director of Salesforce Research Asia at Salesforce, located in Singapore. He has been also a tenured Associate Professor of the School of Information Systems at Singapore Management University, Singapore. Prior to joining SMU, he was a tenured Associate Professor of the School of Computer Engineering at Nanyang Technological University, Singapore. He received his Bachelor degree in Computer Science from Tsinghua University, Beijing, China, in 2002, and both his Master and Ph.D. degrees in Computer Science and Engineering from the Chinese University of Hong Kong, in 2004 and 2006, respectively.

Please cite this article as: X. Wu, D. Sahoo and S.C.H. Hoi, Recent advances in deep learning for object detection, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.01.085